Bitcoin och Cryptocurrency Technologies

Arvind Narayanan, Joseph Bonneau, Edward Felten,
Andrew Miller, Steven Goldfeder

med ett förord av Jeremy Clark
Utkast – 9 februari 2016

Feedback välkomnas! E-post bitcoinbook@lists.cs.princeton.edu

För det senaste utkastet och kompletterande material inklusive programmeringsuppgifter,
se vår Coursera kurs .

Den officiella versionen av denna bok kommer att publiceras av Princeton University Press 2016.
Om du vill bli meddelad när den är tillgänglig, vänligen registrera dig här .

Introduktion till boken

Det finns mycket spänning kring Bitcoin och kryptovalutor. Optimister hävdar att Bitcoin kommer fundamentalt förändra betalningar, ekonomi och till och med politik runt om i världen. Pessimisterna hävdar Bitcoin är till sin natur trasig och kommer att drabbas av en oundviklig och spektakulär kollaps.
Bakom dessa olika åsikter ligger betydande förvirring om vad Bitcoin är och hur det fungerar. Vi skrev den här boken för att hjälpa till att skära igenom hypen och komma till kärnan av vad som gör Bitcoin unik. För att verkligen förstå vad som är speciellt med Bitcoin måste vi förstå hur det fungerar på ett tekniskt sätt nivå. Bitcoin är verkligen en ny teknik och vi kan bara komma så långt genom att förklara det på ett enkelt sätt analogier till tidigare teknologier.
Vi antar att du har en grundläggande förståelse för datavetenskap - hur datorer fungerar, data strukturer och algoritmer, och viss programmeringserfarenhet. Om du är en grundutbildning eller doktorand i datavetenskap, en mjukvaruutvecklare, en entreprenör eller en teknik hobbyist, den här läroboken är för dig.
I den här boken tar vi upp de viktiga frågorna om Bitcoin. Hur fungerar Bitcoin? Vad gör det annorlunda? Hur säkra är dina bitcoins? Hur anonyma är Bitcoin-användare? Vad applikationer kan vi bygga med Bitcoin som plattform? Kan kryptovalutor regleras? Om vi var designa en ny kryptovaluta idag, vad skulle vi ändra? Vad kan framtiden erbjuda?
Varje kapitel har en serie läxfrågor som hjälper dig att förstå dessa frågor på ett djupare sätt nivå. Dessutom finns det en serie programmeringsuppgifter där du kommer att implementera olika komponenter i Bitcoin i förenklade modeller. Om du är en auditiv elev, det mesta av materialet i detta boken finns även som en serie videoföreläsningar. Här hittar du alla dessa på vår Coursera kurs . Du bör också komplettera din inlärning med information du kan hitta online inklusive Bitcoin-wikin, forum och forskningsartiklar och genom att interagera med dina kamrater och Bitcoin-communityt.
Efter att ha läst den här boken kommer du att veta allt du behöver för att kunna skilja fakta från fiktion när läsa påståenden om Bitcoin och andra kryptovalutor. Du har den konceptuella grunden du behöver konstruera säker programvara som interagerar med Bitcoin-nätverket. Och du kommer att kunna integrera idéer från Bitcoin i dina egna projekt.

Ett tack

Vi är oerhört tacksamma för eleverna som hjälpte till att utveckla programmeringsuppgifter och till alla som gett feedback på utkasten till denna bok. Princeton-studenter Shivam Agarwal, Miles Carlsten, Paul Ellenbogen, Pranav Gokhale, Alex Iriza, Harry Kalodner och Dillon Reisman, och Stanford-studenterna Allison Berke, Benedikt Bünz och Alex Leishman förtjänar särskilt beröm. Var också tacksamma till Dan Boneh och Albert Szmigielski.

Förord — Den långa vägen till Bitcoin

Vägen till Bitcoin är full av liken av misslyckade försök. Jag har sammanställt en lista på ca hundratals kryptografiska betalningssystem, både elektroniska kontanter och kreditkortsbaserade teknologier anmärkningsvärt på något sätt. Vissa är akademiska förslag som har citerats väl medan andra är faktiska system som har installerats och testats. Av alla namn på den här listan finns det förmodligen bara ett som du känner igen - PayPal. Och PayPal överlevde bara för att det snabbt svängde bort från originalet idé om kryptografiska betalningar på handhållna enheter!

Det finns mycket att lära av den här historien. Var kommer idéerna i Bitcoin ifrån? Varför göra några teknologier överlever medan många andra dör? Vad krävs för att komplexa tekniska innovationer ska kommersialiseras framgångsrikt? Om inte annat kommer den här historien att ge dig en uppskattning av hur anmärkningsvärt är det att vi äntligen har en verklig, fungerande betalningsmekanism som är inbyggd i Internet.

Tabell 1: Anmärkningsvärda elektroniska betalningssystem och förslag

---

Traditionella finansiella arrangemang

Tillbaka i tiden innan det fanns regeringar, innan det fanns valuta, ett system som fungerade för att anskaffa varor var byteshandel. Låt oss säga att Alice vill ha ett verktyg och Bob vill ha medicin. Om var och en av dem
råkar ha det den andra personen behöver, då kan de byta och båda tillfredsställa sina behov.

Å andra sidan, låt oss säga att Alice har mat som hon är villig att byta mot ett verktyg, medan Bob, som har en verktyg, har inget behov av mat. Han vill ha medicin istället. Alice och Bob kan inte byta med var och en annat, men om det finns en tredje person, Carol, som har medicin som hon är villig att byta mot mat, då blir möjligt att ordna ett trevägsbyte där alla får det de behöver.

Nackdelen är förstås samordning - att arrangera en grupp människor vars behov och önskemål anpassa, på samma plats samtidigt. Två system uppstod för att lösa samordning: kredit och kontanter. Historiker, antropologer och ekonomer diskuterar vilken av de två som utvecklades först, men det är oväsentliga för våra syften.

I ett kreditbaserat system, i exemplet ovan, skulle Alice och Bob kunna handla med varandra. Bob skulle ge Alice verktyget och Bob får en tjänst som är skyldig honom. Med andra ord, Alice har en skuld att hon behöver göra upp med Bob någon gång i framtiden. Alices materiella behov är nu tillfredsställda, men hon har en skuld som hon skulle vilja avskriva, så det är hennes nya "önskemål". Om Alice stöter på Carol i i framtiden kan Alice byta ut sin mat mot Carols medicin och sedan gå tillbaka till Bob med medicinen och avskriva skulden.

Å andra sidan, i ett kontantbaserat system skulle Alice köpa verktyget av Bob. Senare kanske hon säljer hennes mat till Carol, och Carol kan sälja sin medicin till Bob och slutföra cykeln. Dessa affärer kan ske i valfri ordning, förutsatt att köparen i varje transaktion har kontanter till hands. I slutändan av naturligtvis är det som om inga pengar någonsin bytt ägare.

Inget av systemen är klart överlägset. Ett kontantbaserat system måste "stövlas" med några initialer tilldelning av kontanter, utan vilka inga affärer kan ske. Ett kreditbaserat system behöver inte bootstrapping, men nackdelen är att alla som är skyldiga en skuld tar en viss risk. Det finns en chans att den andra personen aldrig kommer tillbaka för att reglera skulden.
Kontanter låter oss också vara exakta om hur mycket något är värt. Om du byter är det svårt säga om ett verktyg är värt mer än medicin eller medicin är värt mer än mat. Kontanter låter oss använda siffror för att tala om värde. Det är därför vi använder ett blandat system idag – även när vi använder det kredit, mäter vi skulder i mängden kontanter det skulle ta för att reglera den.
Dessa idéer dyker upp i många sammanhang, särskilt onlinesystem där användare handlar med virtuella varor någon typ. Till exempel måste peer-to-peer fildelningsnätverk hantera problemet med "freeloaders", det vill säga användare som laddar ner filer utan att dela i sin tur. Medan du byter filer kan

---

arbete, det finns också frågan om samordning: att hitta den perfekta personen som har exakt filen dig vill ha och vill ha exakt den fil du har. I projekt som MojoNation och akademiska förslag som Karma, användare får en viss initial tilldelning av virtuella kontanter som de måste spendera för att ta emot en fil och tjäna
när de skickar en kopia av en fil till en annan användare. I båda fallen hjälper en eller flera centrala servrar att behålla
spåra användarnas saldon och kan erbjuda växlingstjänster mellan deras interna valuta och traditionell valuta. Även om MojoNation inte överlevde tillräckligt länge för att genomföra ett sådant utbyte, det blev den intellektuella förfadern till vissa protokoll som används idag: BitTorrent och Tahoe-LAFS.

Problemet med kreditkort online

Kredit och kontanter är grundläggande idéer, till den grad att vi kan sortera mängden elektroniska betalningsmetoder i två högar. Bitcoin ligger uppenbarligen i "kontanthögen", men låt oss titta på den andra först.
Kreditkortstransaktioner är den dominerande betalningsmetoden som används på webben idag. Om du har någonsin köpt något från en onlinesäljare som Amazon, du vet hur arrangemanget går.
Du skriver in dina kreditkortsuppgifter, du skickar dem till Amazon och sedan vänder Amazon om med dessa kreditkortsuppgifter och de pratar med "systemet" - ett finansiellt system som involverar processorer, banker, kreditkortsföretag och andra mellanhänder.
Å andra sidan, om du använder något som PayPal, är det du ser en mellanliggande arkitektur.
Det finns ett företag som sitter mellan dig och säljaren, så du skickar dina kreditkortsuppgifter till detta mellanhand, som godkänner transaktionen och meddelar säljaren. Förmedlaren kommer att lösa sitt balansera med säljaren i slutet av varje dag.
Vad du vinner på den här arkitekturen är att du inte behöver ge säljaren dina kreditkortsuppgifter, vilket kan vara en säkerhetsrisk. Du kanske inte ens behöver ge säljaren din identitet, vilket skulle göra förbättra din integritet också. Nackdelen är att du tappar enkelheten i att interagera direkt med säljaren. Både du och säljaren kan behöva ha ett konto hos samma mellanhand.
Idag är de flesta av oss bekväma med att ge ut vår kreditkortsinformation när vi handlar online, eller åtminstone har vi motvilligt accepterat det. Vi är också vana vid att företag samlar in data om våra online shopping och surfaktivitet. Men på 1990-talet var nätet nytt, standarder för
Kryptering på protokollnivå höll precis på att växa fram, och dessa farhågor gjorde konsumenterna djupt osäkra och tveksam. I synnerhet ansågs det vara galet att lämna över dina kreditkortsuppgifter till online leverantörer med okänt rykte över en osäker kanal. I en sådan miljö fanns det mycket intresse för mellanliggande arkitektur.
Ett företag som heter FirstVirtual var en tidig betalningsförmedlare, grundad 1994. De har för övrigt var ett av de första företagen att skapa ett rent virtuellt kontor med anställda spridda över hela

land och kommunicera över Internet — därav namnet.
5

Sida 6

FirstVirtuals föreslagna system var lite som PayPals nuvarande system men föregick det i många år. Som användare skulle du registrera dig hos dem och ange dina kreditkortsuppgifter. När du vill köpa något från en säljare kontaktar säljaren FirstVirtual med information om den begärda betalningen, FirstVirtual bekräftar dessa uppgifter med dig och om du godkänner faktureras ditt kreditkort. Men två detaljer är intressanta. För det första skedde all denna kommunikation via e-post; webbläsare tillbaka i dag har precis börjat stödja krypteringsprotokoll som HTTPS och multi-party betalningsprotokollets karaktär tillade andra komplexiteter. (Andra mellanhänder tog tillvägagångssättet koda information till URL:er eller använda ett anpassat krypteringsprotokoll ovanpå HTTP.) För det andra, den kunden skulle ha nittio dagar på sig att bestrida avgiften, och handlaren skulle få pengarna först efter tre månader! Idag får handlaren betalt omedelbart, men det finns fortfarande risken att kunden kommer att lämna in en återkrav eller bestrida kreditkortsutdraget. Om det händer, handlaren måste återlämna betalningen till kreditkortsföretaget.

I mitten av 90-talet fanns det ett konkurrerande förhållningssätt till den mellanliggande arkitekturen som vi kommer att kalla
SET arkitektur. SET slipper också behovet för kunder att skicka kreditkortsinformation till handlare, men det undviker dessutom att användaren behöver registrera sig hos mellanhanden. I SET, när du är redo att göra ett köp skickar din webbläsare din syn på transaktionsdetaljerna till en shopping applikation på din dator som tillsammans med dina kreditkortsuppgifter krypterar den på ett sådant sätt att endast mellanhanden kan dekryptera det, och ingen annan kan (inklusive säljaren). Efter att ha krypterat dina uppgifter på det här sättet kan du skicka dem till säljaren med vetskap om att det är säkert. Säljaren vidarebefordrar blint
den krypterade informationen till mellanhanden — tillsammans med sin egen syn på transaktionsdetaljerna. De mellanhand dekrypterar dina uppgifter och godkänner transaktionen endast om din syn överensstämmer med säljarens
se.

SET var en standard utvecklad av VISA och MasterCard, tillsammans med många tekniska tungviktare för dagen: Netscape, IBM, Microsoft, Verisign och RSA. Det var en paraplyspecifikation som förenade flera befintliga förslag.

Ett företag som implementerade SET hette CyberCash. Det var ett intressant företag i många sätt. Förutom kreditkortsbetalningshantering hade de en digital kontantprodukt som heter CyberCoin. Detta var ett mikrobetalningssystem - avsett för små betalningar som att betala några ören för att läsa en tidningsartikel på nätet. Det innebar att du förmodligen aldrig skulle ha mer än $10 i din CyberCoin-konto när som helst. Ändå lyckades de, underhållande nog, få den amerikanska regeringen (FDIC) försäkring för varje konto för upp till 100 000 USD.

Det finns mer. När CyberCash fungerade fanns det ett missriktat – och nu övergivet – USA statlig begränsning av exporten av kryptografi, som ansågs vara ett vapen. Det betydde programvara som innehöll meningsfull kryptering kunde inte erbjudas för nedladdning till andra användare länder. CyberCash kunde dock få ett särskilt undantag för sin programvara från Utrikesdepartementet. Regeringens argument var att extrahera krypteringstekniken av CyberCashs mjukvara skulle vara svårare än att skriva krypton från början.
6

Sida 7

Slutligen har CyberCash den tvivelaktiga utmärkelsen att vara ett av de få företag som påverkas av Y2K bugg — det fick deras betalningsbehandlingsprogram att dubbelfakturera vissa kunder. De gick senare

gick i konkurs 2001. Deras immateriella rättigheter förvärvades av Verisign som sedan vände och sålde den till PayPal där den bor idag.

Varför fungerade inte SET? Det grundläggande problemet har att göra med certifikat. Ett certifikat är ett sätt att säkert associera en kryptografisk identitet, det vill säga en offentlig nyckel, med en verklig identitet. Det är vad en webbplatsen måste få, från företag som Verisign som kallas certifieringsmyndigheter, i ordning för att visas som säker i din webbläsare (typiskt indikerad med en låsikon). Sätter säkerheten före användbarhet, beslutade CyberCash och SET att inte bara processorer och handlare i deras system måste få certifikat, alla användare måste också få ett. Att få ett certifikat är ungefär som trevligt som att göra dina skatter, så systemet var en katastrof. Under decennierna har vanliga användare gjort det sa ett bestämt och kollektivt "nej" till alla system som kräver slutanvändarcertifikat och sådana förslag har nu förpassats till akademiska uppsatser. Bitcoin kringgår detta håriga problem skickligt genom att undvika verkliga identiteter helt och hållet. I Bitcoin är offentliga nycklar i sig de identiteter som användarna är känd, som vi kommer att se i kapitel 1.

I mitten av 90-talet, när SET standardiserades, letade också World Wide Web Consortium att standardisera ekonomiska betalningar. De ville göra det genom att utöka HTTP-protokollet istället så att användare inte skulle behöva extra programvara för transaktioner – de kunde bara använda sin webbläsare. Faktiskt, de hade ett mycket allmänt förslag på hur du skulle kunna utöka protokollet, och ett av användningsfallen som de hade gjorde betalningar. Detta hände aldrig -- hela förlängningsramverket var det aldrig distribueras i alla webbläsare. 2015, nästan två decennier senare, har W3C meddelat att de vill ta en ny spricka på det, och att Bitcoin kommer att vara en del av den standardiseringen den här gången. Givet allt tidigare misslyckanden kommer jag dock inte att hålla andan.

Från kredit till (krypto) kontanter

Låt oss nu övergå till kontanter. Vi jämförde kontanter och krediter tidigare och konstaterade att ett kontantsystem måste göra det vara "stövlade", men fördelen är att det undviker möjligheten för en köpare att inte betala sin skuld.

Kontanter erbjuder ytterligare två fördelar. Det första är bättre anonymitet. Eftersom ditt kreditkort är utfärdat i ditt namn kan banken spåra alla dina utgifter. Men när du betalar kontant kommer inte banken in i bilden, och den andra parten behöver inte veta vem du är. För det andra kan kontanter möjliggöra offlinetransaktioner där det inte finns något behov av att ringa hem till en tredje part för att få transaktion godkänd. Kanske senare går de till en tredje part som en bank för att sätta in pengarna, men så är det mycket mindre krångel.

Bitcoin erbjuder inte riktigt dessa två egenskaper, men kommer tillräckligt nära för att vara användbar. Bitcoin är det inte anonym till samma nivå som kontanter är. Du behöver inte använda din riktiga identitet för att betala i Bitcoin, men det är möjligt att dina transaktioner kan knytas samman baserat på den offentliga huvudboken för transaktioner [7]

---

med smarta algoritmer, och sedan vidare kopplad till din identitet om du inte är försiktig. Vi kommer in på röriga men fascinerande detaljer bakom Bitcoin-anonymitet i kapitel 6.

Bitcoin fungerar inte på ett helt offline sätt heller. Den goda nyheten är att det inte kräver en central server, istället förlitar sig på ett peer-to-peer-nätverk som är motståndskraftigt på det sätt som Internet i sig är. I Kapitel 3 kommer vi att titta på knep som "gröna adresser" och mikrobetalningar som gör att vi kan göra offline betalningar i vissa situationer eller under vissa antaganden.

De tidigaste idéerna om att tillämpa kryptografi på kontanter kom från David Chaum 1983. Låt oss förstå detta genom en fysisk analogi. Låt oss säga att jag börjar dela ut papperslappar som säger: ”Bäraren av detta sedel kan lösa in den för en dollar genom att presentera den för mig” med min signatur bifogad. Om folk litar på

att jag håller mitt löfte och anser att min signatur är oförglömlig, de kan skicka runt dessa bitar av papper precis som sedlar. Faktum är att sedlarna själva fick sin start som skuldebrev utfärdade av affärsbanker. Det är först i ganska ny historia som regeringar gick in för att centralisera penningmängd och lagligt kräver banker att lösa in sedlar.

Jag kan göra samma sak elektroniskt med digitala signaturer, men det blir den irriterande "dubbeln spending" problem — om du får en bit data som representerar en enhet virtuella kontanter kan du tjäna två (eller flera) kopior av den och skicka den vidare till olika personer. För att hålla fast vid vår liknelse, låt oss sträcka ut det en lite och anta att människor kan göra perfekta kopior och vi har inget sätt att skilja kopior från original. Kan vi lösa dubbla utgifter i denna värld?

Här är en möjlig lösning: Jag lägger in unika serienummer i varje lapp jag ger ut. När du tar emot en sådan lapp från någon, du kollar min signatur, men du ringer mig också på telefonen för att fråga om en lapp med det serienumret har redan använts. Förhoppningsvis säger jag nej, i så fall accepterar du notera. Jag kommer att anteckna serienumret som spenderat i min reskontra, och om du försöker spendera den sedeln kommer det inte fungerar eftersom mottagaren kommer att ringa mig och jag ska berätta att sedeln redan är förbrukad. Vad du ska behöver göra istället är att med jämna mellanrum ge mig alla anteckningar du har fått, så skickar jag detsamma till dig Antalet *nya* sedlar med färska serienummer.

Det här fungerar. Det är krångligt i verkligheten, men enkelt digitalt förutsatt att jag har satt upp en server till göra signering och journalföring av serienummer. Det enda problemet är att det här egentligen inte är några kontanter mer, eftersom det inte är anonymt - när jag skickar ett meddelande till dig kan jag spela in serienumret med din identitet, och jag kan göra detsamma när någon annan senare löser in den. Det betyder att jag kan behålla spåra alla platser där du spenderar dina pengar.

Det är här Chaums innovation kommer in. Han kom på att både hålla systemet anonymt och förhindra dubbla utgifter genom att uppfinna den digitala motsvarigheten till följande procedur: när jag utfärdar en ny anteckning till dig, *du* väljer serienumret. Du skriver ner det på papperet, men täcker det så att jag inte kan se det. Sedan skriver jag under, men kan fortfarande inte se serienumret. Detta kallas en "blind signatur" i kryptografi. Det är i ditt intresse att välja ett långt, slumpmässigt serienummer för att säkerställa att det gör det troligen vara unik. Jag behöver inte oroa dig för att du väljer ett serienummer som redan har valts — du kan bara skjuta dig själv i foten genom att göra det och sluta med en lapp som inte går att spendera.

Detta var det första seriösa digitala kontantförslaget. Det fungerar, men det kräver fortfarande en server som drivs av en central myndighet, till exempel en bank, och för alla att lita på den enheten. Dessutom behöver varje transaktion deltagandet av denna server att gå igenom. Om servern går ner tillfälligt, går betalningarna till ett stopp. Några år senare, 1988, gjorde Chaum i samarbete med två andra kryptografer Fiat och Naor föreslås *offline* elektroniska kontanter. Vid första anblicken kan detta tyckas vara omöjligt: om du försöker spendera samma digitala sedel eller mynt i två olika butiker, hur kan de stoppa detta om de inte är båda ansluten till samma betalningsnätverk eller centrala enhet?

Den smarta idén är att sluta oroa sig för att förhindra dubbla utgifter och fokusera på att upptäcka det efteråt faktum, när handlaren återansluter till bankservern. Det är trots allt därför du kan använda ditt kreditkort på ett flygplan även om det inte finns någon nätverksanslutning uppe i skyarna. Transaktionen bearbetning sker senare när flygbolaget kan återansluta till nätverket. Om ditt kort nekas, du är skyldig flygbolaget (eller din bank) pengar. Om du tänker efter så är det ganska mycket traditionell ekonomi

baserat på idén att upptäcka ett fel eller förlust, följt av ett försök att få tillbaka pengarna eller straffa gärningsmannen. Om du skriver en personlig check till någon har de ingen garanti för att pengar finns faktiskt på ditt konto, men de kan komma efter dig om checken studsar. Tänkbart om ett offline elektroniskt kontantsystem antogs allmänt, skulle rättssystemet komma att erkännas dubbla utgifter som ett brott.

Chaum, Fiat och Naors idé för att upptäcka dubbla utgifter var en intrikat kryptografisk dans. Vid en hög nivå, vad den uppnådde var detta: varje digitalt mynt som ges ut till dig kodar din identitet, men i sådana ett sätt som ingen förutom du, inte ens banken, kan avkoda det. Varje gång du spenderar ditt mynt, mottagaren kommer att kräva att du avkodar en slumpmässig delmängd av kodningen, och de kommer att hålla ett register över

detta. Denna avkodning räcker inte för att de ska kunna fastställa din identitet. Men om du någonsin dubblar spendera ett mynt, så kommer båda mottagarna att gå till banken för att lösa in sina sedlar, och när de gör det Detta kan banken sätta ihop de två informationsbitarna för att helt avkoda din identitet,

med en överväldigande hög sannolikhet.

Du kanske undrar om någon kan framställa dig som en dubbel spenderare i det här systemet. Säg att du spenderar ett mynt

med mig, och sedan vände jag mig om och försökte dubbla utgifterna (utan att lösa in den med banken och få ett nytt mynt med min identitet kodad). Det här kommer inte att fungera – den nya mottagaren kommer att be mig att göra det

avkoda en slumpmässig delmängd, och detta kommer nästan säkert inte att vara detsamma som delmängden du avkodade för

mig, så jag kommer inte att kunna följa deras begäran om avkodning.

Under åren har många kryptografer tittat på denna konstruktion och förbättrat den på olika sätt sätt. I Chaum-Fiat-Naor-schemat, om ett mynt är värt $100 och du vill köpa något som kostar bara $75, säg, det finns inget sätt att dela upp det myntet i $75 och en $25. Allt du kan göra är att gå tillbaka till

banken, växla in $100-myntet och be om ett $75-mynt och ett $25-mynt. Men en tidning av Okamoto och Ohta använder "Merkle trees" för att skapa ett system som låter dig dela upp dina mynt. Merkle träd skulle dyka upp i Bitcoin också, och vi kommer att träffa dem i kapitel 1. Chaum-Fiat-Naor-schemat lämnar också ett stort utrymme för effektivitetsförbättringar. I synnerhet tillämpningen av något kallade nollkunskapsbevis för detta schema (främst av Brands; och Camenisch, Hohenberger,

---

och Lysyanskaya) var mycket givande – nollkunskapsbevis har också tillämpats på Bitcoin när vi kommer att se i kapitel 6.

Men tillbaka till Chaum: han tog sina idéer och kommersialiserade dem. Han bildade ett företag 1989 som heter DigiCash, förmodligen det tidigaste företaget som försökte lösa problemet med onlinebetalningar. Dem hade om ett femårigt försprång på andra företag som FirstVirtual och CyberCash som vi just diskuteras. Själva kontanterna i Digicashs system hette Ecash och de hade ett annat system som hette cyberbucks. Det fanns banker som faktiskt implementerade det - några få i USA och minst en i Finland. Detta var på 1990-talet, långt före Bitcoin, vilket kan komma som en överraskning för vissa Bitcoin entusiaster som ser banker som teknikfobiska, antiinnovativa giganter.

Ecash bygger på Chaums protokoll. Kunderna är anonyma, så bankerna kan inte spåra hur de mår spendera sina pengar. Men handlare i ecash är inte anonyma. De måste lämna tillbaka mynt så snart när de tar emot dem, så att banken vet hur mycket de tjänar, vid vilka tidpunkter och så vidare.

Figur 2: Skärmdump av DigiCash

Figur 2 visar en skärmdump från programvaran. Som du kan se visar den dig också din balans alla mynt som du har som har getts ut till dig från banken. Eftersom det inte finns något sätt att dela dina mynt, banken ger dig en hel uppsättning mynt i valörer av en cent, två cent, fyra cent,

Sida 11

och så vidare — tvåpotenser. På så sätt kan du (eller din programvara, å dina vägnar) alltid välja en uppsättning mynt för att betala för det exakta beloppet för en transaktion.
När du vill göra en transaktion, säg som i det här exemplet, vill du göra en donation till ideell integritetsgrupp EPIC, klickar du på en donationslänk som tar dig till Digicash-webbplatsen. Det skulle då öppna en omvänd webbanslutning tillbaka till din dator. Det betyder din dator var tvungen att ha förmågan att acceptera inkommande anslutningar och fungera som en server. Du måste ha din din egen IP-adress och din internetleverantör måste tillåta inkommande anslutningar. Om anslutningen var lyckades, då skulle ecash-programvaran starta på din dator och du skulle kunna godkänna transaktion och skicka pengarna.
Chaum hade flera patent på Digicash-teknik, i synnerhet blindsignaturschemat som det Begagnade. Detta var kontroversiellt, och det hindrade andra människor från att utveckla ecash-system som använde samma protokoll. Men ett gäng kryptografer som umgicks med det som kallades cypherpunkarna e-postlista ville ha ett alternativ. Cyperpunks var föregångaren till e-postlistan där Satoshi Nakamoto skulle senare tillkännage Bitcoin för världen, och detta är ingen slump. Vi ska prata om cypherpunk-rörelsen och rötterna till Bitcoin i kapitel 7.
Cypherpunk-kryptograferna implementerade en version av ecash som heter MagicMoney. Det kränkte patenten, men fakturerades som endast för experimentell användning. Det var en rolig mjukvara att spela med. Gränssnittet var helt textbaserat. Du kan skicka transaktioner via e-post. Du skulle bara kopiera och klistra in transaktionerna i din e-post och skicka den till en annan användare. Förhoppningsvis skulle du använda end-to-end e-krypteringsprogram som PGP för att skydda transaktionen under överföring.
Sedan finns det ett förslag som heter Lucre av Ben Laurie med bidrag från många andra människor. Vinning försöker ersätta blindsignaturschemat i ecash med ett icke-patentbelastat alternativ, med resten av systemet i stort sett likadant.
Ännu ett förslag, av Ian Goldberg, försöker lösa problemet med att inte kunna dela upp dina mynt till ändra. Hans idé var att köpmannen kunde skicka tillbaka mynt till dig om de hade några mynt, så att du kan betala för mycket för föremålet om du inte hade exakt växel, och då skulle du få några mynt tillbaka. Men lägg märke till att detta introducerar ett anonymitetsproblem. Som vi såg tidigare, i ecash, är avsändare anonym men handlare är det inte. När handlaren skickar pengar tillbaka, tekniskt sett, är de det avsändaren, så de är anonyma. Men du som måste lämna tillbaka pengarna till banken är det inte anonym. Det finns inget sätt att designa det här systemet utan att bryta anonymiteten för användare som försöker köpa varor. Så Goldberg kom med ett förslag där det fanns olika typer av mynt som skulle tillåta dessa transaktioner att inträffa, låta dig få tillbaka växeln och fortfarande bevara din anonymitet.
Varför misslyckades DigiCash? Det största problemet med DigiCash var att det var svårt att övertala banker och köpmän att anta det. Eftersom det inte var många handlare som accepterade ecash, så har användare ville inte det heller. Ännu värre, det stödde inte transaktioner från användare till användare, eller åtminstone inte särskilt bra. Det var verkligen centrerad på transaktionen mellan användare och handlare. Så om köpmän inte var ombord så fanns det

Sida 12

inget annat sätt att starta intresset för systemet. Så i slutet av dagen förlorade DigiCash och krediten

kortbolag vann.
Som en sidoanteckning tillåter Bitcoin transaktioner från användare till handlare och användare till användare. Faktum är att protokollet
har inte ett begrepp om säljare som är skilt från begreppet användare. Stödet för
transaktioner från användare till användare bidrog förmodligen till Bitcoins framgång. Det var något att göra med dina bitcoins redan från början: skicka det till andra användare, medan communityn försökte slå till
stöd för Bitcoin och få handlare att acceptera det.
Under företagets senare år experimenterade DigiCash också med manipuleringssäker hårdvara för att prova att *förhindra* dubbel utgifter snarare än att bara upptäcka det. I det här systemet får du en liten hårdvara enhet som vanligtvis kallades en plånbok, eller något slags kort. Enheten skulle hålla reda på din saldo, som skulle minska när du spenderade pengar och öka om du laddade kortet med mer pengar. Poängen med enheten är att det inte ska finnas något sätt att fysiskt eller digitalt gå in och manipulera med dess disk. Så om räknaren når noll, så slutar kortet att kunna spendera pengar tills den laddas om.
Det fanns många andra företag som hade elektroniska kassasystem baserade på manipuleringssäkra hårdvara. DigiCash arbetade senare med ett företag som heter CAFE som var baserat i Europa. Annan företag som bildades kring denna idé hette Mondex och det förvärvades senare av Mastercard. Visum hade även en egen variant som heter VisaCash.
Bild 3: Mondex-system, som visar användarkort och plånbok.
12

Figur 3 visar användarsidan av Mondex-systemet. Det finns ett smartkort och det finns en plånboksenhet, och du kan ladda någon av dem med kontanter. Och om du ville byta pengar från användare till användare, givaren skulle först lägga sitt kort i plånboken och flytta pengar från kortet till plånboken.
Sedan skulle mottagaren stoppa sitt kort i plånboken och sedan flytta pengarna till det andra kort. Detta var ett sätt att växla digitala kontanter, och det var anonymt.
Mondex testade sin teknik i ett gäng samhällen. Ett samhälle råkade vara en stad
mycket nära där jag växte upp: Guelph, Ontario. Du har säkert redan gissat att det inte riktigt gjorde det ta tag i. Ett stort problem med Mondex-kort är att de är som kontanter — om du tappar dem eller om de får det stulen, pengarna är borta. Ännu värre, om det finns någon form av fel med kortet, om kortläsaren
inte skulle läsa det, det finns inget sätt att ta reda på om det var saldo på det kortet eller inte. I dessa scenarier, Mondex skulle vanligtvis äta kostnaden. De skulle anta att kortet var laddat och ersätta användaren
för de förlorade pengarna. Det kan naturligtvis kosta ett företag mycket pengar.
Dessutom var plånboken långsam och klumpig. Det gick mycket snabbare att betala med kreditkort eller kontant. Och
återförsäljare hatade att ha flera betalterminaler; de ville bara ha ett för kreditkort. Alla dessa
faktorer tillsammans gjorde Mondex in.
Dessa kort var dock smarta kort, vilket betyder att de har små mikrokontroller på sig,
och att tekniken har visat sig vara framgångsrik. I många länder idag, inklusive Kanada, där jag bor, varje enskilt kreditkort och varje enskilt betalkort har nu smartkortsteknik i sig. Den används för en dock ett annat syfte. Det används inte för att förhindra dubbla utgifter – problemet uppstår inte
eftersom det inte är en kontantbaserad teknik. Banken, snarare än ditt kort, håller reda på ditt saldo eller Tillgänglig kredit. Istället används chippet för autentisering, det vill säga för att bevisa att du känner till PIN-koden som är kopplat till ditt konto. Men Mondex använde det långt innan denna teknik var det
allmänt antagen av banksektorn.

Att slå ut pengar ur luften

I DigiCash-systemet, om du har ett digitalt kontantobjekt som är värt $100, vad gör det egentligen värd $100? Svaret är enkelt: för att få ecash värd 100 USD måste du ta ut 100 USD

av ditt bankkonto och ge det till banken som gav dig e-kassan. Men det fanns ett gäng olika förslag på hur man gör detta och olika företag gjorde det olika. En långsökt möjlighet: tänk om regeringen i ett visst land faktiskt godkände tjänster att skapa digitala pengar, skapa nya pengar ur tomma luften? Det var tanken bakom NetCash, även om den aldrig blev det bortom förslagsstadiet. Ett annat system, som användes av e-Gold, var att lägga en hög med guld i ett valv och att ge ut digitala kontanter endast upp till guldets värde. Ett annat företag som heter Digigold var inte helt uppbackad av guld, men hade partiella reserver.

Alla dessa idéer kopplar i slutändan värdet av digitala kontanter till dollarn eller en vara. Om dollarn är värdet går upp eller ner, kommer värdet på dina digitala pengar att förändras tillsammans med det. En radikalt

---

En annan möjlighet är att låta digitala pengar vara sin egen valuta, utfärdad och värderad oberoende av någon annan valuta.

För att skapa en fritt flytande digital valuta som sannolikt kommer att få verkligt värde måste du ha något som är ont om design. Faktum är att knapphet också är anledningen till att guld eller diamanter har varit används som stöd för pengar. I den digitala sfären är ett sätt att uppnå knapphet att designa systemet så att prägla pengar kräver att man löser ett beräkningsproblem (eller "pussel") som det tar ett tag att spricka. Detta är vad som händer i Bitcoin "mining", som vi kommer att titta på i kapitel 5.

Grundidén — att lösningar på beräkningspussel kan vara digitala objekt som har några värde — är ganska gammal. Det föreslogs först av kryptograferna Dwork och Naor som en potentiell lösning att e-posta skräppost redan 1992. Tänk om din dator skulle behöva göra det varje gång du skickade ett e-postmeddelande lösa ett av dessa pussel som skulle ta några sekunder att lösa? För att upprätthålla detta krav måste mottagarens e-postprogram skulle helt enkelt ignorera din e-post som du inte bifogade lösningen till beräkningspussel. För den genomsnittliga användaren skulle det inte vara så mycket av ett hinder för att skicka e-post eftersom du inte skickar e-post så ofta. Men om du är en spammare, försöker du skicka ut tusentals eller miljontals e-postmeddelanden på en gång, och att lösa dessa beräkningspussel kan bli hindrande. En liknande idé upptäcktes senare oberoende av Adam Back 1997 i ett förslag kallas Hashcash.

Dessa beräkningspussel måste ha vissa specifika egenskaper för att vara ett användbart spamavskräckande medel. För det första borde det vara omöjligt för en spammare att lösa ett pussel och bifoga lösningen till varje e-postmeddelande han skickar. För att säkerställa detta bör pusslet vara specifikt för e-postmeddelandet: det bör bero på avsändaren och mottagare, innehållet i e-postmeddelandet och den ungefärliga tidpunkt då det skickas. För det andra mottagaren ska enkelt kunna kontrollera pussellösningen utan att behöva upprepa processen lösa pusslet. För det tredje bör varje pussel vara helt oberoende av de andra, i den meningen att Att lösa ett pussel minskar inte tiden det tar att lösa något annat pussel. Till sist, eftersom hårdvaran förbättras med tiden och att lösa ett givet beräkningspussel blir snabbare och billigare bör mottagarna kunna anpassa svårighetsgraden på de pussellösningar som de accepterar.

Dessa egenskaper kan uppnås genom att använda kryptografiska hashfunktioner för att designa pussel, och vi ska studera detta i kapitel 1.

Bitcoin använder i princip samma beräkningspussel som Hashcash, men med några mindre förbättringar. Bitcoin gör mycket mer än Hashcash gör, trots allt, det tar en hel bok att förklara Bitcoin! Jag nämner bara detta eftersom Hashcash-uppfinnaren Adam Back har sagt, "Bitcoin är Hashcash utökas med inflationskontroll." Jag tycker att det är lite överdrivet. Det är ungefär som att säga, "a Tesla är bara ett batteri på hjul."

Som med alla bra idéer inom kryptografi finns det många varianter av beräkningspussel som syftar till uppnå lite olika egenskaper. Ett förslag kommer från Rivest och Shamir, R och S in

RSA-kryptosystemet. Observera att i Hashcash är din kostnad för att lösa ett antal pussel helt enkelt
summan av de individuella kostnaderna, genom design. Men detta skiljer sig från kostnadsstrukturen för en
regering
mynta pengar. Om du tänker på hur anti-förfalskning teknik i en pappersvaluta, det finns en enorm

---

initial kostnad för att skaffa all utrustning, skapa säkerhetsfunktioner och så vidare. Men när de väl har gjort det gjort allt det, deras kostnader minskar, och det spelar ingen roll om de skriver ut en eller hundra räkningar.
Att prägla papperspengar har med andra ord en enorm fast kostnad men låg marginalkostnad. Rivest och Shamir ville designa beräkningspussel som skulle efterlikna dessa egenskaper, så att prägla den första
mynt är enormt beräkningsmässigt utmanande, men att prägla efterföljande mynt är mycket billigare. Deras förslaget använde också hashfunktioner, men på ett annat sätt. Vi kommer inte att gå in på detaljerna om deras lösning, men problemet de försökte lösa är intressant på hög nivå.
Varför kom Hashcash aldrig ikapp för sitt avsedda syfte att förhindra spam? Kanske bara spam
var inte ett tillräckligt stort problem att lösa. För de flesta människor är spam ett besvär, men inte något som de vill spendera sina datorcykler på att bekämpa. Vi har spamfilter idag som fungerar
ganska bra på att hålla skräppost borta från våra inkorgar. Det är också möjligt att Hashcash faktiskt inte skulle ha gjort det
stoppade spammare. I synnerhet skickar de flesta spammare idag sin skräppost med hjälp av "botnät", stora grupper
av andras datorer som de tar kontroll över med skadlig programvara. De kan lika gärna använda dessa datorer för att skörda hashcash. Som sagt, idén att använda beräkningspussel för att begränsa tillgång till resurser är fortfarande en idé som håller på att spridas. Du kan se det i några förslag på ersättning nätverksprotokoll, såsom MinimaLT.

Registrera allt i en reskontra

En annan nyckelkomponent i Bitcoin är blockkedjan: en huvudbok där alla Bitcoin-transaktioner finns säkert registrerat. Idéerna bakom blockkedjan är återigen ganska gamla, och går tillbaka till ett papper av Haber och Stornetta 1991. Deras förslag var en metod för säker tidsstämpling av digitalt
dokument, snarare än ett system med digitala pengar. Målet med tidsstämpling är att ge en ungefärlig idé om när ett dokument kom till. Ännu viktigare, tidsstämpling förmedlar korrekt
ordningen för skapande av dessa dokument: om det ena kom till före det andra,
tidsstämplar återspeglar det. Säkerhetsegenskapen kräver att ett dokuments tidsstämpel inte kan vara det ändrats i efterhand.
I Haber och Stornettas system finns en tidsstämplingstjänst som kunder skickar dokument till
tidsstämpel. När servern tar emot ett dokument signerar den dokumentet tillsammans med det aktuella tid och samt en länk eller en pekare till föregående dokument, och utfärdar ett "certifikat" med detta information. Pekaren i fråga är en speciell typ av pekare som länkar till en databit istället för en plats. Det betyder att om informationen i fråga ändras blir pekaren automatiskt ogiltig. I
I kapitel 1 kommer vi att studera hur vi kan skapa sådana pekare med hjälp av hashfunktioner.
Vad detta uppnår är att varje dokuments certifikat säkerställer integriteten hos innehållet i föregående dokument. Faktum är att du kan tillämpa detta argument rekursivt: varje certifikat fixar i princip hela historiken för dokument och certifikat fram till dess. Om vi antar att varje kund i
systemet håller reda på åtminstone ett fåtal certifikat — deras egna dokuments certifikat och de av

---

föregående och följande dokument — då kan deltagarna tillsammans säkerställa att historiken kan inte ändras i efterhand. I synnerhet bevaras den relativa ordningen av dokument.

Figur 4: länkad tidsstämpling . För att skapa ett certifikat för ett dokument innehåller tidsstämpelservern en hash-pekare till föregående dokuments certifikat, aktuell tid, och signerar dessa tre data element tillsammans.

En senare artikel föreslog en effektivitetsförbättring: istället för att länka dokument individuellt kan vi samla dem i block och länka ihop blocken i en kedja. Inom varje block skulle dokumenten återigen länkas samman, men i en trädstruktur istället för linjärt. Detta minskar mängden kontroll som behövs för att verifiera att ett visst dokument förekommer vid en viss punkt i historien systemet. Visuellt ser detta hybridschema ut som i figur 5.

Figur 5: effektiv länkad tidsstämpling . Pilar representerar hash-pekare och prickade vertikala linjer ange tidsintervall.

Denna datastruktur utgör skelettet av Bitcoins blockkedja, som vi kommer att se i kapitel 3. Bitcoin förfinar det på ett subtilt men viktigt sätt: ett Hashcash-liknande protokoll används för att fördröja hur snabbt nya block läggs till i kedjan. Denna modifiering har djupgående och gynnsamma konsekvenser för Bitcoins säkerhetsmodell. Det finns inte längre behov av betrodda servrar; i stället registreras händelser av en samling av opålitliga noder som kallas "gruvarbetare". Varje gruvarbetare håller reda på block, snarare än att behöva lita på att vanliga användare gör det. Vem som helst kan bli en gruvarbetare genom att lösa beräkningspussel för att skapa block. Bitcoin blir också av med signaturer och förlitar sig bara på hash-pekare för att säkerställa integriteten hos datastruktur. Slutligen är de faktiska tidsstämplarna inte av stor betydelse i Bitcoin, och poängen med

---

systemet ska registrera den relativa ordningen av transaktioner på ett manipuleringssäkert sätt. Faktum är att Bitcoin block skapas inte i ett fast schema. Systemet ser till att en ny skapas var 10:e minuter i genomsnitt, men det finns avsevärd variation i tiden mellan på varandra följande block.

I huvudsak kombinerar Bitcoin idén att använda beräkningspussel för att reglera skapandet av nya valutaenheter med idén om säker tidsstämpling för att registrera en redovisning av transaktioner och förhindra dubbla utgifter. Det fanns tidigare, mindre sofistikerade förslag som kombinerade dessa två idéer. De första kallas b-money, och det var av Wei Dai 1998. I b-money kan vem som helst skapa pengar med hjälp av en haschliknande system. Det finns ett peer-to-peer-nätverk, ungefär som i Bitcoin. Varje nod upprätthåller en huvudbok, men det är inte en global huvudbok som i Bitcoin-blockkedjan. Varje nod har sin egen reskontra över vad det tror att allas balans är.

Ett annat liknande förslag, av Nick Szabo, heter Bitgold. Szabo säger att han hade idén till Bitgold as tidigt som 1998, men kom inte igång med att blogga om det förrän 2005. Anledningen till att jag nämner detta är att det finns en mindre konspirationsteori som populariserats av Nathaniel Popper, en New York Times reporter som skrev en mycket bra bok om Bitcoins historia. Popper noterar att blogginläggets tidsstämplar var ändrats efter att Satoshi publicerade Bitcoin Whitepaper så att Bitgold-förslaget ser ut som det var skrevs upp ungefär två månader efter att Bitcoin släpptes. Popper tror, liksom många andra observatörer, att Szabo kan vara Satoshi, och han citerar tidsstämpeländringen som bevis på att Szabo/Satoshi försöker dölja det faktum att han uppfann Bitgold innan han visste om Bitcoin.

Problemet med denna förklaring är att om du faktiskt läser innehållet i blogginläggen så är Szabo det mycket tydlig med att ha haft den här idén 1998, och han försöker inte ändra dessa datum. Alltså ett mer rimlig förklaring är att han precis stötte inlägget till toppen av sin blogg efter att Bitcoin populariserades liknande idéer, för att se till att folk var medvetna om hans tidigare förslag.

Bitcoin har flera viktiga skillnader från b-money och Bitgold. I dessa förslag

beräkningspussel används direkt för att skapa valuta. Vem som helst kan lösa ett pussel och lösningen är en enhet av pengar i sig. I Bitcoin utgör pussellösningar i sig inte pengar. De är använda för att säkra blockkedjan, och endast indirekt leda till att prägla pengar under en begränsad tid. Andra, b-money och Bitgold förlitar sig på tidsstämplingstjänster som undertecknar skapandet eller överföringen av pengar.

Bitcoin, som vi har sett, kräver inte pålitlig tidsstämpling, utan försöker bara bevara den relativa ordning av block och transaktioner.

Slutligen, i b-money och Bitgold, om det råder oenighet om huvudboken mellan servrarna eller noderna, det finns inget tydligt sätt att lösa det. Att låta majoriteten bestämma verkar vara implicit i båda författarnas skrifter. Men eftersom vem som helst kan sätta upp en nod - eller hundra, gömmer sig bakom olika identiteter - dessa mekanismer är inte särskilt säkra, såvida det inte finns en centraliserad gatekeeper som kontrollerar inträdet nätverket. I Bitcoin, däremot, för att en angripare ska kunna ändra historien måste de lösa beräkningar pussel i snabbare takt än resten av deltagarna tillsammans. Detta är inte bara säkrare, det tillåter oss att kvantifiera systemets säkerhet.

17

B-money och Bitgold var informella förslag — b-money var ett inlägg på en e-postlista och Bitgold var en rad blogginlägg. Varken tog fart eller genomfördes ens direkt. Till skillnad från den vita Bitcoin papper, det fanns inte en fullständig specifikation eller någon kod. Förslagen spolar över frågor som kan eller kan inte vara lösbar. Den första, som vi redan har nämnt, är hur man löser meningsskiljaktigheter om huvudbok. Ett annat problem är att bestämma hur svårt beräkningspusslet ska vara för att det ska kunna göras mynta en valutaenhet. Eftersom hårdvara tenderar att bli dramatiskt billigare med tiden för en fast mängden datorkraft, innehåller Bitcoin en mekanism för att automatiskt justera svårighetsgraden av pusslen med jämna mellanrum. B-money och Bitgold inkluderar inte en sådan mekanism, vilket kan resultera i problem eftersom mynt kan förlora sitt värde om det blir trivialt enkelt att skapa nya.

Tips om Satoshi

Du kanske vet att Satoshi Nakamoto är pseudonymen som antogs av skaparen av Bitcoin. Medan hans identitet förblir ett mysterium, han kommunicerade flitigt i Bitcoins tidiga dagar. Låt oss använda detta för att gräva en lite in på frågor som när han började arbeta med Bitcoin, i vilken utsträckning han påverkades av de tidigare idéerna vi har tittat på och vad som motiverade honom.

Satoshi säger att han började koda Bitcoin runt maj 2007. Jag tar honom på ordet; det faktum att han är anonym är inte en anledning att tro att han skulle ljuga om sådana saker. Han registrerade domänen bitcoin.org i augusti 2008. Och vid den tiden började han skicka privata e-postmeddelanden till några personer som han tänkte vara intresserad av förslaget. Sedan lite senare i oktober 2008 släppte han offentligt en vitbok som beskrev protokollet, och kort därefter släppte han den första koden för Bitcoin också. Sedan satt han kvar i ungefär två år, under vilka han postade massor av meddelanden på forum, mailade med massor av människor och svarade på folks oro. På programmeringssidan, han skickade patchar till koden. Han behöll källkoden tillsammans med andra utvecklare, åtgärda problem när de uppstod. I december 2010 hade andra långsamt tagit över underhåll av projektet, och han slutade kommunicera med dem.

Jag har refererat till Satoshi Nakamoto som en "han", men jag har ingen speciell anledning att tro att Satoshi är en man och inte en kvinna. Jag använder bara det manliga pronomenet eftersom Satoshi är ett mansnamn. Det har jag också varit hänvisar till honom som en enskild individ. Det finns en teori om att Satoshi Nakamoto kan vara en samling av individer. Jag köper inte den här teorin – jag tror att Satoshi förmodligen bara är en person. Anledningen är det om vi tittar på hela onlineinteraktionerna som genomförs under Satoshi-pseudonymen, om vi

tänk på de två år som Satoshi ägnade åt att svara på e-postmeddelanden och patcha kod, det är svårt att föreställa sig
att det här kan vara flera personer som delar användarkonton och lösenord och svarar på liknande sätt
och en liknande röst, och se till att de inte motsäger varandra. Det verkar bara mycket enklare
förklaringen att åtminstone denna del av Satoshis aktivitet gjordes av en enda individ.
Dessutom är det tydligt från hans skrifter och patchar att denna person förstod hela kodbasen
av Bitcoin och alla dess designaspekter. Så det är mycket rimligt att anta att samma person skrev
den ursprungliga kodbasen och det vita papperet också. Slutligen är det möjligt att Satoshi hade hjälp med

---

original design. Men efter Bitcoins release kan vi själva se att Satoshi var snabb med
tillskriva all hjälp han fått från andra bidragsgivare. Det skulle vara ur karaktär för honom
vilseleda oss om att hitta på något själv om han hade fått hjälp av andra människor.
Därefter kan vi fråga oss själva: "Vad visste Satoshi om ekas historia?" Att förstå
detta bättre, vi kan börja med att titta på vad han citerar i sin vitbok samt referenserna som
fanns på tidiga versioner av Bitcoin-webbplatsen. I vitboken citerar han några papper om grundläggande
kryptografi och sannolikhetsteori. Han citerar också tidsstämplingsarbetet som vi såg tidigare, och
det är mycket naturligt att tro att han baserade designen av blockkedjan på dessa referenser sedan
likheterna är så uppenbara. Han citerar också Hashcash-förslaget vars beräkningspussel är mycket
liknande den som används i Bitcoin. Han har också en referens till b-pengar. Senare, på hemsidan, han
lagt till referenser till Bitgold och även till ett schema av Hal Finney för återanvändning av beräkningspussel
lösningar.
Men om vi tittar på e-postutbytena som offentliggjordes av personer som korresponderade med
Satoshi Nakamoto i början, finner vi att förslaget om b-pengar faktiskt lades till
efterhand, på förslag av Adam Back. Satoshi mailade sedan Wei Dai som skapade b-pengar
och tydligen var Dai den som berättade för honom om Bitgold. Så dessa förslag var förmodligen inte det
inspiration till den ursprungliga designen. Senare korresponderade han mycket med Hal Finney, och det är ganska bra
rimlig förklaring till varför han citerar Finneys verk, åtminstone på hemsidan.
Baserat på detta verkar det troligt att när man skapade Bitcoin var Hashcash och tidsstämpling
bara saker från ekas historia som Satoshi visste om eller trodde var relevanta. Efter han
fick reda på b-money och Bitgold, men han verkar ha uppskattat deras relevans. I
mitten av 2010, Wikipedia-artikeln om Bitcoin flaggades för radering Wikipedias redaktörer eftersom de
tyckte det inte var anmärkningsvärt. Så det blev en del diskussion mellan Satoshi och andra om hur
att formulera artikeln så att Wikipedia skulle acceptera den. För detta ändamål föreslog Satoshi denna beskrivning
av Bitcoin: "Bitcoin är en implementering av Wei Dais förslag om b-pengar på Cypherpunks 1998 och
Nick Szabos Bitgold-förslag." Så vid det här laget såg Satoshi positionering av Bitcoin som en förlängning av
dessa två idéer eller en implementering av dessa två tidigare system som en bra förklaring av hur det
arbetade.
Men vad sägs om allt annat - Chaumian ecash system och kreditkortsförslagen som
vi tittade på? Kände Satoshi till något av den historien när han designade Bitcoin? Det är svårt att säga. Det gjorde han inte
ge någon indikation på att han känner till den historien, men det är lika troligt att han inte refererade till detta pga
det var inte relevant för Bitcoin. Bitcoin använder en helt annan decentraliserad modell och så det finns ingen
övertygande anledning att uppehålla sig vid gamla centraliserade system som misslyckades.
Satoshi själv påpekar detta, genom att nämna Chaumian ecash i förbigående, i ett av sina inlägg till
Bitcoin-forum. Han skrev om ett annat förslag som heter opencoin.org och noterar att de verkar vara det
”Pratar om de gamla chaumianska centralmyntgrejerna, men kanske bara för att det var det enda
tillgängligt. Kanske skulle de vara intresserade av en ny riktning. Många människor avfärdar automatiskt

e-valuta som en förlorad sak på grund av alla företag som misslyckats sedan 1990-talet. Jag hoppas att det är uppenbart

---

det var bara den centralt kontrollerade naturen hos dessa system som dömde dem. Jag tror att detta är den första gång vi försöker ett decentraliserat, icke-förtroendebaserat system." Det ger oss en ganska bra uppfattning om vad Satoshi tänkte på de tidigare förslagen, och specifikt hur han kände att Bitcoin var annorlunda. Bitcoins decentralisering är verkligen en avgörande egenskap som skiljer den från nästan allt vi har tittat på.

Ett annat intressant citat från Satoshi antyder att han kanske inte är en akademiker. Mest akademiskt forskare funderar över idéer och skriver ner dem direkt, innan de bygger systemet.

Satoshi säger att han tog en motsatt inställning: "Jag gjorde faktiskt Bitcoin typ baklänges. jag var tvungen skriva all kod innan jag kunde övertyga mig själv om att jag kunde lösa alla problem, sedan skrev jag papper. Jag tror att jag kommer att kunna släppa koden snabbare än jag kunde skriva en detaljerad specifikation."

Eftersom det finns lite myter kring Satoshi är det värt att nämna att han gjorde misstag som alla andra annat och det var inte ett perfekt framtidsorakel. Det finns buggar och tvivelaktiga designval i den ursprungliga Bitcoin-koden såväl som i dess design. Det fanns till exempel en funktion att skicka bitcoins till IP-adresser som aldrig hängde med och i efterhand var en dålig idé. När han beskrev vad Bitcoin var användbar för, var hans scenarier centrerade på idén att använda den över internet. Det användningsfallet är centralt för Bitcoin, naturligtvis, men det är inte den enda. Han antydde inte en vision om att gå in på en kaffe shoppa och att kunna betala sitt kaffe med till exempel Bitcoin.

En sista fråga kan vi ställa oss själva, färgad av vad vi förstår från den digitala historien kontanter är, "Varför behåller Satoshi sin anonymitet?" Det finns många möjliga orsaker. Till att börja med, det kanske bara är för skojs skull. Många skriver romaner anonymt, och det finns graffitikonstnärer som Banksy som bibehåller sin anonymitet. I själva verket i samhället som Satoshi var involverad i då tid, Cypherpunk-gemenskapen och e-postlistan för kryptografi, var det vanlig praxis för personer att posta anonymt.

Å andra sidan kunde det ha legat juridiska bekymmer bakom Satoshis val. Två amerikanska företag, Liberty Reserve och e-Gold hamnade i juridiska problem för penningtvätt. 2006, en av grundarna of Liberty Reserve flydde från USA, fruktade att han skulle åtalas för penningtvätt kostnader. E-Golds grundare däremot stannade i USA, och det var en faktiskt åtalades och erkände sig så småningom skyldig till anklagelserna. Denna erkännande registrerades precis innan Satoshi startade Bitcoin-webbplatsen och började maila folk om sitt förslag. Som sagt, många människor har uppfunnit ecash-system, och ingen annan var rädd för de juridiska konsekvenserna eller har valt att vara anonym. Så detta kan ha varit anledningen, det kanske inte var det anledning.

Det är också värt att komma ihåg att vissa aspekter av ecash patenterades, och att medlemmarna i Cypherpunk-rörelsen var oroade över att implementera ecash-system på grund av dessa patent. I Faktum är att ett inlägg till cypherpunks e-postlista föreslog att en grupp anonyma kodare skulle implementera ecash så att om någon skulle stämma så skulle de inte kunna hitta kodarna. Även om det är svårt att tror att Bitcoin skulle bryta mot ecash-patenten med tanke på hur olika designen är, kanske Satoshi var extra försiktig. Eller så var han bara inspirerad av idén om en anonym kodare från cypherpunk community.

Ett sista skäl som ofta nämns är personlig säkerhet. Vi vet att Satoshi har många bitcoins från hans gruvdrift tidigt, och på grund av Bitcoins framgångar är dessa nu värda mycket pengar. Jag tror att detta är en rimlig anledning. Att välja att vara anonym är trots allt inte ett beslut du fattar en gång, det är något

som du gör på en kontinuerlig basis. Som sagt, det var förmodligen inte Satoshis ursprungliga anledning. Första gången
Satoshi använde namnet Satoshi Nakamoto, han hade inte ens släppt whitepapern eller kodbasen
för Bitcoin, och det är svårt att föreställa sig att han hade någon aning om att det skulle bli så framgångsrikt som det var. I
Faktum är att på många punkter i sin tidiga historia var Satoshi optimistisk men försiktig med Bitcoins utsikter. Han verkar ha förstått att många tidigare försök hade misslyckats och att Bitcoin också kan misslyckas.

Slutord

Framgången för Bitcoin är ganska anmärkningsvärd om du tänker på alla satsningar som misslyckades med att försöka göra vad
det gör det. Bitcoin har flera anmärkningsvärda innovationer inklusive blockkedjan och en decentraliserad modell som stöder transaktioner från användare till användare. Det ger en praktiskt användbar men mindre än perfekt nivå av
anonymitet för användare. I kapitel 6 tar vi en detaljerad titt på anonymitet i Bitcoin. I en mening är det svagare än den starka anonymiteten i DigiCash, men i en annan mening är den starkare. Det är för att i DigiCash, det var bara avsändarna av pengarna som bibehöll sin anonymitet, och inte
köpmän. Bitcoin ger både avsändare och handlare (oavsett om det är användare eller handlare) samma nivå av anonymitet.
Låt mig avsluta med några lärdomar som vi kan lära oss av Bitcoin genom den tidigare linsen
system som vi har tittat på. Det första är att inte ge upp ett problem. Bara för att folk misslyckades
20 år med att utveckla digitala kontanter betyder inte att det inte finns ett system där ute som kommer att fungera. De
andra är att vara villig att kompromissa. Om du vill ha perfekt anonymitet eller perfekt decentralisering kommer du att göra det
behöver förmodligen försämra andra delar av din design. Bitcoin, i efterhand, verkar ha gjort det
rätta kompromisser. Det minskar anonymiteten lite och kräver att deltagarna är online och
kopplade till peer-to-peer-nätverket, men detta visade sig vara acceptabelt för användarna.
En sista lektion är framgång genom siffror. Bitcoin kunde bygga upp en gemenskap av passionerade
användare såväl som utvecklare som är villiga att bidra till tekniken med öppen källkod. Detta är en markant annorlunda tillvägagångssätt än tidigare försök med digitala kontanter, som vanligtvis utvecklades av en
företag, där de enda förespråkarna för tekniken är de anställda på företaget självt.
Bitcoins nuvarande framgång beror till stor del på den livliga stödjande gemenskap som drivit på
teknik, fick folk att använda den och fick köpmän att adoptera den.
21

Vidare läsning

En tillgänglig översikt över digitala kontantsystem fokuserad på praktiska frågor:
P. Wayner. Digital Cash: handel på nätet (2:a upplagan). Morgan Kaufmann, 1997.
En kryptografiskt orienterad översikt över e-kontantsystem (kapitel 1) och mikrobetalningar (kapitel 7):
B. Rosenberg (red.) Handbook of Financial Cryptography and Security. CRC Press, 2011.
Även om det inte är Chaums tidigaste tidning om e-cash, är detta utan tvekan det mest innovativa, och det bildade en
mall som replikeras av många andra tidningar:
D. Chaum, A. Fiat, M. Naor. Ospårbara elektroniska kontanter. CRYPTO 1998.
Många tidningar förbättrade effektiviteten hos Chaum-Fiat-Naor med hjälp av moderna kryptografiska tekniker, men det viktigaste är utan tvekan:
J. Camenisch, S. Hohenberger, A. Lysyanskaya, Compact e-cash. Teori och tillämpningar av Kryptografiska tekniker, 2005

Några praktiska säkerhetsobservationer om finansbranschen och förslag, inklusive Mondex:
R. Andersson. Säkerhetsteknik (2nd ed). Wiley, 2008.
En översikt över implementeringen av Chaums förslag till kontanter:
B. Schoenmakers. Grundläggande säkerhet för kontantbetalningssystemet. State of the Art i tillämpad Kryptografi, 1997.
Två artiklar citerade av Satoshi Nakamoto i Bitcoin Whitepaper som är integrerade i Bitcons design:
En rygg. Hashcash - A Denial of Service Counter-Measure, Online, 2002.
S. Haber, WS Stornetta. Säkra namn för bitsträngar. CCS, 1997.
22

Kapitel 1: Introduktion till kryptografi och kryptovalutor

Alla valutor behöver något sätt att kontrollera utbudet och genomdriva olika säkerhetsegenskaper för att förhindra fusk. I fiat-valutor kontrollerar organisationer som centralbanker penningmängden och lägger till funktioner mot förfalskning till fysisk valuta. Dessa säkerhetsfunktioner höjer ribban för en angripare, men de gör inte pengar omöjliga att förfalska. I slutändan är brottsbekämpning nödvändig för hindra människor från att bryta mot systemets regler.
Även kryptovalutor måste ha säkerhetsåtgärder som hindrar människor från att manipulera staten av systemet, och från tvetydiga, det vill säga göra inbördes inkonsekventa uttalanden till olika människor. Om Alice till exempel övertygar Bob om att hon betalade ett digitalt mynt till honom borde hon inte kunna göra det
övertyga Carol om att hon betalade samma mynt till henne. Men till skillnad från fiat-valutor, säkerhetsreglerna för
kryptovalutor måste genomdrivas rent tekniskt och utan att förlita sig på en central auktoritet.
Som ordet antyder använder kryptovalutor mycket kryptografi. Kryptografi ger en mekanism för säker kodning av reglerna för ett kryptovalutasystem i själva systemet. Vi kan använd den för att förhindra manipulering och oklarheter, samt för att koda reglerna för att skapa nya enheter av valutan till ett matematiskt protokoll. Innan vi riktigt kan förstå kryptovalutor då måste vi fördjupa oss i de kryptografiska grunderna som de litar på.
Kryptografi är ett djupt akademiskt forskningsfält som använder många avancerade matematiska tekniker som är notoriskt subtila och komplicerade att förstå. Lyckligtvis förlitar sig Bitcoin bara på en handfull relativt enkla och välkända kryptografiska konstruktioner. I det här kapitlet ska vi studera specifikt kryptografiska hash och digitala signaturer, två primitiver som visar sig vara mycket användbar för att bygga kryptovalutor. Framtida kapitel kommer att introducera mer komplicerad kryptografik system, såsom nollkunskapsbevis, som används i föreslagna tillägg och modifieringar av Bitcoin.
När vi har lärt oss de nödvändiga kryptografiska primitiven kommer vi att diskutera några av de sätt som de används för att bygga kryptovalutor. Vi kommer att avsluta detta kapitel med några exempel på enkla kryptovalutor som illustrerar några av designutmaningarna som vi måste hantera.

1.1 Kryptografiska hashfunktioner

Den första kryptografiska primitiv som vi behöver för att förstå är en ***kryptografisk hashfunktion*** **.** A ***hashfunktion*** är en matematisk funktion med följande tre egenskaper:

- Inmatningen kan vara vilken sträng som helst av vilken storlek som helst.
- Den ger en utdata med fast storlek. I syfte att göra diskussionen i detta kapitel konkret kommer vi att anta en 256-bitars utdatastorlek. Men vår diskussion gäller för alla utdatastorlek så länge den är tillräckligt stor.
- Det är effektivt beräkningsbart. Intuitivt betyder detta att du kan räkna ut för en given inmatningssträng

Sida 24

ta reda på vad utmatningen av hashfunktionen är inom rimlig tid. Mer tekniskt sett,
beräkna hash av en n -bitars sträng bör ha en löpande tid som är O (n).
Dessa egenskaper definierar en allmän hashfunktion, en som kan användas för att bygga en datastruktur som t.ex som ett hashbord. Vi kommer att fokusera enbart på *kryptografiska* hashfunktioner. För en hashfunktion för att vara kryptografiskt säker kommer vi att kräva att den har följande tre ytterligare egenskaper: (1) kollisionsmotstånd, (2) gömma sig, (3) pusselvänlighet.
Vi kommer att titta närmare på var och en av dessa egenskaper för att få en förståelse för varför det är användbart att ha
en funktion som beter sig så. Den läsare som har studerat kryptografi bör vara medveten om att
Behandlingen av hashfunktioner i den här boken skiljer sig lite från en vanlig kryptografibok. De
I synnerhet pusselvänlighetsegenskap är inte ett allmänt krav för kryptografisk hash
funktioner, men en som kommer att vara användbar för kryptovalutor specifikt.
Property 1. Collision resistens Den första egenskap som vi behöver från en kryptografisk hashfunktion är att den är kollisionsbeständig. En kollision uppstår när två distinkta ingångar producerar samma utsignal. A hashfunktion $H(.)$ är kollisionsbeständig om ingen kan hitta en kollision. Formellt:
Kollision Resistens: En hashfunktion H sägs vara kollision resistent om det är omöjligt att hitta två värden, x och y , så att $x \neq y$, ännu $H(x) = H(y)$.

Figur 1,1 En hash kollision. X och y är distinkta värden, men när input i hashfunktion H , de producera samma produktion.
Lägg märke till att vi sagt *ingen kan hitta* en kollision, men vi inte säga att inga kollisioner förekommer. Egentligen, vi
vet med säkerhet att kollisioner existerar, och vi kan bevisa detta med ett enkelt räkneargument. De inmatningsutrymme till hash-funktionen innehåller alla strängar av alla längder, men utdatautrymmet innehåller bara
strängar av en viss fast längd. Eftersom ingångsutrymmet är större än utmatningsutrymmet (det är faktiskt inmatningsutrymmet är oändligt, medan utmatningsutrymmet är ändligt), måste det finnas inmatningssträngar som mappar till
samma utgångssträng. Faktum är att det enligt Pigeonhole-principen nödvändigtvis kommer att finnas ett mycket stort antal
av möjliga ingångar som mappar till en viss utgång.

Sida 25

Figur 1,2 Eftersom antalet ingångar överstiger antalet utgångar, vi garanterat att det måste finnas minst en utgång till vilken hashfunktionen mappar mer än en ingång.

Nu, för att göra saken ännu värre, sa vi att det måste vara omöjligt att hitta en kollision. Ändå finns det metoder som garanterat hittar en kollision. Överväg följande enkla metod för att hitta en kollision för en hash-funktion med en 256-bitars utgående storlek: pick $2^{256} + 1$ distinkta värden, beräkna
hash för var och en av dem, och kontrollera om det finns två utgångar som är lika. Eftersom vi plockade mer ingångar än möjliga utgångar, måste ett par av dem kollidera när du använder hash-funktionen.
Metoden ovan kommer garanterat att hitta en kollision. Men om vi väljer slumpmässiga indata och beräknar

hash värden, hittar vi en kollision med hög sannolikhet långt innan undersöka 2^{256} + 1 ingångar. Faktum är att om vi slumpmässigt väljer bara två 2^{130} + 1 ingångar, visar det sig att det finns en 99,8% chans att minst två av dem kommer att kollidera. Det faktum att vi kan hitta en kollision genom att bara undersöka ungefär kvadratroten av antalet möjliga utgångar resulterar från ett fenomen i sannolikhets känd som ***födelsedagen paradox*** . I läxfrågorna i slutet av detta kapitel kommer vi att undersöka detta mer i detalj.

Denna kollisionsdetekteringsalgoritm fungerar för varje hashfunktion. Men det är såklart problemet med det att detta tar väldigt, väldigt lång tid att göra. För en hashfunktion med en 256-bitars utgång skulle du ha att beräkna hashfunktionen 2^{256} + 1 gånger i värsta fall, och ca 2^{128} gånger i genomsnitt. Det är naturligtvis ett astronomiskt stort antal — om en dator beräknar 10 000 hash per sekund, skulle ta mer än en octillion (10^{27}) År för att beräkna 2^{128} hash! För ett annat sätt att tänka om detta kan vi säga att om varje dator som någonsin gjorts av mänskligheten var datorer sedan början av hela universum, fram till nu, är oddsen att de skulle ha hittat en kollision fortfarande oändligt liten. Så liten att det är mycket mindre än oddsen att jorden kommer att förstöras av en jättemeteor inom de kommande två sekunderna.

Vi har alltså sett en generell men opraktiskt algoritm för att hitta en kollision för *någon* hashfunktion. A svårare fråga är: finns det någon annan metod som kan användas på en viss hash funktion för att hitta en kollision? Med andra ord, även om den generiska kollisionsdetekteringsalgoritmen inte är genomförbart att använda, kan det fortfarande finnas någon annan algoritm som effektivt kan hitta en kollision för en specifik hashfunktion.

Tänk till exempel på följande hashfunktion:

$$H(x) = x \bmod 2^{256}$$

Den här funktionen uppfyller våra krav på en hashfunktion eftersom den accepterar indata av valfri längd, returnerar a utdata med fast storlek (256 bitar) och är effektivt beräkningsbar. Men denna funktion har också en effektiv metod för att hitta en kollision. Observera att den här funktionen bara returnerar de sista 256 bitarna av ingången. Ett kollision skulle då vara de värden 3 och $3 + 2^{256}$. Detta enkla exempel illustrerar att även om vår Generisk kollisionsdetekteringsmetod är inte användbar i praktiken, det finns åtminstone några hashfunktioner för som det finns en effektiv kollisionsdetektionsmetod.

Men för andra hashfunktioner vet vi inte om sådana metoder finns. Vi misstänker att de är kollision resistenta. Det finns dock inga hash funktioner *visat* sig vara kollision beständig. Den kryptografiska hashfunktioner som vi förlitar oss på i praktiken är bara funktioner som folk verkligen har försökt med svårt att hitta kollisioner och har ännu inte lyckats. I vissa fall, som den gamla MD5-hashfunktionen,

kollisioner hittades så småningom efter år av arbete, vilket ledde till att funktionen försvann och fasas ut ur praktisk användning. Och så vi väljer att tro att de är kollisionsbeständiga.

Ansökan: Meddelande smälter Nu när vi vet vad kollisions motståndet är, är den logiska frågan: Vad är kollisionsmotstånd användbart för? Här är ett program: Om vi vet att två ingångar x och y till en kollisionsresistenta hashfunktion $\boldsymbol{H}$ är olika, då är det säkert att anta att deras hashes $\boldsymbol{H}$ (x) och $\boldsymbol{H}$ (y) är olika - om någon visste en x och y som var olika men hade samma hash, att skulle bryta vårt antagande att $\boldsymbol{H}$ är kollision motståndskraftig.

Detta argument ger oss möjlighet att använda hash utgångar som ***meddelande smälta*** . Överväg SecureBox, en autentiserat online-fillagringssystem som tillåter användare att ladda upp filer och säkerställa deras integritet när de laddar ner dem. Anta att Alice laddar upp riktigt stora filer och vill kunna verifiera senare att filen hon laddar ner är densamma som den hon laddar upp. Ett sätt att göra det skulle vara att spara hela den stora filen lokalt och jämför den direkt med filen hon laddar ner. Även om detta fungerar, det motverkar till stor del syftet med att ladda upp det i första hand; om Alice behöver ha tillgång till en lokal kopia av filen för att säkerställa dess integritet, kan hon bara använda den lokala kopian direkt.

Kollisionsfria hash ger en elegant och effektiv lösning på detta problem. Alice behöver bara kom ihåg den ursprungliga filens hash. När hon senare laddar ner filen från SecureBox, beräknar hashen för den nedladdade filen och jämför den med den hon lagrade. Om hasharna är samma sak, då kan hon dra slutsatsen att filen verkligen är den hon laddade upp, men om de är olika, då kan Alice dra slutsatsen att filen har manipulerats. Att komma ihåg hashen tillåter henne alltså för att upptäcka *misstag* korruption av filen under överföring eller SecureBox servrar, men också *avsiktlig* modifiering av filen från servern. Sådana garantier inför potentiellt skadliga beteende från andra enheter är kärnan i vad kryptografi ger oss.

Hashen fungerar som en sammanfattning med fast längd, eller entydig sammanfattning, av ett meddelande. Detta ger oss en mycket
effektivt sätt att komma ihåg saker vi sett förut och känna igen dem igen. Medan hela filen kan ha varit gigabyte lång, hashen är av fast längd, 256-bitar för hashfunktionen i vår exempel. Detta minskar vårt lagringsbehov avsevärt. Senare i detta kapitel och genom hela bok kommer vi att se applikationer för vilka det är användbart att använda en hash som ett meddelandesammandrag.

Sida 27

Property 2: Dölja Den andra egenskapen som vi vill från våra hashfunktioner är att det är ***gömställe*** . De gömmer egendom hävdar att om vi med tanke på utgången av hashfunktionen $y = H(x)$, det finns ingen möjlig sätt att räkna ut vad ingången, X , var. Problemet är att den här egenskapen inte kan vara sann i det angivna form. Tänk på följande enkla exempel: vi ska göra ett experiment där vi slår ett mynt. Om resultatet av myntvändningen var huvuden, vi kommer att tillkännage hashen för strängen "huvuden". Om resultatet blev svansar, vi kommer att tillkännage hashen för strängen "svansar".

Vi ber sedan någon, en motståndare, som inte såg myntet vända, utan bara såg denna hash-utgång, att ta reda på vad strängen var som hashades (vi kommer snart att se varför vi kanske vill spela spel som detta). Som svar skulle de helt enkelt beräkna både hashen för strängens "huvuden" och hashen för stränga "svansar", och de kunde se vilken de fick. Och så, på bara ett par steg, kan de ta reda på vad ingången var.

Motståndaren kunde gissa vad strängen var eftersom det bara fanns två möjliga värden på x, och det var lätt för motståndaren att bara prova båda. För att kunna uppnå gömmningen egendom, måste det vara så att det inte finns något värde på x som är särskilt troligt. Det vill säga, X måste väljas från en uppsättning som i någon mening är väldigt spridd. Om x är vald från en sådan uppsättning, denna metod
att prova några värden på x som är särskilt sannolika kommer inte att fungera.

Den stora frågan är: kan vi uppnå gömstället när de värden som vi vill inte kommer

från en spridd uppsättning som i vårt experiment med "huvuden" och "svansar"? Lyckligtvis är svaret ja! Så Kanske kan vi dölja även en ingång som inte är sprids ut genom att sammanlänka det med en annan ingång som *är* Sprid ut. Vi kan nu vara lite mer exakta om vad vi menar med att dölja (den dubbla vertikalen stapel ‖ betecknar sammanlänkning).

. *Gömmer* En hashfunktion H är gömmer om: när ett hemligt värde r är vald från en sannolikhetsfördelning som har *hög min-entropi* , sedan ges $H(r \| x)$ det är omöjligt att finna x .

I informationsteori, ***min-entropi*** är ett mått på hur förutsägbart ett resultat är, och hög min-entropi fångar den intuitiva idén att fördelningen (dvs slumpvariabeln) är väldigt spridd. Vad det specifikt betyder är att när vi samplar från distributionen finns det inget särskilt värde det kommer sannolikt att inträffa. Så, för ett konkret exempel, om r väljs likformigt från bland alla av strängarna som är 256 bitar lång, då någon speciell sträng valdes med sannolikhet $1/2^{256}$, som är en oändligt litet värde.

Användning:. Åtaganden Nu ska vi titta på en tillämpning av gömmer egendom. I synnerhet vad vi vill göra är något som kallas en ***förpliktelse*** . Ett åtagande är den digitala analogen av att ta en värde, försegla det i ett kuvert och lägga ut kuvertet på bordet där alla kan se det. När du gör det har du förbundit dig till vad som finns inuti kuvertet. Men du har inte öppnat det, så även om du har förbundit dig till ett värde förblir värdet en hemlighet för alla andra. Senare, du kan öppna kuvertet och avslöja värdet som du förbundit dig till tidigare.

27

Sida 28

. *Lojalitetssystem* Ett åtagande schema består av två algoritmer:

- **com: = begå (*msg, nonce*)** Den begår funktionen tar ett meddelande och hemligt slump värde, kallat nonce, som input och returnerar ett åtagande.
- **kontrollera (*com, msg, nonce*)** Den kontrollera funktionen tar ett åtagande, nonce, och meddelandet som inmatning. Den returnerar sant om *com* == begå (*msg* , *nonce*) och falskt annars.

Vi kräver att följande två säkerhetsegenskaper gäller:

- ***Hiding*** : Med tanke på *com* , är det omöjligt att hitta *msg*
- ***Bindning*** : Det är omöjligt att hitta två par *(msg, nonce)* och *(msg 'nonce')* sådan att *msg* ≠ *msg '* och begår (*msg, nonce*) == commit (*msg', nonce'*)

Om du vill använda ett system engagemang, måste vi först att generera en slumpmässig *nonce.* Vi tillämpar sedan *begå* funktionen denna nonce tillsammans med *msg* , varvid värdet åtagit sig att, och vi publicerar engagemang *com* . Detta steg är analogt med att lägga det förseglade kuvertet på bordet. Vid ett senare punkt, om vi vill avslöja värdet som de förbundit sig till tidigare, publicerar vi den slumpmässiga nonce som vi använde för att skapa detta åtagande, och meddelandet *, msg* . Nu, vem som helst kan kontrollera att *msg* var verkligen det budskap som åtog sig tidigare. Detta steg är analogt med att öppna upp kuvertet. Varje gång du åta sig ett värde, är det viktigt att du väljer en ny slumpvärde *nonce* . I kryptografi, termen *nonce* används för att hänvisa till ett värde som endast kan användas en gång.

De två säkerhetsegenskaperna dikterar att algoritmerna faktiskt beter sig som att försegla och öppna en kuvert. Först, med tanke på *com* , engagemang, någon som tittar på kuvertet kan inte räkna ut vad meddelandet är. Den andra egenskapen är att den är bindande. Detta säkerställer att när du förbinder dig till vad som är i kuvertet kan du inte ändra dig senare. Det vill säga, det är omöjligt att hitta två olika meddelanden, så att du kan förbinda dig till ett meddelande, och sedan hävda att du har åtagit dig

annan.
Så hur vet vi att dessa två egenskaper håller? Innan vi kan svara på detta måste vi diskutera hur vi faktiskt ska genomföra ett åtagandesystem. Vi kan göra det med en kryptografik hash-funktion. Tänk på följande åtagandeschema:
commit (*msg, nonce)* : = H (*nonce* ‖ *msg*) där *nonce* är en slumpmässig 256-bitarsvärde
För att begå ett meddelande genererar vi en slumpmässig 256-bitars nonce. Sedan sammanfogar vi nonce och meddelandet och returnera hashen av detta sammanlänkade värde som åtagandet. För att verifiera, någon kommer att beräkna samma hash för den nonce de fick sammanlänkade med meddelandet. Och de kommer att kontrollera om det är lika med engagemanget som de såg.
Ta en ny titt på de två fastigheter som vi kräver av våra åtagandesystem. Om vi ersätter instansieringen av *commit* och *kontrollera* såväl som *H (nonce* ‖ *msg)* för *com* , då dessa egenskaper

---

bli:

- ***Hiding*** : Givet H (*nonce* ‖ *msg)* , är det omöjligt att hitta *msg*
- ***Bindning*** : Det är omöjligt att hitta två par *(msg, nonce)* och *(msg', nonce ')* på så sätt att *msg* ≠ *msg'* och H (*nonce* ‖ *msg*) == H (*nonce ' ‖ msg'*)

Den *gömmer* egendom åtaganden är exakt gömmer egendom som vi krävs för vår hash funktioner. Om *nyckeln* valdes som ett slumpmässigt 256-bitars värde sedan gömmer egenskapen säger att om vi hash
sammanlänkningen av *nyckeln* och meddelandet, då är det omöjligt att återhämta sig budskapet från hash produktion. Och det visar sig att den *bindande egenskapen* antyds av kollisionen resistenta egendom [1]
underliggande hashfunktion. Om hashfunktionen är kollisionsbeständig kommer den att vara omöjlig att hitta distinkta värden msg och msg 'så att H (*nonce* ‖ *msg)* = H (*nonce'* ‖ *msg ')* , eftersom sådana värden skulle verkligen vara en kollision.
Därför, om *H* är en hash-funktion som är krocksäkra och dölja, detta åtagande systemet kommer arbete, i den meningen att den kommer att ha nödvändiga säkerhetsegenskaper.

Property 3. Puzzle vänlighet Den tredje säkerhets egendom vi ska behov från hashfunktioner är att de är pusselvänliga. Den här egenskapen är lite komplicerad. Vi kommer först att förklara vad tekniska krav på denna fastighet är och ge sedan en ansökan som illustrerar varför detta egendom är användbar.
Pussel vänlighet. En hashfunktion *H* sägs vara pussel-vänligt om för varje möjlig n-bit-utsignal värde *y* , om k är vald bland en fördelning med hög min-entropi, så är det omöjligt att hitta *x* sådant att H (k ‖ x) = y i tiden signifikant mindre än 2^n .

Intuitivt, vad detta betyder är att om någon vill rikta hash-funktionen för att komma ut till vissa särskilt utgångsvärde *y* , att om det är en del av den ingång som är vald i en lämpligt randomiserad sätt, det är väldigt svårt att hitta ett annat värde som träffar exakt det målet.
Applikations. Sök pussel nu, låt oss betrakta ett program som visar nyttan av detta fast egendom. I denna ansökan kommer vi att bygga en ***söka pussel*** , ett matematiskt problem som kräver att man söker ett mycket stort utrymme för att hitta lösningen. I synnerhet har ett sökpussel nej genvägar. Det vill säga, det finns inget sätt att hitta en giltig lösning än att söka i det stora utrymmet.

[1] De omvända konsekvenserna håller inte. Det vill säga, det är möjligt att du kan hitta kollisioner, men ingen av dem är av bilda H (*nonce* ‖ *msg)* == H (*nonce ' ‖ msg')* . Till exempel, om du bara kan hitta en kollision där två distinkta noncer generera samma åtagande för samma budskap, då är åtagandesystemet fortfarande bindande, men

underliggande hash-funktion är inte kollisionsbeständig.
29

Sida 30

Sök pussel. En sökning pussel består av

- en hashfunktion, H ,
- ett värde, id (som vi kallar ***pussel-ID***), som valts från en hög min-entropi fördelning
- och en målinställningspunkt Y

En lösning på detta pussel är ett värde, x , så att
$H(id \parallel x) \in \boldsymbol{Y}$.

Den intuition är detta: om H har en n-bitars utsignal, då det kan ta någon av 2^n värden. Lösa pusslet kräver att man hittar en ingång så att utmatningen faller inom mängden Y, som vanligtvis är mycket mindre än uppsättningen av alla utgångar. Storleken på Y avgör hur svårt pusslet är. Om Y är mängden av alla n-bitars strängar
pusslet är trivialt, medan om Y bara har 1 element är pusslet maximalt svårt. Det faktum att pussel-id har hög min-entropi säkerställer att det inte finns några genvägar. Tvärtom, om en viss ID-värdet var troligt, då kunde någon fuska, t.ex. genom att förbereda en lösning på pusslet med det ID.
Om ett sökpussel är pusselvänligt, innebär detta att det inte finns någon lösningsstrategi för detta pussel som är mycket bättre än att bara försöka slumpmässiga värden för x . Och så, om vi vill ställa ett pussel som är svårt att lösa kan vi göra det på det här sättet så länge vi kan generera pussel-ID:n på ett lämpligt slumpmässigt sätt. Det är vi
kommer att använda den här idén senare när vi pratar om Bitcoin-brytning, vilket är ett slags beräkningspussel.
SHA-256. Vi har diskuterat tre egenskaper hash funktioner och en tillämpning av var och en av dem.
Låt oss nu diskutera en speciell hashfunktion som vi kommer att använda mycket i den här boken. Det finns många
hash-funktioner som finns, men det här är den som Bitcoin främst använder, och den är ganska bra att använda. Det kallas ***SHA-256*** .
Kom ihåg att vi kräver att våra hashfunktioner fungerar på ingångar av godtycklig längd. Lyckligtvis så länge vi kan bygga en hashfunktion som fungerar på indata med fast längd, det finns en generisk metod för att konvertera den
till en hashfunktion som fungerar på godtyckliga ingångar. Det heter ***Merkle-Damgard förändra*** .
SHA-256 är en av ett antal vanliga hashfunktioner som använder sig av denna metod. I vanlig terminologi kallas den underliggande kollisionsbeständiga hashfunktionen med fast längd ***komprimeringsfunktionen*** . Det har bevisats att om den underliggande kompressionsfunktionen är kollision resistent, då är den övergripande hashfunktionen också kollisionsbeständig.
Merkle-Damgard-transformationen är ganska enkel. Säga komprimeringsfunktionen tar ingångar med längd m och producerar en utsignal från en mindre längd n . Ingången till hashfunktionen, som kan vara av vilken storlek som helst,
är uppdelad i ***block*** med längden m-n . Konstruktionen fungerar enligt följande: passera varje block tillsammans med
utgången från föregående block till komprimeringsfunktionen. Lägg märke till att ingångslängden då blir $(M\text{-}n) + n = m$, vilket är ingångslängden till komprimeringsfunktionen. För det första blocket, till vilket det inte finns någon föregående block utgång, i stället använder vi en ***Initialization Vector (IV)*** . Detta nummer återanvänds
för varje anrop till hash-funktionen, och i praktiken kan du bara slå upp det i ett standarddokument. De sista blockets utdata är resultatet som du returnerar.
30

SHA-256 använder en komprimeringsfunktion som tar 768-bitars indata och producerar 256-bitars utdata. De blockstorleken är 512 bitar. Se figur 1.3 för en grafisk skildring av hur SHA-256 fungerar.

Figur 1,3: SHA-256 hashfunktion (förenklat). SHA-256 användningar den Merkle-Damgard transform för att slå
en kollisionsbeständig kompressionsfunktion med fast längd till en hashfunktion som accepterar godtycklig längd ingångar. Ingången är "stoppad" så att dess längd är en multipel av 512 bitar.
Vi har pratat om hashfunktioner, kryptografiska hashfunktioner med speciella egenskaper, applikationer av dessa egenskaper och en specifik hashfunktion som vi använder i Bitcoin. I nästa avsnitt ska vi diskutera sätt att använda hashfunktioner för att bygga mer komplicerade datastrukturer som används i distribuerade system som Bitcoin.
Sidebar: modellering av hash-funktioner. Hash-funktioner är den schweiziska armékniven för kryptografi: de hitta en plats i en spektakulär mängd olika applikationer. Baksidan av denna mångsidighet är så annorlunda applikationer kräver lite olika egenskaper hos hashfunktioner för att säkerställa säkerheten. Det är bevisat notoriskt svårt att hitta en lista över hashfunktionsegenskaper som skulle resultera i bevisbara säkerhet över hela linjen.

I den här texten har vi valt ut tre egenskaper som är avgörande för hur hashfunktioner används i Bitcoin och andra kryptovalutor. Även inom detta utrymme är inte alla dessa fastigheter det nödvändig för varje användning av hashfunktioner. Till exempel är pusselvänlighet bara viktigt i Bitcoin-brytning, som vi kommer att se.

Designers av säkra system kastar ofta in handduken och modellhash fungerar som funktioner som mata ut ett oberoende slumpmässigt värde för varje möjlig ingång. Användningen av detta "slumpmässiga orakel modell" för att bevisa säkerhet är fortfarande kontroversiellt inom kryptografi. Oavsett ens ställning till denna debatt, resonemang om hur man kan minska de säkerhetsegenskaper som vi vill ha i våra applikationer till grundläggande egenskaper hos de underliggande primitiva är en värdefull intellektuell övning för bygga säkra system. Vår presentation i det här kapitlet är utformad för att hjälpa dig att lära dig denna färdighet.
31

1.2 Hashpekare och datastrukturer

I detta avsnitt kommer vi att diskutera ***hash pekare*** och deras tillämpningar. En hash-pekare är en data struktur som visar sig vara användbar i många av de system som vi kommer att prata om. En hash-pekare är helt enkelt en pekare till var viss information lagras tillsammans med en kryptografisk hash av information. Medan en vanlig pekare ger dig ett sätt att hämta informationen, en hash-pekare ger dig också ett sätt att verifiera att informationen inte har ändrats.

Figur 1,4 Hash pekaren. En hash pekaren är en pekare till den plats där data lagras tillsammans med en kryptografisk hash av värdet av dessa data vid någon bestämd tidpunkt.

Vi kan använda hashpekare för att bygga alla typer av datastrukturer. Intuitivt kan vi ta en bekant data struktur som använder pekare som en länkad lista eller ett binärt sökträd och implementerar det med hash pekare, istället för pekare som vi normalt skulle göra.
Block kedja. I figur 1.5, byggde vi en länkad lista med hjälp av hash pekare. Vi kommer att kalla dessa data strukturera ett ***blocksträngen*** . Medan som i en vanlig länkad lista där du har en serie block, var och en

blocket har data såväl som en pekare till föregående block i listan, i en blockkedja det föregående blocket pekaren kommer att ersättas med en hash-pekare. Så varje block berättar inte bara för oss var värdet på föregående block var, men det innehåller också en sammanfattning av det värdet som gör att vi kan verifiera att värdet
har inte förändrats. Vi lagrar huvudet på listan, som bara är en vanlig hash-pekare som pekar på senaste datablocket.
32

Sida 33

Figur 1,5 Block kedja. Ett blockkedja är en länkad lista som är byggd med hash pekare i stället för pekare.

Ett användningsfall för ett block kedja är en ***manipuleringsvisande logg*** . Det vill säga vi vill bygga en loggdatastruktur som
lagrar ett gäng data och låter oss lägga till data i slutet av loggen. Men om någon ändrar sig data som finns tidigare i loggen, vi kommer att upptäcka det.
För att förstå varför en blockkedja uppnår denna manipuleringssäkra egenskap, låt oss fråga vad som händer om en
motståndaren vill manipulera data som finns i mitten av kedjan. Närmare bestämt motståndarens Målet är att göra det på ett sådant sätt att någon som bara kommer ihåg hash-pekaren i spetsen av blockkedjan kommer inte att kunna upptäcka manipuleringen. För att uppnå detta mål ändrar motståndaren uppgifter av något blocket k . Eftersom data har ändrats, hash i block $k + 1$, som är en hash av hela blocket k , inte kommer att matcha upp. Kom ihåg att vi är statistiskt garanterade att det nya hash kommer inte att matcha det ändrade innehållet eftersom hash-funktionen är kollisionsbeständig. Och det kommer vi att göra
detektera bristande överensstämmelse mellan de nya data i blocket k och hash pekare i block $k + 1$. Av självklart kan motståndaren fortsätta att försöka täcka över denna förändring genom att ändra nästa blocks hash också. Motståndaren kan fortsätta göra detta, men denna strategi kommer att misslyckas när han når huvudet av listan. Närmare bestämt, så länge vi lagrar hash-pekaren högst upp i listan på en plats där motståndaren kan inte ändra det, motståndaren kommer inte att kunna ändra något block utan att bli upptäckt. Resultatet av detta är att om motståndaren vill manipulera data var som helst i hela denna kedja, i
För att hålla berättelsen konsekvent måste han manipulera hashpekarna hela vägen tillbaka till början. Och han kommer i slutändan att stöta på en vägspärr eftersom han inte kommer att kunna manipulera listans huvud. Sålunda framkommer att genom att bara komma ihåg denna enda hash-pekare, vi har i princip kommit ihåg en manipuleringssäker hash av hela listan. Så vi kan bygga en blockkedja som att den här innehåller så många block som vi vill, går tillbaka till något speciellt block i början av listan, som vi kallar ***genesis blocket*** .
Du kanske har märkt att blockkedjekonstruktionen liknar Merkle-Damgard-konstruktionen som vi såg i föregående avsnitt. De är faktiskt ganska lika, och samma säkerhetsargument gäller båda.
33

Sida 34

Figur 1,6 manipuleringsuppen logg. Om en motståndare ändrar data var som helst i blocket kedjan, kommer det att resultera
i hash-pekaren i följande block är felaktig. Om vi lagrar huvudet på listan, så även om motståndaren modifierar alla pekare för att överensstämma med den modifierade datan, huvudpekaren kommer att vara felaktiga och vi kommer att upptäcka manipuleringen.

Merkle träd. En annan användbar datastruktur som vi kan bygga med hjälp av hash pekare är ett binärt träd. A binärt träd med hash pekare är känd som en ***Merkle träd*** efter dess uppfinnare Ralph Merkle. Anta att vi har ett antal block som innehåller data. Dessa block utgör löven på vårt träd. Vi grupperar oss dessa datablock i par om två, och sedan för varje par bygger vi en datastruktur som har två hash pekare, en till vart och ett av dessa block. Dessa datastrukturer gör nästa nivå upp i trädet. Vi in omvandla dessa till grupper om två, och för varje par, skapa en ny datastruktur som innehåller hash av varje. Vi fortsätter att göra detta tills vi når ett enda block, trädets rot.

Sida 35

Figur 1,7 Merkle träd. I en Merkle träd, är datablocken grupperas i par och omkastningen av var och en av dessa block lagras i en överordnad nod. Föräldranoderna är i sin tur grupperade i par och deras hash lagras en nivå upp i trädet. Detta fortsätter hela vägen upp i trädet tills vi når rotnoden.

Som tidigare minns vi bara hash-pekaren i toppen av trädet. Vi har nu förmågan gå ner genom hashpekarna till valfri punkt i listan. Detta gör att vi kan se till att data har inte manipulerats eftersom, precis som vi såg med blockkedjan, om en motståndare manipulerar med något datablock längst ner i trädet, vilket kommer att orsaka hashpekaren som är en nivå upp för att inte matcha, och även om han fortsätter att manipulera med detta block, kommer förändringen så småningom
fortplanta sig till toppen av trädet där han inte kommer att kunna manipulera med hashpekaren som vi har lagrat. Så återigen, alla försök att manipulera någon bit av data kommer att upptäckas genom att bara komma ihåg hashpekaren överst.
Bevis på medlemskap. En annan trevlig funktion av Merkle träd är att, till skillnad från blocket kedja som vi byggt
tidigare tillåter det ett kortfattat bevis på medlemskap. Säg att någon vill bevisa att en viss data blocket är en medlem av Merkle Tree. Som vanligt minns vi bara roten. Då måste de visa upp sig oss detta datablock och blocken på vägen från datablocket till roten. Vi kan ignorera resten av trädet, eftersom blocken på denna väg är tillräckligt för att vi ska kunna verifiera hasharna ända upp till trädets rot. Se figur 1.8 för en grafisk beskrivning av hur detta fungerar.
Om det finns n noder i trädet, bara cirka $log\ (n)$ artiklar måste visas. Och eftersom varje steg bara kräver att beräkna hash barnets blocket, det tar ungefär $log\ (n)$ tid för oss att kontrollera det. Och så även om Merkle-trädet innehåller ett mycket stort antal block, kan vi fortfarande bevisa medlemskap i en relativt kort tid. Verifiering körs alltså i tid och rum som är logaritmiskt i antalet

---

Sida 36

noder i trädet.

Figur 1,8 Bevis på medlemskap. För att bevisa att ett datablock är inkluderad i trädet, behöver man bara visa blocken i sökvägen från det datablocket till roten.
En ***sorterade Merkle träd*** är bara en Merkle träd där vi tar blocken i botten, och vi kan sortera använder någon beställningsfunktion. Detta kan vara alfabetisk, lexikografisk ordning, numerisk ordning eller någon annan kom överens om beställning.
Bevis på icke-medlemskap. Med sorterade Merkle träd, blir det möjligt att kontrollera icke-medlemskap i en logaritmisk tid och rum. Det vill säga, vi kan bevisa att ett visst block inte finns i Merkle-trädet.

Och sättet vi gör det på är helt enkelt genom att visa en sökväg till objektet som är precis före var objektet är i fråga skulle vara och visar vägen till objektet som är precis efter där det skulle vara. Om dessa två artiklarna är konsekutiva i trädet, då fungerar detta som ett bevis på att objektet i fråga inte ingår.
För om det fanns med skulle det behöva vara mellan de två objekten som visas, men det finns inget utrymme mellan dem när de är på varandra.
Vi har diskuterat att använda hashpekare i länkade listor och binära träd, men mer generellt visar det sig att vi kan använda hashpekare i vilken pekarebaserad datastruktur som helst så länge som datastrukturen har inga cykler. Om det finns cykler i datastrukturen kommer vi inte att kunna göra alla hasharna matchar. Om du tänker på det, i en acyklisk datastruktur, kan vi börja nära löven, eller nära de saker som inte har några pekare som kommer ut ur dem, beräkna hasharna för dessa och arbeta oss sedan tillbaka mot början. Men i en struktur med cykler kan vi inte ta slut börja med och räkna tillbaka från.
Så, för att överväga ett annat exempel, kan vi bygga en riktad acyklisk graf av hashpekare. Och kunna verifiera medlemskap i den grafen mycket effektivt. Och det blir lätt att beräkna. Använder hash pekare på detta sätt är ett allmänt trick som du kommer att se gång på gång i samband med distribuerade datastrukturer och genom de algoritmer som vi diskuterar senare i detta kapitel och

---

genom hela den här boken.

1.3 Digitala signaturer

I det här avsnittet kommer vi att titta på ***digitala signaturer*** . Detta är den andra kryptografiska primitiv, tillsammans med
hashfunktioner, som vi behöver som byggstenar för diskussionen om kryptovaluta senare. En digital signaturen är tänkt att vara den digitala analoga till en handskriven signatur på papper. Vi önskar två egenskaper från digitala signaturer som väl motsvarar den handskrivna signaturanalogin. För det första, bara du kan göra din signatur, men alla som ser den kan verifiera att den är giltig. För det andra vill vi signaturen som ska knytas till ett visst dokument så att signaturen inte kan användas för att indikera ditt samtycke eller påskrift av ett annat dokument. För handskrivna signaturer, detta senare egendom är analog med att försäkra att någon inte kan ta din signatur och klippa bort den dokumentera och limma fast den på botten av en annan.
Hur kan vi bygga detta i digital form med hjälp av kryptografi? Låt oss först göra det föregående intuitivt diskussionen lite mer konkret. Detta kommer att tillåta oss att resonera bättre om system för digitala signaturer och diskutera deras säkerhetsegenskaper.
. *Digital signaturschema* A signaturschema digital består av följande tre algoritmer:

- **(sk, pk): = generateKeys (*keysize*)** Den generateKeys metod tar en nyckelstorlek och genererar ett nyckelpar. Den hemliga nyckeln *sk* hålls privat och används för att signera meddelanden. *pk* är allmänheten verifieringsnyckel som du ger till alla. Alla med denna nyckel kan verifiera din signatur.
- **sig: = sign (*sk* , *meddelande*)** Tecknet metoden tar ett meddelande och en hemlig nyckel, *sk* , som indata och utgångar en signatur för *meddelande* enligt *sk*
- **isValid: = kontrollera (*pk* , *meddelande* , *sig*)** Den kontrollera metoden tar ett meddelande, en signatur, och en
offentlig nyckel som indata. Den returnerar ett booleskt värde, *isValid* , som kommer att vara ***sant*** om *sig* är en giltig
signatur för *meddelande* under offentlig nyckel *pk* och ***falskt*** annars.

Vi kräver att följande två fastigheter håller:

- *Giltiga signaturer måste verifieras*

verifiera (*pk* , *meddelande* , **sign** (*sk* , *meddelande*)) == **true**

- signaturer är ***existentiellt förfalsk***

Vi noterar att **generateKeys** och **tecken** kan vara randomiserade algoritmer. GenereraKeys hade faktiskt bättre randomiseras eftersom det borde generera olika nycklar för olika personer. **kontrollera** , på andra sidan kommer alltid att vara deterministisk.
Låt oss nu undersöka de två egenskaperna som vi kräver av ett digitalt signatursystem mer i detalj.
Den första egenskapen är enkel - att giltiga signaturer måste verifieras. Om jag signera ett meddelande med *sk* , min hemlig nyckel, och någon försöker senare validera den signaturen över samma meddelande med min offentliga nyckel, *pk* , måste signaturen validera korrekt. Denna egenskap är ett grundläggande krav för att signaturer ska vara

---

användbar överhuvudtaget.
Unforgeability. Är det andra kravet att det är beräkningsmässigt omöjligt att förfalska signaturer.
Det vill säga en motståndare som känner till din offentliga nyckel och får se dina signaturer på någon annan meddelanden kan inte förfalska din signatur på något meddelande som han inte har sett din signatur för. Detta oförglömlighet egendom är i allmänhet formaliserad i termer av ett spel som vi spelar med en motståndare. De användning av spel är ganska vanligt i kryptografiska säkerhetsbevis.
I oförglömlighetsspelet finns det en motståndare som hävdar att han kan förfalska signaturer och en utmanare som kommer att testa detta påstående. Det första vi gör är att vi använder **generateKeys** att generera en hemlighet
signeringsnyckel och en motsvarande offentlig verifieringsnyckel. Vi ger den hemliga nyckeln till utmanaren, och vi ger den offentliga nyckeln till både utmanaren och motståndaren. Så motståndaren vet bara information som är offentlig, och hans uppdrag är att försöka förfalska ett budskap. Utmanaren vet hemligheten nyckel. Så han kan göra signaturer.
Intuitivt matchar uppsättningen av detta spel verkliga förhållanden. En riktig angripare skulle troligen vara det kunna se giltiga signaturer från sitt blivande offer på ett antal olika dokument. Och kanske kan angriparen till och med manipulera offret till att skriva under på oskadliga dokument om det är användbart för angriparen.
För att modellera detta i vårt spel, kommer vi att tillåta angriparen att få signaturer på vissa dokument av hans val, så länge han vill, så länge antalet gissningar är rimliga. För att ge en intuitiv idé om vad vi menar med ett rimligt antal gissningar, skulle vi tillåta angriparen att försöka 1 miljon gissningar, men inte 2^{80} gissningar. [2]
När angriparen är nöjd med att han har sett tillräckligt många signaturer, väljer angriparen ett meddelande, *M* , att de kommer att försöka skapa en signatur på. Den enda begränsningen på *M* är att det måste vara en meddelande som angriparen inte tidigare har sett en signatur för (eftersom angriparen kan uppenbarligen skicka tillbaka en signatur som han fick!). Utmanaren kör **kontrollera** algoritm för att avgöra om signaturen som produceras av angriparen är en giltig signatur på *M* under offentliga verifieringsnyckel. Om det lyckas verifiera, vinner angriparen spelet.

[2] I asymptotiska termer tillåter vi angriparen att prova ett antal gissningar som är en polynomfunktion av nyckelstorleken, men inte mer (t.ex. kan angriparen inte försöka exponentiellt många gissningar).

Figur 1.9 Unforgeability spel. Motståndaren och utmanaren spela unforgeability spelet. Om angriparen kan framgångsrikt mata ut en signatur på ett meddelande som han inte tidigare har sett, han vinner. Om han inte kan, vinner utmanaren och systemet med digitala signaturer är oförglömligt.

Vi säger att signaturschemat är oförglömligt om och bara om, oavsett vilken algoritm motståndaren använder, är hans chans att framgångsrikt förfalska ett meddelande extremt liten — så liten att vi kan anta att det aldrig kommer att hända i praktiken.

Praktiska bekymmer. Det finns ett antal praktiska saker som vi behöver göra för att vända den algoritm idé till en digital signaturmekanism som kan implementeras i praktiken. Till exempel många signaturalgoritmer är randomiserade (särskilt den som används i Bitcoin) och vi behöver därför en bra källa till slumpmässighet. Vikten av detta kan verkligen inte underskattas lika illa slumpmässighet kommer att göra din annars säkra algoritm osäker.

Ett annat praktiskt problem är meddelandestorleken. I praktiken finns det en gräns för meddelandestorleken du kan skriva under eftersom riktiga system kommer att fungera på bitsträngar med begränsad längd. Det finns ett enkelt sätt att komma runt denna begränsning: signera hashen för meddelandet, snarare än själva meddelandet. Om vi
använder en kryptografisk hashfunktion med en 256-bitars utdata, då kan vi effektivt signera ett meddelande om någon
längd så länge som vårt signaturschema kan signera 256-bitars meddelanden. Som vi diskuterade tidigare är det säkert att
använd hashen för meddelandet som ett meddelandesammandrag på detta sätt eftersom hashfunktionen är kollision

---

resistenta.

Ett annat knep som vi kommer att använda senare är att du kan signera en hash-pekare. Om du signerar en hash-pekare, då
signaturen täcker, eller skyddar, hela strukturen - inte bara själva hashpekaren utan
allt kedjan av hashpekare pekar på. Till exempel, om du skulle signera hash-pekaren som
var i slutet av en blockkedja, blir resultatet att du i praktiken skulle signera det digitalt
hela blockkedjan.

ECDSA. Nu ska vi komma in i muttrar och bultar. Bitcoin använder ett speciellt digitalt signaturschema, dvs kallad Elliptic Curve Digital Signature Algorithm (ECDSA). ECDSA är en amerikansk regeringsstandard, en uppdatering av den tidigare DSA-algoritmen anpassad för att använda elliptiska kurvor. Dessa algoritmer har fått omfattande kryptografisk analys under åren och anses allmänt vara säker.

Mer specifikt använder Bitcoin ECDSA över den elliptiska standardkurvan "secp256k1" som uppskattas att tillhandahålla 128 bitar av säkerhet (det vill säga, är det så svårt att bryta denna algoritm som utför 2^{128} kryptografiska operationer med symmetrisk nyckel som att anropa en hashfunktion). Medan denna kurva är en publicerad standard används den sällan utanför Bitcoin, med andra applikationer som använder ECDSA (som nyckelutbyte i TLS för säker webbsurfning) vanligtvis med den vanligare "secp256r1"-kurvan.

Detta är bara en egenhet med Bitcoin, eftersom detta valdes av Satoshi i den tidiga specifikationen av systemet och är nu svårt att ändra.

Vi kommer inte att gå in på alla detaljer om hur ECDSA fungerar eftersom det är en del komplicerad matematik inblandad, och
att förstå att det inte är nödvändigt för något annat innehåll i den här boken. Om du är intresserad av detaljerna, hänvisa till vårt avsnitt för vidare läsning i slutet av detta kapitel. Det kan vara bra att ha en uppfattning om

storlekar av olika kvantiteter, dock:

Privat nyckel: 256 bitar
Offentlig nyckel, okomprimerad: 512 bitar
Offentlig nyckel, komprimerad: 257 bitar
Meddelande som ska signeras: 256 bitar
Signatur: 512 bitar

Observera att även om ECDSA tekniskt sett bara kan signera meddelanden som är 256 bitar långa, är detta inte ett problem:
meddelanden hashas alltid innan de signeras, så effektivt meddelanden i alla storlekar kan vara effektivt signerad.

Med ECDSA är en bra källa till slumpmässighet avgörande eftersom en dålig källa till slumpmässighet sannolikt kommer
läcka din nyckel. Det är intuitivt logiskt att om du använder dålig slumpmässighet för att generera en nyckel, då nyckel som du genererar kommer sannolikt inte att vara säker. Men det är en egenhet med ECDSA att, även om du använder dåligt
[3]
[3] För de som är bekanta med DSA är detta en allmän egenhet i DSA och inte specifikt för den elliptiska kurvvarianten.

---

slumpmässighet bara i att göra en signatur, med din perfekta nyckel, som också kommer att läcka din privata nyckel. Och sedan är det game over; om du läcker din privata nyckel kan en motståndare förfalska din signatur. Vi måste därför vara särskilt försiktig med att använda bra slumpmässighet i praktiken, och använda en dålig källa till slumpmässighet är en vanlig fallgrop hos annars säkra system.
Detta avslutar vår diskussion om digitala signaturer som en kryptografisk primitiv. I nästa avsnitt,
vi kommer att diskutera några tillämpningar av digitala signaturer som kommer att visa sig vara användbara för att bygga
kryptovalutor.
Sidofält: kryptovalutor och kryptering. Om du har väntat på att få reda på vilken kryptering algoritm används i Bitcoin, vi är ledsna att göra dig besviken. Det finns ingen kryptering i Bitcoin, eftersom ingenting behöver krypteras, som vi ska se. Kryptering är bara en av en rik svit av tekniker som möjliggjorts av modern kryptografi. Många av dem, såsom åtagandesystem, involverar att dölja information på något sätt, men de skiljer sig från kryptering.

1.4 Offentliga nycklar som identiteter

Låt oss titta på ett trevligt trick som går ihop med digitala signaturer. Tanken är att ta en offentlig nyckel, en av dessa offentliga verifieringsnycklar från ett digitalt signaturschema, och likställ det med en identitet av en person eller en aktör i ett system. Om du ser ett meddelande med en signatur som verifierar korrekt under a publika nyckel, *pk* , då kan du se det som *pk* säger meddelandet. Du kan bokstavligen tänka på en publik nyckel som en slags skådespelare, eller en part i ett system som kan göra uttalanden genom att underteckna dessa uttalanden. Ur denna synvinkel är den offentliga nyckeln en identitet. För att någon ska kunna tala för identitet *pk* , måste de vet motsvarande hemlig nyckel, *sk* .
En konsekvens av att behandla offentliga nycklar som identiteter är att du kan skapa en ny identitet när du vill vill - du helt enkelt skapa en ny fräsch nyckelpar, *sk* och *pk* via **generateKeys** verksamheten i vår system för digitala signaturer. *pk* är den nya offentlig identitet som du kan använda, och *sk* är motsvarande hemlig nyckel som bara du vet och låter dig tala för räkning av identiteten *pk* . I praktiken kan du

använda hash av *pk* som din identitet eftersom publika nycklar är stora. Om du gör det, då för att verifiera att ett meddelande kommer från din identitet, kommer man måste kontrollera (1) som *pk* hashar verkligen till identitet, och (2) den meddelande verifierar enligt publik nyckel *pk* .
Dessutom, som standard, din publika nyckel *pk* kommer i princip ser slumpmässigt, och ingen kommer att kunna avslöja din verkliga världen identitet genom att undersöka *pk* . Du kan generera en ny identitet som ser [4] slumpmässigt, som ser ut som ett ansikte i mängden och som bara du kan kontrollera.
Decentraliserad identitetshantering. Detta leder oss till tanken på en decentraliserad identitetshantering. Istället för att ha en central myndighet som du måste gå till för att registrera dig som användare i ett system, du kan registrera dig som användare helt själv. Du behöver inte ha ett användarnamn och det behöver du inte heller

[4] Naturligtvis, när du börjar göra uttalanden med denna identitet, kan dessa uttalanden läcka information som tillåter en för att ansluta *pk* till verkliga världen identitet. Vi kommer att diskutera detta mer i detalj inom kort.

---

informera någon om att du kommer att använda ett visst namn. Vill du ha en ny identitet så kan du generera bara en när som helst och du kan göra så många du vill. Om du föredrar att bli känd av fem olika namn, inga problem! Gör bara fem identiteter. Om du vill vara lite anonym ett tag kan du skapa en ny identitet, använda den bara en liten stund och sedan slänga den. Alla dessa saker är möjliga med decentraliserad identitetshantering, och det är faktiskt så Bitcoin, gör identitet. Dessa identiteter kallas ***adresser*** i Bitcoin jargong. Du kommer ofta att höra termen adress som används i samband med Bitcoin och kryptovalutor, och det är egentligen bara en hash av en publik nyckel. Det är en identitet som någon skapat ur tomma intet, som en del av denna decentraliserade identitet ledningssystem.
Sidebar. Tanken att du kan skapa en identitet utan en centraliserad auktoritet kan tyckas kontraintuitiv. När allt kommer omkring, om någon annan har tur och genererar samma nyckel som du kan de inte stjäla dina bitcoins?
Svaret är att sannolikheten för att någon annan genererar samma 256-bitars nyckel som du är så liten att vi inte behöver oroa oss för det i praktiken. Vi är för alla ändamål garanterat att det aldrig kommer att hända.
Mer generellt, i motsats till nybörjares intuition att probabilistiska system är oförutsägbara och svårt att resonera om, ofta är det motsatta sant — statistikteorin tillåter oss att exakt kvantifiera chanserna för händelser vi är intresserade av och gör säkra påståenden om sådana system.
Men det finns en subtilitet: den probabilistiska garantin gäller endast när nycklar genereras slumpmässigt. Genereringen av slumpmässighet är ofta en svag punkt i verkliga system. Om två användares datorer använder samma källa till slumpmässighet eller använd förutsägbar slumpmässighet, då de teoretiska garantierna nej längre gälla. Så det är avgörande att använda en bra källa till slumpmässighet när du genererar nycklar för att säkerställa att praktiska garantier matchar de teoretiska.

Vid första anblicken kan det tyckas att decentraliserad identitetshantering leder till stor anonymitet och Integritet. När allt kommer omkring kan du skapa en identitet som ser slumpmässigt ut helt själv utan att berätta din verkliga identiteten. Men det är inte så enkelt. Med tiden skapar identiteten du skapar en serie av uttalanden. Folk ser dessa uttalanden och vet därmed att den som äger denna identitet har gjort ett vissa serier av åtgärder. De kan börja koppla ihop punkterna genom att använda den här serien av åtgärder för att sluta sig till saker om din verkliga identitet. En observatör kan länka samman dessa saker över tid och göra slutsatser som leder dem till slutsatser som, "Jösses, den här personen beter sig mycket som Joe. Kanske detta

personen är Joe."
Med andra ord, i Bitcoin behöver du inte explicit registrera eller avslöja din verkliga identitet, men mönstret för ditt beteende kanske i sig identifierar. Detta är den grundläggande integritetsfrågan i en kryptovaluta som Bitcoin, och vi kommer verkligen att ägna hela kapitel 6 åt det.

1.5 En enkel kryptovaluta

Låt oss nu gå från kryptografi till kryptovalutor. Att äta våra kryptografiska grönsaker kommer att börja att löna sig här, och vi kommer gradvis att se hur delarna passar ihop och varför kryptografiska operationer som hashfunktioner och digitala signaturer är faktiskt användbara. I det här avsnittet kommer vi att diskutera två mycket
enkla kryptovalutor. Naturligtvis kommer det att krävas mycket av resten av boken för att beskriva allt konsekvenserna av hur själva Bitcoin fungerar.

GoofyCoin

Den första av de två är GoofyCoin, som är ungefär den enklaste kryptovaluta vi kan tänka oss. där är bara två regler för GoofyCoin. Den första regeln är att en utsedd enhet, Goofy, kan skapa nya mynt när han vill och dessa nyskapade mynt tillhör honom.

För att skapa ett mynt genererar Långben ett unikt mynt-ID uniqueCoinID som han aldrig har genererat tidigare och konstruerar strängen "CreateCoin [uniqueCoinID]". Han beräknar sedan den digitala signaturen för denna sträng med sin hemliga signeringsnyckel. Snöret är tillsammans med Långbens signatur ett mynt. Någon kan verifiera att myntet innehåller Goofys giltiga signatur av ett CreateCoin-uttalande och därför är en giltigt mynt.

Den andra regeln för GoofyCoin är att den som äger ett mynt kan överföra det till någon annan.

Att överföra ett mynt är inte bara en fråga om att skicka myntdatastrukturen till mottagaren – det är det görs med hjälp av kryptografiska operationer.

Låt oss säga att Långben vill överföra ett mynt som han skapade till Alice. För att göra detta skapar han en ny uttalande som säger "Betala detta till Alice" där "detta" är en hashpekare som refererar till myntet i fråga. Och som vi såg tidigare är identiteter egentligen bara offentliga nycklar, så "Alice" syftar på Alices offentliga
nyckel. Slutligen signerar Långben strängen som representerar uttalandet. Eftersom Långben är den som ursprungligen
ägde det myntet måste han underteckna alla transaktioner som spenderar myntet. När denna datastruktur representerar Långbens transaktion undertecknad av honom existerar, Alice äger myntet. Hon kan bevisa för vem som helst
att hon äger myntet, eftersom hon kan presentera datastrukturen med Långbens giltiga signatur.

Dessutom pekar det på ett giltigt mynt som ägdes av Långben. Så giltigheten och ägandet av mynt är självklara i systemet.

När Alice väl äger myntet kan hon spendera det i sin tur. För att göra detta skapar hon ett uttalande som säger: "Betala
detta mynt till Bobs offentliga nyckel" där "detta" är en hashpekare till myntet som ägdes av henne. Och av naturligtvis undertecknar Alice detta uttalande. Vem som helst, när den presenteras med detta mynt, kan verifiera att Bob är
ägare. De skulle följa kedjan av hashpekare tillbaka till myntets skapelse och verifiera det kl
varje steg undertecknade den rättmätige ägaren ett uttalande som säger "betala detta mynt till [ny ägare]".

Figur 1.10 GoofyCoin mynt. Visas här är ett mynt som har skapats (botten) och tillbringade två gånger (mitten och toppen).

För att sammanfatta, reglerna för GoofyCoin är:

- Långben kan skapa nya mynt genom att helt enkelt underteckna ett uttalande om att han gör ett nytt mynt med en unikt mynt-ID.
- Den som äger ett mynt kan ge det vidare till någon annan genom att underteckna ett uttalande som säger: på detta mynt till X" (där X anges som en offentlig nyckel)
- Vem som helst kan verifiera ett mynts giltighet genom att följa kedjan av hashpekare tillbaka till dess skapad av Långben, verifiera alla signaturer längs vägen.

Naturligtvis finns det ett grundläggande säkerhetsproblem med GoofyCoin. Låt oss säga att Alice gav sitt mynt vidare till Bob genom att skicka hennes undertecknade uttalande till Bob men berättade det inte för någon annan. Hon kunde skapa en annan undertecknat uttalande som betalar samma mynt till Chuck. För Chuck verkar det som att det är perfekt giltig transaktion, och nu är han ägare till myntet. Bob och Chuck skulle båda ha ett giltigt utseende säger sig vara ägare till detta mynt. Detta kallas en attack med dubbla utgifter — Alice spenderar samma mynt två gånger. Intuitivt vet vi att mynt inte är tänkt att fungera på det sättet.

Faktum är att attacker med dubbla utgifter är ett av de viktigaste problemen som alla kryptovalutor måste lösa. GoofyCoin löser inte attacken med dubbla utgifter och därför är den inte säker. GoofyCoin är enkel, och dess mekanism för att överföra mynt är faktiskt väldigt lik Bitcoin, men för att den är det osäker det kommer inte att klippa det som en kryptovaluta.

ScroogeCoin

För att lösa problemet med dubbla utgifter kommer vi att designa en annan kryptovaluta, som vi kommer att kalla ScroogeCoin. ScroogeCoin är byggt av GoofyCoin, men det är lite mer komplicerat när det gäller data strukturer.

Den första nyckeln Tanken är att en utsedd enhet som kallas Scrooge publicerar en *append endast huvudbok* som innehåller historiken för alla transaktioner som har hänt. Egenskapen endast för tillägg säkerställer att all data som skrivs till denna reskontra kommer att finnas kvar för alltid. Om huvudboken verkligen bara är tillägg kan vi använda det för att försvara sig mot dubbla utgifter genom att kräva att alla transaktioner skrivs i huvudboken innan de accepteras. På så sätt kommer det att vara offentligt synligt om mynt tidigare skickats till en annan ägare.

För att implementera denna tilläggsfunktionalitet kan Scrooge bygga en blockkedja (datastrukturen vi diskuterats tidigare) som han kommer att signera digitalt. Det är en serie datablock, vart och ett med en transaktion i sig (i praktiken, som en optimering, skulle vi verkligen lägga flera transaktioner i samma block, som Bitcoin gör.) Varje block har ID för en transaktion, transaktionens innehåll och en hash-pekare till föregående block. Scrooge signerar digitalt den sista hashpekaren, som binder all data i hela denna struktur och publicerar signaturen tillsammans med blockkedjan.

Figur 1,11 ScroogeCoin blocket kedjan.

I ScroogeCoin räknas en transaktion bara om den är i blockkedjan som är signerad av Scrooge. Vem som helst kan

verifiera att en transaktion godkändes av Scrooge genom att kontrollera Scrooges signatur på blocket att den dyker upp. Scrooge ser till att han inte godkänner en transaktion som försöker dubbla utgifter ett redan förbrukat mynt.
Varför behöver vi en blockkedja med hashpekare utöver att Scrooge ska signera varje block? Detta säkerställer den endast tilläggsegenskapen. Om Scrooge försöker lägga till eller ta bort en transaktion till historiken, eller
ändra en befintlig transaktion kommer det att påverka alla följande block på grund av hashpekarna. Som så länge någon övervakar den senaste hashpekaren publicerad av Scrooge, kommer ändringen att vara det uppenbart och lätt att fånga. I ett system där Scrooge signerade block individuellt, måste du behålla spår av varenda signatur Scrooge som någonsin utfärdats. En blockkedja gör det väldigt enkelt för vilka två som helst
individer för att verifiera att de har observerat exakt samma historik för transaktioner undertecknade av Scrooge.
I ScroogeCoin finns det två typer av transaktioner. Den första typen är CreateCoins, som är precis som operation Långben kan göra i GoofyCoin som gör ett nytt mynt. Med ScroogeCoin förlänger vi semantik lite för att tillåta flera mynt att skapas i en transaktion.

Figur 1.12 CreateCoins transaktion. Detta CreateCoins transaktion skapar flera mynt. Varje mynt har ett serienummer i transaktionen. Varje mynt har också ett värde; det är värt ett visst antal scroogecoins. Slutligen har varje mynt en mottagare, vilket är en offentlig nyckel som får myntet när det är det skapas. Så CreateCoins skapar ett gäng nya mynt med olika värden och tilldelar dem personer som initiala ägare. Vi hänvisar till mynt av CoinIDs. Ett CoinID är en kombination av ett transaktions-ID och myntets serienummer inom den transaktionen.

En CreateCoins-transaktion är alltid giltig per definition om den är undertecknad av Scrooge. Vi kommer inte att oroa oss
när Scrooge har rätt att skapa mynt eller hur många, precis som vi inte oroade oss i GoofyCoin om hur Långben väljs som den enhet som får skapa mynt.
Den andra typen av transaktion är PayCoins. Den förbrukar en del mynt, det vill säga förstör dem, och skapar nya mynt med samma totala värde. De nya mynten kan tillhöra olika personer (offentliga nycklar). Denna transaktion måste undertecknas av alla som betalar in ett mynt. Så om du är ägare till ett av mynten som kommer att konsumeras i den här transaktionen, då måste du signera digitalt transaktion för att säga att du verkligen är okej med att spendera detta mynt.
Reglerna för ScroogeCoin säger att PayCoins transaktion är giltig om fyra saker är sanna:

- De förbrukade mynten är giltiga, det vill säga de skapades verkligen i tidigare transaktioner.
- De förbrukade mynten har inte redan förbrukats i någon tidigare transaktion. Det är det detta är inte en dubbelutgift.
- Det totala värdet av mynten som kommer ut från denna transaktion är lika med det totala värdet av mynt som gick in. Det vill säga, bara Scrooge kan skapa nytt värde.
- Transaktionen är giltigt undertecknad av ägarna av alla konsumerade mynt.

Figur 1,13 **A PayCoins transaktion.**

Om alla dessa villkor är uppfyllda är denna PayCoins-transaktion giltig och Scrooge kommer att acceptera den. Han kommer att skriva in det i historien genom att lägga till det i blockkedjan, varefter alla kan se att detta transaktion har skett. Det är först vid denna tidpunkt som deltagarna kan acceptera att transaktionen faktiskt har inträffat. Tills den publiceras kan den förebyggas av en transaktion med dubbla utgifter även om det i övrigt är giltigt enligt de tre första villkoren.

Mynt i detta system är oföränderliga - de ändras aldrig, delas upp eller kombineras. Varje mynt är skapas en gång i en transaktion och senare konsumeras i någon annan transaktion. Men vi kan få samma effekt som att kunna dela upp eller kombinera mynt genom att använda transaktioner. Till exempel att dela upp ett mynt, skapar Alice en ny transaktion som förbrukar det ena myntet och sedan producerar två nya mynt av samma totala värde. De två nya mynten kan tilldelas tillbaka till henne. Så fast mynt är oföränderliga i detta system, det har all flexibilitet som ett system som inte hade oföränderligt mynt.

Nu kommer vi till kärnproblemet med ScroogeCoin. ScroogeCoin kommer att fungera i den meningen att människor kan se vilka mynt som är giltiga. Det förhindrar dubbla utgifter, eftersom alla kan titta in i blocket kedja och se att alla transaktioner är giltiga och att varje mynt bara konsumeras en gång. Men Problemet är Scrooge — han har för mycket inflytande. Han kan inte skapa falska transaktioner, för det kan han inte förfalska andras signaturer. Men han kunde sluta stödja transaktioner från vissa användare och förneka dem servar och gör deras mynt oanvändbara. Om Scrooge är girig (som hans tecknade namne föreslår) han kunde vägra att publicera transaktioner såvida de inte överför någon mandat transaktionsavgift till honom. Scrooge kan naturligtvis också skapa hur många nya mynt som helst för sig själv. Eller Scrooge kunde bli uttråkad av hela systemet och sluta uppdatera blockkedjan helt.

Problemet här är centraliseringen. Även om Scrooge är nöjd med detta system, är vi som användare av det, är kanske inte. Även om ScroogeCoin kan verka som ett orealistiskt förslag, är mycket av den tidiga forskningen om kryptosystem antog att det verkligen skulle finnas någon central betrodd auktoritet, vanligtvis kallad en *bank* . När allt kommer omkring har de flesta verkliga valutor en pålitlig emittent (vanligtvis en statlig myntverk) ansvarig för att skapa valuta och avgöra vilka sedlar som är giltiga. Däremot kryptovalutor med en central myndighet i stort sett misslyckats i praktiken. Det finns många anledningar till detta, men i efterhand verkar det som att det är svårt att få människor att acceptera en kryptovaluta med en centraliserad auktoritet.

Därför är den centrala tekniska utmaningen som vi måste lösa för att förbättra ScroogeCoin och skapa ett fungerande system är: kan vi descroogify systemet? Det vill säga kan vi bli av med det centraliserad Scrooge-figur? Kan vi ha en kryptovaluta som fungerar som ScroogeCoin i många sätt, men har ingen central betrodd auktoritet?

För att göra det måste vi ta reda på hur alla användare kan komma överens om en enda publicerad blockkedja som historik över vilka transaktioner som har skett. De måste alla komma överens om vilka transaktioner som är giltiga, och vilka transaktioner som faktiskt har skett. De måste också kunna tilldela ID till saker i en decentraliserat sätt. Slutligen måste präglingen av nya mynt kontrolleras på ett decentraliserat sätt. Om vi kan lösa alla dessa problem, sedan kan vi bygga en valuta som skulle vara som ScroogeCoin men utan ett centraliserat parti. I själva verket skulle detta vara ett system som mycket liknar Bitcoin.

Vidare läsning

Steven Levy s *Crypto* är en trevlig, icke-teknisk titt på utvecklingen av modern kryptografi

och människorna bakom:
. Levy, Steven ***Crypto: hur koden Rebels Vispa regeringen - Spara Privacy i den digitala eran.*** **Penguin, 2001.**
Modern kryptografi är ett ganska teoretiskt område. Kryptografer använder matematik för att definiera primitiver, protokoll och deras önskade säkerhetsegenskaper på ett formellt sätt, och för att bevisa att de är säkra baserat på allmänt accepterade antaganden om beräkningshårdheten hos specifik matematisk uppgifter. I det här kapitlet har vi använt intuitivt språk för att diskutera hashfunktioner och digitala signaturer. För läsaren som är intresserad av att utforska dessa och andra kryptografiska koncept på ett mer matematiskt sätt sätt och mer detaljerat hänvisar vi dig till:
Katz, Jonathan och Yehuda Lindell. ***Introduction to Modern Cryptography, Second Edition.*** **CRC Press, 2014.**
För en introduktion till tillämpad kryptografi, se:
. Ferguson, Niels, Bruce Schneier, och Tadayoshi Kohno ***Cryptography engineering: konstruktionsprinciper och praktiska tillämpningar*** **. John Wiley & Sons, 2012.**

Att granska NIST-standarden som definierar SHA-2 är ett bra sätt att få en intuition för vilken kryptografisk standarder ser ut som:
FIPS PUB 180-4, Secure Hash Standard , Federal Information Processing Standards Publikation . Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, 2008.
Slutligen, här är uppsatsen som beskriver den standardiserade versionen av ECDSA-signaturalgoritmen.
Johnson, Don, Alfred Menezes, och Scott Vanstone. Den elliptisk kurva digital signatur algoritm (ECDSA) . International Journal of Information Security 1.1 (2001): 36-63.

Övningar

1. **Authenticated datastrukturer.** Du designar SecureBox, en autentiserad fillagring online

systemet. För enkelhetens skull finns det bara en enda mapp. Användare måste kunna lägga till, redigera, ta bort och

hämta filer och för att lista mappinnehållet. När en användare hämtar en fil måste SecureBox tillhandahålla en

bevis på att filen inte har manipulerats sedan den senaste uppdateringen. Om en fil med det angivna namnet

inte existerar, måste servern rapportera det — igen med ett bevis.

Vi vill minimera storleken på dessa bevis, tidskomplexiteten för att verifiera dem och storleken på

sammandraget som användaren måste lagra mellan operationerna. (Naturligtvis, för att kunna verifiera bevis, användare

måste hela tiden lagra en del tillstånd som inte är noll härledd från mappinnehållet. Förutom

denna sammanfattning har användaren inget minne av innehållet i filerna hon lade till.)

Här är ett naivt tillvägagångssätt. Användarens sammanfattning är en hash av hela mappens innehåll, och bevis är det

kopior av hela mappens innehåll. Detta resulterar i en liten sammanfattning men stora bevis och lång verifiering

gånger. Dessutom måste användaren hämta hela mappen innan han utför lägg till/ta bort/redigera operationer

så att hon kan räkna om sammandraget.

Alternativt kan sammanfattningen bestå av en separat hash för varje fil, och varje fil skulle vara sin egen

bevis. Nackdelen med detta tillvägagångssätt är att det kräver sammanfattningsutrymme som är linjärt i antalet

filer i systemet.

Kan du utforma ett protokoll där bevisstorlek, verifieringstid och sammanfattningsstorlek alla är sublinjära? Du

kan behöva ett underprotokoll som involverar en viss mängd tvåvägskommunikation för användaren

kunna uppdatera sin sammanfattning när hon kör och lägga till, ta bort eller redigera.

Tips: använd Merkle-trädidén från avsnitt 1.2.

2. **födelsedagsattack.** Låt H vara en idealisk hashfunktion som producerar en n-bits utdata. Med ideal menar vi

att så långt vi vet, är oberoende och likformigt fördelade i $\{0,1\}$ varje hashvärde ^{n}
. Trivialt,

Sida 50

vi kan gå igenom två ^{n}
+ 1 olika värden så hittar vi garanterat en kollision. Om vi är det

begränsat för utrymme kan vi bara lagra 1 ingångspar och fortsätta att prova nya ingångar tills vi träffar

samma utgång igen. Detta har tidskomplexitet O (2 ^{n}
), men har O(1) rymdkomplexitet. Alternativt

vi kunde beräkna kontrollsummor av omkring O (2 $^{n/2}$
) olika ingångar och lagra alla ingångspar. Som

vi såg i texten, det finns en god chans att två av dessa utdata skulle kollidera (”födelsedagen

paradox"). Det visar att vi kan nå en tid och rymd trade-off: O (2 $^{N/2}$
) Tid och O (2 $^{n/2}$
) Plats.

1. (Lätt) Visa att avvägningen mellan tid och rum är parameteriserbar: vi kan uppnå vilket utrymme som helst

komplexitet mellan O (1) och O ($2^{n/2}$) med en motsvarande minskning i tidskomplexitet.
2. (Mycket hård) Finns det ett angrepp för vilka produkten av tid och rum komplexiteten är o (2^{n})?
[Kom ihåg den lilla oh-notationen.]

3. **hashfunktion egenskaper** (igen). Låt H vara en hashfunktion som är både döljande och pusselvänlig.

Överväga G (z) = H (z) ‖ z_{sista} där $z_{\text{förra}}$ representerar den sista biten av z. Visa att G är pusselvänlig men inte

gömmer sig.

4. **Randomness** .I ScroogeCoin Anta Mallory försöker generera (sk, pk) par tills hennes hemliga nyckeln

matchar någon annans. Vad kommer hon att kunna göra? Hur lång tid tar det innan hon lyckas, på

genomsnitt? Tänk om Alices slumptalsgenerator har en bugg och hennes nyckelgenereringsprocedur

producerar bara 1 000 distinkta par?

Kapitel 2: Hur Bitcoin uppnår decentralisering

I det här kapitlet kommer vi att diskutera decentralisering i Bitcoin. I det första kapitlet tittade vi på krypton grunderna som ligger bakom Bitcoin och vi avslutade med en enkel valuta som vi kallade ScroogeCoin. ScroogeCoin uppnår mycket av det vi vill ha i en redovisningsbaserad kryptovaluta, men den har en iögonfallande problem — det förlitar sig på den centraliserade myndigheten som heter Scrooge. Vi avslutade med frågan om hur man decentraliserar, eller de-Scrooge-ifierar, denna valuta, och att svara på den frågan kommer att vara i fokus detta kapitel.

När du läser igenom det här kapitlet, notera att mekanismen genom vilken Bitcoin uppnår decentralisering är inte rent tekniskt, utan det är en kombination av tekniska metoder och smart incitamentteknik. I slutet av detta kapitel bör du ha en riktigt bra uppskattning för hur denna decentralisering sker, och mer allmänt hur Bitcoin fungerar och varför det är säkert.

2.1 Centralisering vs. decentralisering

Decentralisering är ett viktigt koncept som inte är unikt för Bitcoin. Tanken på att tävla paradigm för centralisering kontra decentralisering uppstår i en mängd olika digitala teknologier. För att bäst förstå hur det utspelar sig i Bitcoin är det användbart att förstå den centrala konflikten — spänningen mellan dessa två paradigm — i en mängd andra sammanhang.

Å ena sidan har vi Internet, ett berömt decentraliserat system som har historiskt sett konkurrerade med och segrade mot "walled-garden"-alternativ som AOL:s och CompuServes informationstjänster. Sedan finns det e-post, som i sin kärna är ett decentraliserat system baserat på Simple Mail Transfer Protocol (SMTP), en öppen standard. Medan den har konkurrens från proprietära meddelandesystem som Facebook eller LinkedIn mail, e-post har lyckats förbli den standard för person-till-person kommunikation online. När det gäller snabbmeddelanden och text meddelandehantering har vi en hybridmodell som inte kategoriskt kan beskrivas som centraliserad eller decentraliserat. Äntligen finns det sociala nätverk: trots många gemensamma ansträngningar från hobbyister, utvecklare och entreprenörer för att skapa alternativ till den dominerande centraliserade modellen, centraliserad system som Facebook och LinkedIn dominerar fortfarande detta utrymme. I själva verket är denna konflikt länge före den
digital era och vi ser en liknande kamp mellan de två modellerna i historien om telefoni, radio, tv och film.

Decentralisering är inte allt eller inget; nästan inget system är rent decentraliserat eller rent centraliserat. Till exempel är e-post i grunden ett decentraliserat system baserat på ett standardiserat protokoll, SMTP, och alla som vill kan driva en egen e-postserver. Ändå, vad har hänt i marknaden är att ett litet antal centraliserade webbmailleverantörer har blivit dominerande. Liknande, medan Bitcoin-protokollet är decentraliserat, tjänster som Bitcoin-utbyten, där du kan konvertera

Bitcoin till andra valutor, och plånboksmjukvara, eller programvara som låter människor hantera sina bitcoins kan vara centraliserade eller decentraliserade i olika grad.

Med detta i åtanke, låt oss bryta ner frågan om hur Bitcoin-protokollet uppnår decentralisering i fem mer specifika frågor:

1. Vem sköter redovisningen av transaktioner?
2. Vem har auktoritet över vilka transaktioner som är giltiga?
3. Vem skapar nya bitcoins?
4. Vem bestämmer hur systemets regler förändras?
5. Hur får bitcoins bytesvärde?

De tre första frågorna återspeglar de tekniska detaljerna i Bitcoin-protokollet, och det är dessa frågor det kommer att vara fokus i detta kapitel.

Olika aspekter av Bitcoin faller på olika punkter på centraliserings-/decentraliseringsspektrumet. Peer-to-peer-nätverket är närapå rent decentraliserat eftersom vem som helst kan köra en Bitcoin-nod och det finns en ganska låg inträdesbarriär. Du kan gå online och enkelt ladda ner en Bitcoin-klient och köra en nod på din bärbara dator eller din PC. För närvarande finns det flera tusen sådana noder. Bitcoin *mining* , som vi kommer att studera senare i det här kapitlet, är tekniskt sett också öppet för alla, men det kräver en mycket hög
kapitalkostnad. På grund av detta har det skett en hög grad av centralisering, eller en koncentration av makt, i Bitcoins gruv-ekosystem. Många i Bitcoin-gemenskapen ser detta som ganska oönskat.
En tredje aspekt är uppdateringar av programvaran som Bitcoin-noder kör, och detta har betydelse för hur och när systemets regler förändras. Man kan föreställa sig att det finns många interoperabla implementeringar av protokollet, som med e-post. Men i praktiken kör de flesta noder referensen implementering, och dess utvecklare är betrodda av samhället och har mycket makt.

2.2 Utdelad konsensus

Vi har diskuterat, på ett generiskt sätt, centralisering och decentralisering. Låt oss nu undersöka decentralisering i Bitcoin på en mer teknisk nivå. Ett nyckelbegrepp som kommer att dyka upp i detta diskussion är ***konsensus*** , och specifikt, ***distribueras konsensus*** . Det viktigaste tekniska problemet att lösa i att bygga ett distribuerat e-cash-system uppnår distribuerad konsensus. Intuitivt kan du tänka dig vårt mål att decentralisera ScroogeCoin, den hypotetiska valutan som vi såg i det första kapitlet.

Distribuerad konsensus har olika tillämpningar, och den har studerats i decennier inom datorer vetenskap. Den traditionella motiverande tillämpningen är tillförlitlighet i distribuerade system. Föreställ dig att du är med
ansvarig för backend för ett stort socialt nätverksföretag som Facebook. System av det här slaget har vanligtvis tusentals eller till och med miljontals servrar, som tillsammans bildar en massiv distribuerad databas som registrerar alla åtgärder som händer i systemet. Varje del av information måste vara spelas in på flera olika noder i denna backend, och noderna måste vara synkroniserade med helheten

systemets tillstånd.

Konsekvenserna av att ha ett distribuerat konsensusprotokoll sträcker sig långt utöver detta traditionella Ansökan. Om vi hade ett sådant protokoll skulle vi kunna använda det för att bygga ett enormt, distribuerat nyckelvärdelager,
som mappar godtyckliga nycklar, eller namn, till godtyckliga värden. En distribuerad nyckelvärdebutik skulle i sin tur
aktivera många applikationer. Till exempel kan vi använda det för att bygga ett distributivt domännamnssystem, som helt enkelt är en mappning mellan mänskliga förståeliga domännamn till IP-adresser. Vi kunde bygga en publik nyckelkatalog, som är en mappning mellan e-postadresser (eller någon annan form av verkliga identitet) till offentliga nycklar.

Det är intuitionen av vad distribuerad konsensus är, men det är användbart att ge en teknisk definition som detta hjälper oss att avgöra om ett givet protokoll uppfyller kraven eller inte.

Distribuerad konsensus-protokollet. Det finns n noder som var och en har ett ingångsvärde. Några av dessa noder är felaktig eller skadlig. Ett distribuerat konsensusprotokoll har följande två egenskaper:

- Den måste avslutas med alla ärliga noder överens om värdet
- Värdet måste ha genererats av en ärlig nod

Vad betyder detta i samband med Bitcoin? För att förstå hur distribuerad konsensus skulle kunna fungera i Bitcoin, kom ihåg att Bitcoin är ett peer-to-peer-system. När Alice vill betala Bob, vad hon faktiskt gör är att sända en transaktion till alla Bitcoin-noder som utgör peer-to-peer nätverk. Se figur 2.1.

Figur 2,1 Broadcasting en transaktion För att betala Bob, Alice sänder transaktionen till hela Bitcoin peer-to-peer-nätverk.

Förresten, du kanske har märkt att Alice sänder transaktionen till alla Bitcoin peer-to-peer noder, men Bobs dator finns ingenstans på den här bilden. Det är naturligtvis möjligt att Bob kör en av dem noderna i peer-to-peer-nätverket. I själva verket, om han vill bli meddelad att denna transaktion gjordes faktiskt inträffa och att han fick betalt, kan det vara en bra idé att köra en nod. Ändå finns det ingen krav på att Bob ska lyssna på nätverket; Att köra en nod är inte nödvändigt för att Bob ska ta emot medlen. Bitcoins kommer att vara hans oavsett om han driver en nod på nätverket.

Vad exakt är det som noderna kanske vill nå konsensus om i Bitcoin-nätverket? Givet att en mängd olika användare sänder dessa transaktioner till nätverket, måste noderna komma överens om

---

exakt vilka transaktioner som sändes och i vilken ordning dessa transaktioner skedde. Detta kommer att resultera i en enda global reskontra för systemet. Kom ihåg att i ScroogeCoin, för optimering, lägger vi
transaktioner till block. På liknande sätt, i Bitcoin, gör vi konsensus på block-för-block-basis.

Så vid varje given punkt har alla noder i peer-to-peer-nätverket en reskontra som består av en sekvens av block, som vart och ett innehåller en lista över transaktioner som de har nått konsensus om. Dessutom har varje nod en pool av utestående transaktioner som den har hört talas om men inte har ännu inkluderats i blockkedjan. För dessa transaktioner har konsensus ännu inte skett, och så per definition kan varje nod ha en något annorlunda version av den utestående transaktionspoolen. I

praktiken inträffar detta eftersom peer-to-peer-nätverket inte är perfekt, så vissa noder kan ha hört om en transaktion som andra noder inte har hört talas om.

Hur exakt kommer noder till konsensus om ett block? Ett sätt att göra detta: med jämna mellanrum, säg var tionde minut föreslår varje nod i systemet sin egen utestående transaktionspool som nästa block. Sedan exekverar noderna något konsensusprotokoll, där varje nods input är sin egen föreslagna blocket. Nu kan vissa noder vara skadliga och lägga in ogiltiga transaktioner i sina block, men vi kan anta att andra noder kommer att vara ärliga. Om konsensusprotokollet lyckas, ett giltigt block kommer att väljas som utgång. Även om det valda blocket föreslagits av endast en nod, är det ett giltigt utgång så länge blocket är giltigt. Nu kan det finnas någon giltig utestående transaktion som inte gjorde det inkluderas i blocket, men detta är inget problem. Om någon transaktion på något sätt inte blev till just det här blocket, det kunde bara vänta och komma in i nästa block.

Tillvägagångssättet i föregående stycke har vissa likheter med hur Bitcoin fungerar, men det är det inte riktigt hur det fungerar. Det finns ett antal tekniska problem med detta tillvägagångssätt. För det första, konsensus i generellt är ett svårt problem eftersom noder kan krascha eller vara direkt skadliga. För det andra, och specifikt i Bitcoin-sammanhang är nätverket mycket ofullkomligt. Det är ett peer-to-peer-system, och inte alla par noder är anslutna till varandra. Det kan finnas fel i nätverket på grund av dåligt internet anslutning till exempel, och därmed köra ett konsensusprotokoll där alla noder måste delta är inte riktigt möjligt. Slutligen finns det mycket latens i systemet eftersom det är distribuerat över hela systemet Internet.

Sidofält: Bitcoin-protokollet måste nå konsensus inför två typer av hinder: brister i nätverket, såsom latens och noder som kraschar, samt avsiktliga försök av några noder för att undergräva processen.

En speciell konsekvens av denna höga latens är att det inte finns någon uppfattning om global tid. Vad är detta betyder att inte alla noder kan komma överens om en gemensam ordning av händelser helt enkelt baserat på observation
tidsstämplar. Så konsensusprotokollet kan inte innehålla instruktioner av formen "Noden som skickade det första meddelandet i steg 1 måste göra X i steg 2." Detta kommer helt enkelt inte att fungera eftersom inte alla noder kommer att fungera
komma överens om vilket meddelande som skickades först i steg 1 i protokollet.

Omöjliga resultat. Bristen på global tid begränsar kraftigt uppsättningen av algoritmer som kan användas i konsensusprotokollen. Faktum är att på grund av dessa begränsningar distribueras mycket av litteraturen om

---

konsensus är något pessimistisk, och många omöjliga resultat har bevisats. En mycket bra kända omöjliga resultat gäller ***bysantinska Generals Problem*** . I detta klassiska problem
Den bysantinska armén är uppdelad i divisioner, var och en beordrad av en general. Generalerna kommunicerar per budbärare för att utarbeta en gemensam handlingsplan. Vissa generaler kan vara förrädare och kanske försök avsiktligt att undergräva processen så att de lojala generalerna inte kan komma fram till en enhetlig plan. De
Målet med detta problem är att alla lojala generaler ska komma fram till samma plan utan de förrädare generaler som kan få dem att anta en dålig plan. Det har bevisats att detta är omöjligt uppnå om en tredjedel eller fler av generalerna är förrädare.

Ett mycket mer subtilt omöjlighetsresultat, känt för namnen på författarna som först bevisade det, är kallas Fischer-Lynch-Paterson-omöjlighetsresultatet. Under vissa förhållanden, som inkluderar noder som agerar på ett deterministiskt sätt, de visade att konsensus är omöjligt med ens en enda felaktig process.

Trots dessa omöjliga resultat finns det några konsensusprotokoll i litteraturen. En av mer känd bland dessa protokoll är ***Paxos*** . Paxos gör vissa kompromisser. Å ena sidan, det ger aldrig ett inkonsekvent resultat. Å andra sidan accepterar den avvägningen som under vissa omständigheter förhållanden, om än sällsynta, kan protokollet fastna och misslyckas med att göra några framsteg.

Breaking traditionella antaganden. Men det finns goda nyheter: dessa omöjliga resultat bevisats i en mycket specifik modell. De var avsedda att studera distribuerade databaser, och den här modellen bär inte över mycket bra till Bitcoin-inställningen eftersom Bitcoin bryter mot många av de antaganden som är inbyggda i modeller. På ett sätt berättar resultaten mer om modellen än om problemet med fördelad konsensus.

Ironiskt nog, med det nuvarande forskningsläget, fungerar konsensus i Bitcoin bättre i praktiken än i teori. Det vill säga, vi observerar att konsensus fungerar, men har inte utvecklat teorin för att helt förklara varför det fungerar. Men att utveckla en sådan teori är viktigt eftersom det kan hjälpa oss att förutsäga oförutsedda attacker
och problem, och bara när vi har en stark teoretisk förståelse för hur Bitcoin konsensus fungerar kommer vi att ha starka garantier för Bitcoins säkerhet och stabilitet.

Vilka är antagandena i traditionella modeller för konsensus som Bitcoin bryter mot? Först introducerar den idén om incitament, som är ny för ett distribuerat konsensusprotokoll. Detta är endast möjligt i Bitcoin eftersom det är en valuta och därför har en naturlig mekanism att uppmuntra deltagarna till agera ärligt. Så Bitcoin löser inte riktigt det distribuerade konsensusproblemet i generell mening, men det löser det i det specifika sammanhanget för ett valutasystem.

För det andra, Bitcoin omfattar begreppet slumpmässighet. Som vi kommer att se i de kommande två avsnitten, Bitcoins
konsensusalgoritmen är starkt beroende av randomisering. Det gör också bort med föreställningen om en specifik utgångspunkt och slutpunkt för konsensus. Istället sker konsensus över en lång tidsperiod, cirka en timme i det praktiska systemet. Men även vid slutet av den tiden kan noder inte vara säkra på det någon särskild transaktion eller ett block har hamnat i redovisningen. Istället, allt eftersom tiden går sannolikheten att din vy av ett block kommer att matcha den eventuella konsensusvyn ökar, och

---

Sannolikheten för att åsikterna kommer att skilja sig minskar exponentiellt. Dessa skillnader i modellen är nyckeln till hur Bitcoin kommer runt de traditionella omöjlighetsresultaten för distribuerad konsensus protokoll.

2.3 Konsensus utan identitet med hjälp av en blockkedja

I det här avsnittet kommer vi att studera de tekniska detaljerna i Bitcoins konsensusalgoritm. Kom ihåg att Bitcoin noder har inte beständiga, långsiktiga identiteter. Detta är en annan skillnad från traditionella distribuerade konsensusalgoritmer. En orsak till denna brist på identiteter är den hos en peer-to-peer systemet finns det ingen central auktoritet att tilldela identiteter till deltagare och verifiera att de inte är det

skapa nya noder efter behag. Den tekniska termen för detta är en ***Sybil attack*** . Sybils är bara kopior av noder som en illvillig motståndare kan skapa för att se ut som om det finns många olika deltagare, även om det faktiskt är
alla dessa pseudo-deltagare kontrolleras verkligen av samma motståndare. Den andra anledningen är det Pseudonymitet är i sig ett mål för Bitcoin. Även om det vore möjligt eller lätt att fastställa identiteter för alla noder eller alla deltagare, vi skulle inte nödvändigtvis vilja göra det. Även om Bitcoin inte ger stark anonymitet garanterar genom att de olika transaktionerna som man gör ofta kan kopplas samman tillsammans har den egenskapen att ingen tvingas avslöja sin verkliga identitet, som deras namn eller IP-adress, för att delta. Och det är en viktig egenskap och en central egenskap hos Bitcoins design.

Om noder hade identiteter skulle designen vara enklare. Till att börja med skulle identiteter tillåta oss att sätta i protokollinstruktionerna i formuläret, "Nu ska noden med det lägsta numeriska ID:t ta ett steg." Utan identiteter är uppsättningen av möjliga instruktioner mer begränsad. Men mycket mer allvarliga skäl för noder att ha identiteter är för säkerheten. Om noder identifierades och det inte var trivialt för att skapa nya nodidentiteter, då skulle vi kunna göra antaganden om antalet noder som finns skadlig, och vi kan härleda säkerhetsegenskaper ur det. Av båda dessa skäl, bristen på identiteter introducerar svårigheter för konsensusprotokollet i Bitcoin.

Vi kan kompensera för bristen på identiteter genom att göra ett svagare antagande. Antag att det finns på något sätt en förmåga att välja en slumpmässig nod i systemet. En bra motiverande analogi för detta är en lotteri eller lotteri, eller valfritt antal verkliga system där det är svårt att spåra människor, ge dem identiteter och verifiera dessa identiteter. Det vi gör i de sammanhangen är att dela ut polletter eller biljetter eller något liknande. Det gör det möjligt för oss att senare välja ett slumpmässigt token-ID och ringa till ägaren av det ID:t.
Så för tillfället, ta ett steg av tro och anta att det är möjligt att välja en slumpmässig nod från Bitcoin-nätverk på detta sätt. Antag vidare, för tillfället, att denna tokengenerering och distributionsalgoritmen är tillräckligt smart så att om motståndaren ska försöka skapa en hel del Sybil noder, kommer alla dessa Sybils tillsammans bara att få en token. Detta betyder att motståndaren inte kan multiplicera sin kraft genom att skapa nya noder. Om du tycker att detta är mycket att anta, oroa dig inte. Senare i detta
kapitlet tar vi bort dessa antaganden och visar i detalj hur egenskaper likvärdiga med dessa är realiseras i Bitcoin.

Implicit Consensus. Detta antagande av slumpmässiga nod val möjliggör något som kallas ***implicit konsensus*** . Det finns flera omgångar i vårt protokoll, var och en motsvarar ett annat block i blockkedjan. I varje omgång väljs en slumpmässig nod på något sätt, och denna nod får föreslå nästa block i kedjan. Det finns ingen konsensusalgoritm för att välja blocket och ingen röstning av vilken som helst. Den valda noden föreslår ensidigt vad nästa block i blockkedjan kommer att bli. Men vad händer om den noden är skadlig? Tja, det finns en process för att hantera det, men den är implicit. Andra noder kommer implicit att acceptera eller förkasta det blocket genom att välja om de ska bygga ovanpå det eller inte.
Om de accepterar det blocket kommer de att signalera att de accepterar genom att utöka blockkedjan inklusive accepterat block. Om de däremot avvisar det blocket, kommer de att förlänga kedjan genom att ignorera det blocket,
och bygga ovanpå vilket som är det föregående blocket som de accepterade. Kom ihåg att varje block

innehåller en hash av blocket som den utökar. Detta är den tekniska mekanismen som tillåter noder att signalera vilket block det är som de skjuter ut.

Bitcoin konsensusalgoritm (förenklad)

Denna algoritm är förenklad genom att den antar förmågan att välja en slumpmässig nod på ett sätt som är inte sårbara för Sybil-attacker.

1.
Nya transaktioner sänds till alla noder
2.
Varje nod samlar nya transaktioner till ett block
3.
I varje omgång får en slumpmässig nod sända sitt block
4.
Andra noder accepterar blocket endast om alla transaktioner i det är giltiga (oförbrukade, giltiga signaturer)
5.
Noder uttrycker sin acceptans av blocket genom att inkludera dess hash i nästa block de skapa

Låt oss nu försöka förstå varför denna konsensusalgoritm fungerar. För att göra detta, låt oss överväga hur a illvillig motståndare – som vi kallar Alice – kanske kan undergräva denna process.

Stöld Bitcoins. Can Alice stjäla helt enkelt Bitcoins tillhör en annan användare på en adress hon inte kontrollera? Nej. Även om det är Alices tur att föreslå nästa block i kedjan, kan hon inte stjäla andra användarnas bitcoins. Om du gör det skulle Alice behöva skapa en giltig transaktion som spenderar det myntet. Detta
skulle kräva att Alice förfalskar ägarnas signaturer vilket hon inte kan göra om en säker digital signatur schema används. Så så länge den underliggande kryptografin är solid, kan hon inte bara stjäla bitcoins.

Överbelastningsattack. Låt oss överväga en annan attack. Säg att Alice verkligen ogillar någon annan användare Bob.
Alice kan sedan besluta att hon inte kommer att inkludera några transaktioner som kommer från Bobs adress i någon
block som hon föreslår att komma in i blockkedjan. Med andra ord, hon nekar Bob tjänst.
Även om detta är en giltig attack som Alice kan försöka göra, är det lyckligtvis inget annat än en mindre

---

irritation. Om Bobs transaktion inte kommer in i nästa block som Alice föreslår, kommer han bara vänta tills en ärlig nod får chansen att föreslå ett block och sedan kommer hans transaktion in det blocket. Så det är inte riktigt en bra attack heller.

Dubbel spendera attack. Alice kan försöka lansera en dubbel spendera attack. För att förstå hur det fungerar, låt oss anta att Alice är kund hos någon onlinehandlare eller webbplats som drivs av Bob, som tillhandahåller någon onlinetjänst i utbyte mot betalning i bitcoins. Låt oss säga att Bobs tjänst tillåter nedladdningen

av viss programvara. Så här är hur en attack med dubbla utgifter kan fungera. Alice lägger till ett föremål till henne
kundvagn på Bobs hemsida och servern begär betalning. Sedan skapar Alice en Bitcoin
transaktionen från hennes adress till Bobs och sänder den till nätverket. Låt oss säga att några ärliga
noden skapar nästa block och inkluderar denna transaktion i det blocket. Så det finns nu ett block som
skapades av en ärlig nod som innehåller en transaktion som representerar en betalning från Alice till
köpmannen Bob.

Kom ihåg att en transaktion är en datastruktur som innehåller Alices signatur, en instruktion att betala till
Bobs publika nyckel och en hash. Denna hash representerar en pekare till en tidigare transaktionsutgång som Alice
fått och spenderar nu. Den pekaren måste referera till en transaktion som ingick i vissa
föregående block i konsensuskedjan.

Notera förresten att det finns två olika typer av hashpekare här som lätt kan förväxlas.
Blocken inkluderar en hash-pekare till föregående block som de utökar. Transaktioner inkluderar en eller
fler hash-pekare till tidigare transaktionsutdata som löses in.

Låt oss återvända till hur Alice kan starta en attack med dubbla utgifter. Det senaste blocket genererades av en
ärlig nod och inkluderar en transaktion där Alice betalar Bob för nedladdningen av programvaran. På
När Bob ser denna transaktion inkluderad i blockkedjan drar Bob slutsatsen att Alice har betalat honom och tillåter
Alice för att ladda ner programvaran. Antag att nästa slumpmässiga nod som väljs i nästa omgång
råkar kontrolleras av Alice. Nu eftersom Alice får föreslå nästa block, kan hon fria
ett block som ignorerar blocket som innehåller betalningen till Bob och istället innehåller en pekare till
föregående block. Dessutom, i blocket som hon föreslår, inkluderar Alice en transaktion som överförs
just de mynten som hon skickade till Bob till en annan adress som hon själv kontrollerar. Det här är en
klassiskt mönster med dubbla utgifter. Eftersom de två transaktionerna spenderar samma mynt, kan bara en av dem
ingå i blockkedjan. Om Alice lyckas inkludera betalningen till sin egen adress i
blockchain, då är transaktionen där hon betalar Bob värdelös eftersom den aldrig kan inkluderas senare i
blockkedjan.

. Figur 2,2 En dubbel spendera försök Alice skapar två transaktioner: ett i vilket hon skickar Bob
Bitcoins, och en sekund där hon dubbla spenderar dessa Bitcoins genom att skicka dem till en annan
adress som hon kontrollerar. Eftersom de spenderar samma Bitcoins, kan bara en av dessa transaktioner vara
ingår i blocket kedjan. Pilarna är pekare från ett block till det föregående blocket att det
sträcker sig inklusive en hash av det tidigare blocket i dess eget innehåll. C A
används för att beteckna ett mynt
ägs av Alice.

Och hur vet vi om detta försök med dubbla utgifter kommer att lyckas eller inte? Tja, det beror på
vilket block i slutändan kommer att hamna i den långsiktiga konsensuskedjan - den med Alice → Bob
transaktion eller den med Alice → Alice transaktion. Vad avgör vilket block som blir
ingår? Ärliga noder följer policyn att förlänga den längsta giltiga grenen, så vilken gren kommer

de förlänger? Det finns inget rätt svar! Vid denna tidpunkt är de två grenarna lika långa - de skiljer sig endast i det sista blocket och båda dessa block är giltiga. Noden som väljer nästa block sedan kan besluta att bygga på endera av dem, och detta val kommer till stor del avgöra om eller inte dubbelutgifterna lyckas.

En subtil punkt: ur moralisk synvinkel finns det en tydlig skillnad mellan blocket som innehåller transaktionen som betalar Bob och blocket som innehåller transaktionen där Alice dubbelt spenderar dessa mynt till hennes egen adress. Men denna distinktion är bara baserad på vår kunskap om historien som Alice betalade först Bob och försökte sedan dubbla utgifterna. Ur en teknisk synvinkel, dessa två transaktioner är dock helt identiska och båda blocken är lika giltiga. Noderna som tittar på detta har verkligen inget sätt att säga vilken som är den moraliskt legitima transaktionen.

I praktiken följer noder ofta en heuristik för att förlänga blocket som de först hörde talas om på peer-to-peer-nätverk. Men det är ingen fast regel. Och i alla fall, på grund av nätverkslatens, kunde det lätt vara att blocket som en nod först hörde talas om faktiskt är det som skapades som andra. Så det finns åtminstone en viss chans att nästa nod som får föreslå ett block kommer att förlänga blocket som innehåller den dubbla utgiften. Alice kan ytterligare försöka öka sannolikheten för att detta händer genom muta nästa nod för att göra det. Om nästa nod bygger på dubbelspend-blocket för vad som helst anledning, då kommer denna kedja nu att vara längre än den som inkluderar transaktionen till Bob. Vid denna

---

punkt, nästa ärliga nod är mycket mer sannolikt att fortsätta bygga på denna kedja eftersom den är längre. Denna process kommer att fortsätta och det kommer att bli allt mer sannolikt att blocket som innehåller dubbla utgifter kommer att vara en del av den långsiktiga konsensuskedjan. Blocket som innehåller transaktionen till
Bob, å andra sidan, blir helt ignoreras av nätet, och detta kallas nu en ***föräldralös block*** .

Låt oss nu ompröva hela den här situationen ur Bob-the-merchants synvinkel. Förstå hur Bob kan skydda sig från denna attack med dubbla utgifter är en viktig del av att förstå Bitcoin säkerhet. När Alice sänder transaktionen som representerar hennes betalning till Bob, lyssnar Bob på nätverket och hör om denna transaktion redan innan nästa block skapas. Om Bob var det ännu mer dumdristigt än vi tidigare beskrivit, kan han slutföra kassaprocessen på webbplats och låt Alice ladda ner programvaran direkt i det ögonblicket. Det kallas a ***noll-transaktionsbekräftelse*** . Detta leder till en ännu mer grundläggande dubbelutgiftsattack än den ena beskrivits tidigare. Tidigare, för att attacken med dubbla utgifter skulle inträffa, var vi tvungna att anta att en skadlig
aktören styr noden som föreslår nästa block. Men om Bob tillåter Alice att ladda ner programvara innan transaktionen får ens en enda bekräftelse på blockkedjan, då kan Alice sänder omedelbart en dubbel-utgiftstransaktion, och en ärlig nod kan inkludera den i nästa blockera istället för transaktionen som betalar Bob.

Figur 2,3 Bob Merchant uppfattning. Detta är vad Alices dubbel spendera försök ser ut från Bob handlarens synvinkel. För att skydda sig från denna attack bör Bob vänta tills transaktion som Alice betalar honom med ingår i blockkedjan och har flera bekräftelser.

Å andra sidan skulle en försiktig handlare inte släppa programvaran till Alice ens efter transaktionen inkluderades i ett block och skulle fortsätta att vänta. Om Bob ser att Alice lyckas

inleder en attack med dubbla pengar, inser han att blocket som innehåller Alices betalning till honom har varit föräldralös. Han borde överge transaktionen och inte låta Alice ladda ner programvaran. Istället, om det händer att trots dubbel-spend-försöket bygger nästa flera noder på blocket med Alice → Bob transaktion, då får Bob förtroende för att denna transaktion kommer att vara på lång sikt konsensuskedja.

I allmänhet gäller att ju fler bekräftelser en transaktion får, desto större är sannolikheten att den kommer att avslutas
upp i den långsiktiga konsensuskedjan. Kom ihåg att ärliga noders beteende alltid är att förlänga längsta giltiga gren som de ser. Chansen att den kortare grenen med den dubbla spendera kommer ikapp den längre grenen blir allt mindre när den blir längre än någon annan gren. Detta är särskilt sant om bara en minoritet av noderna är skadliga - för att en kortare gren ska komma ikapp, flera skadliga noder skulle behöva väljas i tät följd.

Faktum är att sannolikheten för dubbla utgifter minskar exponentiellt med antalet bekräftelser. Så om den transaktion som du är intresserad av har fått k bekräftelser, då sannolikheten för att en dubbla utgifter transaktion kommer att hamna på den långsiktiga konsensus kedjan går ner exponentiellt som en funktion av k . Den vanligaste heuristiken som används i Bitcoins ekosystem är att vänta på sex bekräftelser. Det är inget speciellt med nummer sex. Det är bara en bra avvägning mellan hur lång tid du måste vänta och din garanti för att transaktionen du är intresserad av hamnar i konsensusblockkedjan.

Sammanfattningsvis är skyddet mot ogiltiga transaktioner helt kryptografiskt. Men det upprätthålls av konsensus, vilket innebär att om en nod försöker inkludera en kryptografiskt ogiltig transaktion, den enda anledningen till att transaktionen inte hamnar i den långsiktiga konsensuskedjan är att en majoriteten av noderna är ärliga och kommer inte att inkludera en ogiltig transaktion i blockkedjan. På å andra sidan är skyddet mot dubbla utgifter enbart genom konsensus. Kryptografi har inget att göra säg om detta, och två transaktioner som representerar ett dubbelt utgiftsförsök är båda giltiga från en kryptografiskt perspektiv. Men det är konsensus som avgör vilken som kommer att hamna på långsiktig konsensuskedja. Och slutligen, du är aldrig 100 procent säker på att en transaktion du är intresserad av är på konsensusgrenen. Men denna exponentiella sannolikhetsgaranti är ganska bra. Efter ungefär sex transaktioner finns det praktiskt taget ingen chans att du kommer att gå fel.

2.4 Incitament och bevis på arbete

I föregående avsnitt fick vi en grundläggande titt på Bitcoins konsensusalgoritm och en bra intuition för varför vi tror att det är säkert. Men minns från början av kapitlet att Bitcoins decentralisering är dels en teknisk mekanism och dels smart incitamentteknik. Hittills har vi gjort det tittade mest på den tekniska mekanismen. Låt oss nu prata om incitamentstekniken det händer i Bitcoin.

Vi bad dig att ta ett steg i tro tidigare och anta att vi kan välja en slumpmässig nod och, kanske mer problematiskt, att åtminstone 50 procent av tiden, kommer denna process att välja en ärlig nod. Detta antagande om ärlighet är särskilt problematiskt om det finns ekonomiska incitament för deltagare för att undergräva processen, i vilket fall vi inte riktigt kan anta att en nod kommer att vara ärlig. Frågan blir då: kan vi ge noder ett incitament för att uppträda ärligt?

Tänk igen på försöket med dubbla utgifter efter en bekräftelse (Figur 2.3). Kan vi straffa,
61

---

på något sätt, noden som skapade blocket med dubbelutgiftstransaktionen? Tja, inte riktigt. Som vi nämnde tidigare är det svårt att veta vilken som är den moraliskt legitima transaktionen. Men även om vi gjorde det,
det är fortfarande svårt att straffa noder eftersom de inte har identiteter. Så låt oss istället vända på frågan och fråga, kan vi belöna var och en av de noder som skapade blocken som slutade på lång sikt konsensuskedja? Tja, igen, eftersom dessa noder inte avslöjar sina verkliga identiteter, kan vi inte riktigt posta dem kontanter till deras hemadresser. Om det bara fanns någon sorts digital valuta som vi kunde använd istället... du kan nog se vart detta är på väg. Vi kommer att använda bitcoins för att stimulera noder som skapade dessa block.

Låt oss pausa en stund. Allt som vi har beskrivit hittills är bara en abstrakt algoritm för uppnå distribuerad konsensus och är inte specifik för applikationen. Nu ska vi bryta oss ur den modellen, och vi kommer att använda det faktum att applikationen vi bygger genom detta distribuerad konsensusprocess är i själva verket en valuta. Specifikt kommer vi att uppmuntra noder till uppträda ärligt genom att betala dem i enheter av denna valuta.

Block Reward. Hur görs detta? Det finns två separata incitamentmekanismer i Bitcoin. Den första är den ***blocket belöning*** . Enligt reglerna för Bitcoin får noden som skapar ett block innehålla en speciell transaktion i det blocket. Denna transaktion är en myntskapande transaktion, analog med Skapa mynt i Scroogecoin, och noden kan också välja mottagaradressen för denna transaktion. Av naturligtvis kommer den noden vanligtvis att välja en adress som tillhör sig själv. Du kan tänka på detta som en betalning till noden i utbyte mot tjänsten att skapa ett block i konsensuskedjan.

När detta skrivs är värdet på blockbelöningen fast till 25 Bitcoins. Men det halveras faktiskt varje 210 000 block. Baserat på den hastighet av blockskapande som vi kommer att se inom kort, betyder detta att sjunker ungefär vart fjärde år. Vi är nu inne i andra perioden. Under de första fyra åren Bitcoins existens var blockbelöningen 50 bitcoins; nu är det 25. Och det kommer att fortsätta halveras. Detta har några intressanta konsekvenser, som vi kommer att se inom kort.

Du kanske undrar varför blockbelöningen stimulerar till ärligt beteende. Det kan visas, baserat på vad vi har sagt hittills, att den här noden får blockbelöningen oavsett om den föreslår en giltig blockering eller beter sig skadligt. Men detta är inte sant! Tänk på det - hur kommer denna nod att "samla" sin
pris? Det kommer bara att ske om blocket i fråga hamnar på den långsiktiga konsensusgrenen eftersom precis som alla andra transaktioner, kommer myntskapande transaktionen endast att accepteras av andra noder om det hamnar i konsensuskedjan. Det är nyckeltanken bakom denna incitamentmekanism. Det är en mycket subtilt men mycket kraftfullt trick. Det uppmuntrar noder att bete sig på vilket sätt de tror kommer få andra noder att utöka sina block. Så om större delen av nätverket följer den längsta giltiga grenen regeln, uppmuntrar den alla noder att fortsätta följa den regeln. Det är Bitcoins första incitamentmekanism.

Vi nämnde att var 210 000:e block (eller ungefär fyra år) halveras blockbelöningen.
I figur 2.4 kommer lutningen på denna kurva att fortsätta att halveras. Det här är en geometrisk serie, och du kanske
vet att det betyder att det finns en ändlig summa. Det går ut på totalt 21 miljoner bitcoins.

Sida 63

Figur 2,4 Blocket belöning skärs i hälften vart fjärde år begränsa den totala tillförseln av Bitcoins till 21 miljon.

Det är viktigt att notera att detta är det enda sättet på vilket nya bitcoins tillåts skapas. där är ingen annan myntgenereringsmekanism, och det är därför 21 miljoner är ett slutligt och totalt antal (som regler gäller nu, åtminstone) för hur många bitcoins det någonsin kan finnas. Denna nya belöning för att skapa block är
kommer faktiskt att ta slut 2140, som det ser ut nu. Betyder det att systemet kommer att stanna arbetar 2140 och blir osäkra eftersom noder inte längre har incitament att bete sig ärligt? Inte riktigt. Blockbelöningen är bara den första av två incitamentmekanismer i Bitcoin.

Transaktionsavgifter Den andra stimulansåtgärder kallas ***transaktionsavgift*** . Skaparen av någon transaktion kan välja att göra det totala värdet av transaktionsutgångarna mindre än det totala värdet av dess ingångar. Den som skapar blocket som först placerar transaktionen i blockkedjan kommer till samla in mellanskillnaden, som fungerar som en transaktionsavgift. Så om du är en nod som skapar ett block som innehåller, säg, 200 transaktioner, sedan betalas summan av alla dessa 200 transaktionsavgifter till adressen som du lägger i det blocket. Transaktionsavgiften är helt frivillig, men vi förväntar oss, baserat på vår förståelse för systemet, att när blockbelöningen börjar ta slut, kommer den att bli mer och mer viktigt, nästan obligatoriskt, för användare att ta med transaktionsavgifter för att få en rimlig kvalitet av tjänst. Till en viss grad börjar detta hända redan nu. Men det är ännu oklart exakt hur systemet kommer att utvecklas; det beror verkligen på mycket spelteori som inte har fungerat fullt ut ute än. Det är ett intressant område för öppen forskning inom Bitcoin.

Det finns fortfarande några problem kvar med konsensusmekanismen som vi beskrev den. Den första

Sida 64

Det viktigaste är det trosprång som vi bad dig ta att vi på något sätt kan välja en slumpmässig nod. För det andra har vi skapat ett nytt problem genom att ge noder dessa incitament för deltagande. Systemet kan bli instabila eftersom incitamenten orsakar en gratis för alla där alla vill köra en Bitcoin nod i hopp om att fånga några av dessa belöningar. Och en tredje är en ännu knepigare version av detta problem, som är att en motståndare kan skapa ett stort antal Sybil-noder för att försöka undergräva konsensusprocessen.

Mining and proof-of-arbete. Det visar sig att alla dessa problem är relaterade, och alla av dem har samma lösning, som kallas ***proof-of-arbete*** . Nyckeltanken bakom proof-of-work är att vi approximera valet av en slumpmässig nod genom att istället välja noder i proportion till en resurs att vi hoppas att ingen kan monopolisera. Om den resursen till exempel är datorkraft, då det är ett proof-of-work system. Alternativt kan det stå i proportion till ägandet av valutan, och som kallas ***proof-of-spel*** . Även om det inte används i Bitcoin, är proof-of-stake ett legitimt alternativ modell och den används i andra kryptovalutor. Vi kommer att se mer om proof-of-stake och annat arbetsbevisvarianter i 8 kap.

Men tillbaka till bevis-på-arbete. Låt oss försöka få en bättre uppfattning om vad det innebär att välja noder i proportion
till deras datorkraft. Ett annat sätt att förstå detta är att vi tillåter noder att konkurrera
med varandra genom att använda sin datorkraft, och det kommer att resultera i att noder automatiskt blir till
plockas i den proportionen. Ännu en syn på proof-of-work är att vi gör det måttligt svårt
att skapa nya identiteter. Det är en slags skatt på identitetsskapande och därför på Sybil-attacken. Detta
kan alla verka lite vaga, så låt oss gå vidare och titta på detaljerna i arbetsbevissystemet
som används i Bitcoin, vilket borde göra saker mycket tydligare.

Bitcoin uppnår proof-of-arbete med ***hash pussel*** . För att skapa ett block, noden som föreslår
att blockera krävs för att hitta ett nummer eller ***nonce*** , så att när du sammanfoga nonce, den
föregående hash, och listan över transaktioner som omfattar det blocket och tar hashen av denna helhet
sträng, då bör hash-utdata vara ett tal som faller in i ett målutrymme som är ganska litet
förhållande till det mycket större utdatautrymmet för den hashfunktionen. Vi kan definiera ett sådant målutrymme som
varje värde som faller under ett visst målvärde. I detta fall måste nonce uppfylla följande
olikhet:

(*nonce* || *föregående _ hash* || *tx* || *tx* ||... || *tx*)
arget
H
< *t*

Som vi såg tidigare innehåller normalt ett block en serie transaktioner som en nod föreslår. I
Dessutom innehåller ett block också en hash-pekare till föregående block . Dessutom kräver vi nu
[1]
att ett block också innehåller en nonce. Tanken är att vi vill göra det måttligt svårt att hitta en
nonce som uppfyller denna nödvändiga egenskap, vilket är att hasha ihop hela blocket, inklusive
att nonce, kommer att resultera i en viss typ av output. Om hash-funktionen uppfyller

[1] Vi använder termen hashpekare löst. Pekaren är bara en sträng i detta sammanhang eftersom den inte behöver berätta vart vi ska hitta detta block. Vi hittar blocket genom att fråga andra kamrater i nätverket om det. Den viktiga delen är hash som både fungerar som ett ID när vi begär andra peers för blocket och låter oss validera blocket när vi har fått det.

---

pusselvänlighetsegenskap från kapitel 1, då det enda sättet att lyckas lösa detta hashpussel
är att bara prova tillräckligt många nonces en efter en tills du har tur. Så specifikt, om detta målutrymme var
bara en procent av det totala utdatautrymmet, skulle du behöva prova cirka 100 nonces innan du fick
tur. I verkligheten är storleken på detta målutrymme inte alls lika hög som en procent av utmatningsutrymmet.
Det är mycket, mycket mindre än så som vi kommer att se inom kort.

Denna föreställning om hashpussel och bevis på arbete gör helt upp med kravet på att
Välj magiskt en slumpmässig nod. Istället konkurrerar noder helt enkelt oberoende om att lösa dessa
hashpussel hela tiden. Då och då kommer en av dem att ha tur och kommer att hitta en slumpmässig nonce det
uppfyller denna egenskap. Den lyckliga noden får sedan föreslå nästa block. Det är så systemet är
helt decentraliserat. Det är ingen som bestämmer vilken nod det är som får föreslå nästa
blockera.

Svårt att beräkna. Det finns tre viktiga egenskaper hash pussel. Den första är att de behöver

vara ganska svår att beräkna. Vi sa måttligt svårt, men du kommer att se varför detta faktiskt varierar med tid. Som i slutet av 2014, är svårigheten nivån ca 10^{20} hash per block. Med andra ord storleken av målet utrymmet är endast $1/10^{20}$ av storleken på utmatningsutrymmet för hashfunktionen. Det här är mycket av beräkning — det är utanför möjligheterna för en bärbar dator, till exempel. Därför att Detta är det bara vissa noder som bryr sig om att tävla i denna process för att skapa block. Denna process av upprepade gånger försöka och lösa dessa hash pussel kallas ***Bitcoin mining*** , och vi kallar deltagande noder ***gruvarbetare*** . Även om tekniskt sett vem som helst kan vara gruvarbetare har det funnits en hel del kraftkoncentrationen i gruvdriftens ekosystem på grund av de höga kostnaderna för gruvdrift.

Parameterizable kostnad. Den andra egenskapen är att vi vill att kostnaden för att vara parameterizable, inte en fast kostnad för alla tider. Sättet som uppnås är att alla noder i Bitcoin peer-to-peer nätverket kommer automatiskt att räkna om målet, det vill säga storleken på målutrymmet som en bråkdel av utgångsutrymmet, varje 2016-block. De räknar om målet på ett sådant sätt att den genomsnittliga tiden mellan på varandra följande block producerade i Bitcoin-nätverket är cirka 10 minuter. Med 10 minuter genomsnittlig tid mellan blocken, 2016 block fungerar till två veckor. Med andra ord omräkningen av målet sker ungefär varannan vecka.

Låt oss fundera på vad detta betyder. Om du är en gruvarbetare och du har investerat ett visst fast belopp hårdvara till Bitcoin-gruvdrift, men det övergripande gruv-ekosystemet växer, fler gruvarbetare kommer in, eller så distribuerar de snabbare och snabbare hårdvara, det betyder att under en tvåveckorsperiod, något fler block kommer att hittas än väntat. Så noder kommer automatiskt att justera målet, och mängden arbete som du måste göra för att kunna hitta ett block kommer att öka. Så om du lägger in ett fast belopp för hårdvaruinvesteringar, den hastighet med vilken du hittar block är faktiskt beroende av vad andra gruvarbetare gör. Det finns en mycket trevlig formel för att fånga detta, vilket är att sannolikheten att någon given gruvarbetare, Alice, kommer att vinna nästa block motsvarar bråkdelen av global hash makt som hon kontrollerar. Det betyder att om Alice har gruvhårdvara är det cirka 0,1 procent av det totala hashkraft kommer hon att hitta ungefär ett av 1 000 block.

Vad är syftet med denna omställning? Varför vill vi behålla denna 10-minuters invariant? De

---

anledningen är ganska enkel. Om blocken skulle komma väldigt nära varandra, då skulle det bli många ineffektivitet, och vi skulle förlora optimeringsfördelarna med att kunna placera många transaktioner i en enda block. Det finns inget magiskt med siffran 10, och om du gick ner från 10 minuter till 5 minuter, det skulle nog vara bra. Det har varit mycket diskussion om den idealiska blockfördröjningen som altcoins, eller alternativa kryptovalutor, borde ha. Men trots vissa oenigheter om idealisk latens, alla är överens om att det bör vara ett fast belopp. Det kan inte tillåtas att gå ner utan gräns. Det är därför vi har den automatiska målomräkningsfunktionen.

Sättet som denna kostnadsfunktion och arbetsbevis är inrättat gör att vi kan omformulera vår säkerhet antagande. Det är här vi äntligen lämnar det sista trosprånget som vi bad dig att ta tidigare. Istället för att säga att på något sätt är majoriteten av noder ärliga i ett sammanhang där noder inte ens har identiteter och att inte vara tydlig med vad det betyder, kan vi nu tydligt konstatera det många attacker på Bitcoin är omöjliga om majoriteten av gruvarbetare, viktat av hashkraft, är det följa protokollet - eller är ärliga. Detta är sant eftersom om en majoritet av gruvarbetare, viktat av hash

makt, är ärliga, tävlingen för att föreslå nästa block kommer automatiskt att säkerställa att det är en chans på minst 50 procent att nästa block som föreslås vid något tillfälle kommer från en ärlig nod.

Sidebar. Inom forskningsområden för distribuerade system och datasäkerhet, är det vanligt att anta att någon procentandel av noderna är ärliga och för att visa att systemet fungerar som det är tänkt även om de andra noderna beter sig godtyckligt. Det är i princip det tillvägagångssätt vi har tagit här, förutom att vi viktar noder med hashkraft vid beräkning av majoriteten. Den ursprungliga Bitcoin Whitepaper innehåller även denna typ av analys.

Men spelteorin ger en helt annan, och utan tvekan mer sofistikerad och realistiskt sätt att avgöra hur ett system kommer att bete sig. I den här uppfattningen delar vi inte upp noder i ärliga och skadlig. Istället antar vi att *varje* nod agerar i enlighet med sina stimulansåtgärder. Varje nod väljer en (randomiserad) strategi för att maximera dess utdelning, med hänsyn till andra noders potential strategier. Om protokollet och incitamenten är väl utformade, kommer de flesta noder att följa reglerna för det mesta. "Ärligt" beteende är bara en strategi av många, och vi fäster ingen speciell moral framträdande för det.

I den spelteoretiska synen är den stora frågan om det förinställda minerbeteendet är ett "Nash jämvikt", det vill säga om det representerar en stabil situation där ingen gruvarbetare kan inse en högre utdelning genom att avvika från ärligt beteende. Denna fråga är fortfarande omtvistad och ett aktivt område av forskning.

Att lösa hashpussel är sannolikt eftersom ingen kan förutsäga vilket nonce som kommer att resultera i lösa hashpusslet. Det enda sättet att göra det är att prova nonces en efter en och hoppas att man lyckas. Matematiskt är denna process som kallas **Bernoullis prövningar** . En Bernoulli-rättegång är ett experiment med två

---

möjliga utfall, och sannolikheten för att varje utfall inträffar är fixerad mellan på varandra följande försök. Här är de två utfallen huruvida hashen faller i målet eller inte, och att anta hashen funktioner beter sig som en slumpmässig funktion, är sannolikheten för dessa utfall fast. Typiskt noder prova så många nonces att Bernoulli-försök, en diskret sannolikhetsprocess, väl kan approximeras av en kontinuerlig sannolikhets process som kallas en **Poisson-process** , en process i vilken händelser inträffar oberoende med en konstant genomsnittlig takt. Slutresultatet av allt detta är att sannolikhetstätheten funktion som visar den relativa sannolikheten för tiden tills nästa block hittas ser ut som figur 2.5.

Figur 2,5 Sannolikhetstäthetsfunktionen för tiden till nästa block hittas.

Detta är känt som en exponentiell fördelning. Det finns en liten sannolikhet att om ett block har varit hittas nu, nästa block kommer att hittas mycket snart, säg inom några sekunder eller en minut. Och det finns också en liten sannolikhet att det kommer att ta lång tid, säg en timme, att hitta nästa block. Men totalt sett justerar nätverket automatiskt svårighetsgraden så att inter-block-tiden bibehålls på en genomsnittlig lång sikt på 10 minuter. Lägg märke till att figur 2.5 visar hur ofta blocken går

skapas av att hela nätverket inte bryr sig om vilken gruvarbetare som faktiskt hittar blocket.

Om du är en gruvarbetare är du förmodligen intresserad av hur lång tid det tar att hitta ett block. Vad gör ser denna sannolikhetstäthetsfunktion ut? Det kommer att ha samma form, men det kommer bara att ha en annan skala på x-axeln. Återigen kan det representeras av en trevlig ekvation.
För en specifik gruvarbetare:
ean tid till nästa block
m
=
10 *minuter*
fraktion av hash effekt

Om du har 0,1 procent av den totala nätverks-hash-kraften talar den här ekvationen om att du kommer att hitta block en gång var 10 000:e minut, vilket är bara ungefär en vecka. Det är inte bara din genomsnittliga tid mellan block kommer att vara mycket hög, men variansen i tiden mellan block som hittas av dig är kommer också att bli väldigt hög. Detta har några viktiga konsekvenser som vi ska titta på Kapitel 5.
67

Sida 68

Trivialt att verifiera. Låt oss nu gå över till den tredje viktiga egenskapen hos denna bevis på arbetsfunktion som är
att det är trivialt att verifiera att en nod har beräknat bevis på arbete korrekt. Även om det tar en nod, på genomsnitt, 10 20
försöker hitta en nonce som får blockhash att falla under målet, det måste vara det
publiceras som en del av blocket. Det är alltså trivialt för vilken annan nod som helst att titta på blockinnehållet, hash
dem alla tillsammans och verifiera att utmatningen är mindre än målet. Detta är ganska viktigt egendom eftersom det återigen gör att vi kan bli av med centraliseringen. Vi behöver ingen centraliserad myndighet som kontrollerar att gruvarbetare gör sitt jobb korrekt. Alla noder eller gruvarbetare kan omedelbart verifiera att ett block som hittats av en annan gruvarbetare uppfyller denna arbetsbevis-egenskap.

2.5 Att sätta ihop allt

Kostnad för gruvdrift. Låt oss nu titta på gruv ekonomi. Vi nämnde att det är ganska dyrt att arbeta som en gruvarbetare. Vid svårighet nuvarande nivå, att hitta en enda block tar beräkning omkring 10 20
hash och
blockbelöningen är cirka 25 Bitcoins, vilket är en ansenlig summa pengar vid den aktuella börsen Betygsätta. Dessa siffror möjliggör en enkel beräkning av om det är lönsamt för en att bryta, och vi kan fånga detta beslut med ett enkelt uttalande:

Om
gruv belöning > gruv kostnad
sedan gruvarbetare vinster
var
gruvbelöning = blockbelöning + tx-avgifter
gruv cost = hårdvara kostnads + driftskostnader (el, kylning, etc.)

I grund och botten är gruvbelöningen som gruvarbetaren får i form av blockbelöning och transaktion avgifter. Gruvarbetaren frågar sig själv hur det står sig i förhållande till de totala utgifterna, vilket är hårdvaran och elkostnad.

Men det finns några komplikationer med denna enkla ekvation. Den första är att, som du kanske har märkt, hårdvarukostnaden är en fast kostnad medan elkostnaden är en rörlig kostnad som uppstår över tid. En annan komplikation är att belöningen som gruvarbetare får beror på i vilken takt de hitta block, vilket beror inte bara på kraften i deras hårdvara, utan på förhållandet mellan deras hashhastighet till den totala globala hashhastigheten. En tredje komplikation är att kostnaderna som gruvarbetaren ådrar sig vanligtvis är
denominerade i dollar eller någon annan traditionell valuta, men deras belöning är denominerad i bitcoin. Så denna ekvation har ett dolt beroende av Bitcoins växelkurs vid varje given tidpunkt. Och Äntligen, hittills har vi antagit att gruvarbetaren är intresserad av att ärligt följa protokollet. Men miner kan välja att använda någon annan gruvstrategi istället för att alltid försöka utöka längsta giltiga gren. Så den här ekvationen fångar inte alla nyanser av de olika strategierna som gruvarbetaren kan anställa. Att faktiskt analysera om det är vettigt för mig är ett komplicerat spel

---

teoretiskt problem som inte är lätt att besvara.

Vid det här laget har vi fått en ganska bra förståelse för hur en Bitcoin uppnår decentralisering. Vi kommer nu att sammanfatta poängen på hög nivå och lägga ihop allt för att bli ännu bättre förståelse.

Låt oss utgå från identiteter. Som vi har lärt oss krävs inga verkliga identiteter för att delta i Bitcoin-protokollet. Alla användare kan skapa ett pseudonymt nyckelpar när som helst, vilket antal som helst dem. När Alice vill betala Bob i bitcoins anger inte Bitcoin-protokollet hur Alice lär sig Bobs adress. Med tanke på dessa pseudonyma nyckelpar som identiteter är transaktioner i grunden meddelanden som sänds till Bitcoin peer-to-peer-nätverket som är instruktioner för att överföra mynt från ett adress till en annan. Bitcoins är bara transaktionsutgångar, och vi kommer att diskutera detta mer i detalj i nästa kapitel.

Sidebar. Bitcoin har inte fasta valörer som amerikanska dollar, och i synnerhet finns det ingen specialbeteckning för "1 bitcoin." Bitcoins är bara transaktionsutgångar, och i de nuvarande reglerna, de kan ha ett godtyckligt värde med 8 decimaler med precision. Minsta möjliga värde är 0.00000001 BTC (Bitcoins), som kallas en **Satoshi** .

Målet med Bitcoin peer-to-peer-nätverket är att sprida alla nya transaktioner och nya block till alla Bitcoin peer-noder. Men nätverket är mycket ofullkomligt och gör sitt bästa försök att vidarebefordra denna informationen. Säkerheten i systemet kommer inte från perfektionen av peer-to-peer nätverk. Istället kommer säkerheten från blockkedjan och konsensusprotokollet som vi ägnade mycket av detta kapitel åt att studera.

När vi säger att en transaktion ingår i blockkedjan, menar vi egentligen att transaktionen har fått många bekräftelser. Det finns inget fast antal för hur många bekräftelser är nödvändiga innan vi är tillräckligt övertygade om att det inkluderas, men sex är en

vanlig heuristik. Ju fler bekräftelser en transaktion har fått, desto säkrare är du
kan vara att denna transaktion är en del av konsensuskedjan. Det kommer ofta att finnas föräldralösa block, eller block
som inte når in i konsensuskedjan. Det finns en mängd olika orsaker som kan leda till blockering
vara föräldralös. Blocket kan innehålla en ogiltig transaktion eller ett försök med dubbelt spenderande. Det kunde också bara vara ett resultat av nätverkslatens. Det vill säga, två gruvarbetare kan helt enkelt sluta med att hitta nya block
inom bara några sekunder från varandra. Så båda dessa block sändes nästan samtidigt
på nätverket, och en av dem kommer oundvikligen att bli föräldralös.

Till sist tittade vi på hashpussel och gruvdrift. Gruvarbetare är speciella typer av noder som bestämmer sig för att tävla i det här spelet att skapa nya block. De belönas för sin ansträngning när det gäller både nyligen myntade bitcoins (nyblocksbelöningen) och befintliga bitcoins (transaktionsavgifter), förutsatt att andra gruvarbetare bygger på sina block. En subtil men avgörande punkt: säg att Alice och Bob är två olika gruvarbetare, och Alice har 100 gånger så mycket datorkraft som Bob. Det betyder inte att Alice kommer att göra det
vinna alltid loppet mot Bob för att hitta nästa block. Istället har Alice och Bob en sannolikhetskvot

---

att hitta nästa block, i proportionen 100 till 1. På lång sikt kommer Bob att hitta i genomsnitt en procent av antalet block som Alice hittar.

Vi förväntar oss att gruvarbetare vanligtvis kommer att vara någonstans nära den ekonomiska jämvikten i den meningen
att de utgifter som de ådrar sig i form av hårdvara och el kommer att vara ungefär lika med
belöningar som de får. Anledningen är att om en gruvarbetare konsekvent gör en förlust, kommer hon förmodligen att göra det
sluta bryta. Å andra sidan, och om gruvdrift är mycket lönsamt med tanke på typisk hårdvara och el kostnader, då skulle mer gruvhårdvara komma in i nätverket. Den ökade hashhastigheten skulle leda till en ökad svårighetsgrad, och varje gruvarbetares förväntade belöning skulle sjunka.

Denna uppfattning om distribuerad konsensus genomsyrar Bitcoin ganska djupt. I en traditionell valuta, konsensus spelar in i viss begränsad utsträckning. Specifikt finns det en konsensusprocess som bestämmer valutakursen. Det är säkert sant i Bitcoin också; Vi behöver konsensus kring värdet på Bitcoin. Men i Bitcoin behöver vi dessutom konsensus om tillståndet för huvudboken, vilket är vad blockkedjan åstadkommer. Med andra ord, även redovisningen av hur många bitcoins du äger är föremål för konsensus. När vi säger att Alice äger en viss summa eller antal bitcoins, vad vi egentligen menar är att Bitcoin peer-to-peer-nätverket, som registrerats i blockkedjan, anser att summan av alla Alices adresser äger det antalet bitcoins. Det är sanningens ultimata natur i Bitcoin: ägande av bitcoins är inget annat än att andra noder håller med att en viss part äger dessa bitcoins.
Slutligen behöver vi konsensus om reglerna för systemet eftersom ibland reglerna för systemet systemet måste förändras. Det finns två typer av ändringar av reglerna för Bitcoin, känd som ***mjuka gafflar*** och ***hårda gafflar*** . Vi kommer att skjuta upp denna diskussion om skillnaderna till senare kapitel i som vi kommer att diskutera dem i detalj.

Att få en kryptovaluta från marken. En annan subtil koncept är att ***bootstrapping*** . Det finns en knepigt samspel mellan tre olika idéer i Bitcoin: säkerheten i blockkedjan, hälsan hos

gruvdriftens ekosystem och valutans värde. Vi vill självklart att blockkedjan ska vara säker för att Bitcoin ska vara en lönsam valuta. För att blockkedjan ska vara säker måste en motståndare inte kunna det överväldiga konsensusprocessen. Detta innebär i sin tur att en motståndare inte kan skapa mycket gruvdrift noder och ta över 50 procent eller mer av det nya blockskapandet.

Men när kommer det att vara sant? En förutsättning är att ha ett sunt gruvdriftekosystem som till stor del består av ärliga, protokollföljande noder. Men vad är en förutsättning för det — när kan vi vara säkra på att mycket av gruvarbetare kommer att lägga mycket datorkraft på att delta i denna hashpussellösningstävling? Tja, de kommer bara att göra det om växelkursen för Bitcoin är ganska hög eftersom belöningarna som de får är denominerade i Bitcoins medan deras utgifter är i dollar. Så ju mer värdet på valutan går upp, desto mer incitament kommer dessa gruvarbetare att bli.

Men vad säkerställer ett högt och stabilt värde på valutan? Det kan bara hända om användare i allmänhet lita på säkerheten i blockkedjan. Om de tror att nätverket kan bli överväldigat kl när som helst av en angripare, kommer Bitcoin inte att ha ett stort värde som valuta. Så du har

---

detta sammankopplade ömsesidigt beroende mellan säkerheten i blockkedjan, en sund gruvdrift ekosystemet och växelkursen.

På grund av den cykliska karaktären hos detta trevägsberoende är existensen av var och en av dessa baserad på de andras existens. När Bitcoin först skapades, ingen av dessa tre existerade. Det fanns inga andra gruvarbetare än Nakamoto själv som körde gruvmjukvaran. Bitcoin hade inte mycket värde som valuta. Och blockkedjan var faktiskt osäker eftersom den fanns inte mycket gruvdrift pågår och vem som helst kunde lätt ha överväldigat denna process.

Det finns ingen enkel förklaring till hur Bitcoin gick från att inte ha någon av dessa egenskaper till att ha alla tre. Medieuppmärksamhet var en del av historien - ju mer folk hör om Bitcoin, det mer kommer de att bli intresserade av gruvdrift. Och ju mer de blir intresserade av gruvdrift, desto mer folk kommer att ha förtroende för säkerheten i blockkedjan eftersom det nu finns mer gruvdrift aktivitet pågår och så vidare. För övrigt måste varje ny Altcoin som vill lyckas också på något sätt lösa det här problemet med att dra sig upp i sina bootstraps.

51 procent attack. Slutligen, låt oss fundera över vad som skulle hända om konsensus misslyckades och det fanns faktiskt
ett ***51-procentigt angripare*** som kontrollerar 51 procent eller mer av gruveffekten i Bitcoin nätverket. Vi kommer att överväga en mängd olika möjliga attacker och se vilka som faktiskt kan utföras av en sådan angripare.

Först och främst, kan den här angriparen stjäla mynt från en befintlig adress? Som du kanske har gissat, svaret är nej, eftersom det inte är möjligt att stjäla från en befintlig adress om du inte undergräver kryptografi. Det räcker inte för att undergräva konsensusprocessen. Detta är inte helt självklart. Låt oss säg att 51 procent angriparen skapar ett ogiltigt block som innehåller en ogiltig transaktion som representerar att stjäla Bitcoins från en befintlig adress som angriparen inte kontrollerar och överföra dem till sin egen adress. Angriparen kan låtsas att det är en giltig transaktion och fortsätta bygga vidare på detta block. Angriparen kan till och med lyckas göra det till den längsta grenen. Men den andra ärlig noder kommer helt enkelt inte att acceptera detta block med en ogiltig transaktion och kommer att behålla mining baserat på det senaste giltiga blocket som de hittade i nätverket. Så vad som kommer att hända är det där

kommer att vara vad vi kallar en gaffel i kedjan.

Föreställ dig nu detta från angriparens synvinkel som försöker spendera dessa ogiltiga mynt och skicka dem till någon handlare Bob som betalning för vissa varor eller tjänster. Bob springer förmodligen en Bitcoin nod själv, och det kommer att vara en ärlig nod. Bobs nod kommer att avvisa den grenen som ogiltig eftersom den innehåller en ogiltig transaktion. Den är ogiltig eftersom signaturerna inte checkade ut. Så Bobs nod kommer helt enkelt att ignorera den längsta grenen eftersom det är en ogiltig gren. Och på grund av det, Det räcker inte med att undergräva konsensus. Du måste undergräva kryptografi för att stjäla bitcoins. Så vi dra slutsatsen att denna attack inte är möjlig för en 51-procentig angripare.

Vi bör notera att allt detta bara är ett tankeexperiment. Om det i själva verket fanns faktiska tecken på en 51:a procent attack, vad som troligen kommer att hända är att utvecklarna märker detta och reagerar på det. De kommer att uppdatera Bitcoin-mjukvaran, och vi kan förvänta oss att reglerna för systemet, inklusive

---

peer-to-peer-nätverk, kan förändras i någon form för att göra det svårare för denna attack att lyckas. Men vi kan inte riktigt förutse det. Så vi arbetar i en förenklad modell där en attack på 51 procent händer, men förutom det finns det inga ändringar eller justeringar av reglerna för systemet.

Låt oss överväga en annan attack. Kan den 51-procentiga angriparen undertrycka vissa transaktioner? Låt oss säga det finns någon användare, Carol, som angriparen verkligen inte gillar. Angriparen känner några av Carols adresser, och vill försäkra sig om att inga mynt som tillhör någon av dessa adresser möjligen kan vara tillbringade. Är det möjligt? Eftersom han kontrollerar konsensusprocessen för blockkedjan kan angriparen vägra helt enkelt att skapa några nya block som innehåller transaktioner från en av Carols adresser. De angriparen kan vidare vägra att bygga på block som innehåller sådana transaktioner. Det kan han dock inte förhindra att dessa transaktioner sänds till peer-to-peer-nätverket eftersom nätverket beror inte på blockkedjan, eller på konsensus, och vi antar att angriparen inte full kontroll över nätverket. Angriparen kan inte stoppa transaktionerna från att nå majoriteten av noder, så även om attacken lyckas kommer det åtminstone att vara uppenbart att attacken sker.

Kan angriparen ändra blockbelöningen? Det vill säga kan angriparen börja låtsas att blocket belöningen är istället för 25 Bitcoins, säg 100 Bitcoins? Detta är en förändring av reglerna för systemet, och eftersom angriparen inte kontrollerar kopiorna av Bitcoin-programvaran som alla ärliga noder är igång är detta inte heller möjligt. Detta liknar anledningen till att angriparen inte kan inkludera ogiltiga transaktioner. Andra noder kommer helt enkelt inte att känna igen ökningen av blockbelöningen, och angriparen kommer alltså inte att kunna spendera dem.

Slutligen, kan angriparen på något sätt förstöra förtroendet för Bitcoin? Tja, låt oss föreställa oss vad som skulle göra det
hända. Om det fanns en mängd olika försök med dubbla utgifter, situationer där noderna inte sträckte sig den längsta giltiga grenen och andra attackförsök, då kommer folk sannolikt att avgöra det Bitcoin fungerar inte längre som en decentraliserad reskontra som de kan lita på. Folk kommer att tappa förtroendet för
valutan, och vi kan förvänta oss att växelkursen för Bitcoin kommer att rasa. Faktum är att om det är det känt att det finns ett parti som kontrollerar 51 procent av hashmakten, då är det möjligt att folk kommer att förlora förtroendet för Bitcoin även om angriparen inte nödvändigtvis försöker starta några attacker. Så det

är inte bara möjligt, utan faktiskt sannolikt, att en 51-procentig angripare av något slag kommer att förstöra förtroendet för
valutan. Detta är faktiskt det huvudsakliga praktiska hotet om en attack på 51 procent någonsin skulle förverkligas.
Med tanke på mängden utgifter som motståndaren skulle behöva lägga på att attackera Bitcoin
och att uppnå en majoritet på 51 procent, ingen av de andra attackerna som vi beskrev är riktigt vettiga ur ekonomisk synvinkel.

Förhoppningsvis har du vid det här laget fått en riktigt bra förståelse för hur decentralisering är uppnås i Bitcoin. Du bör ha goda kunskaper om hur identiteter fungerar i Bitcoin, hur transaktioner sprids och valideras, peer-to-peer-nätverkets roll i Bitcoin, hur blockchain används för att uppnå konsensus, och hur hashpussel och gruvdrift fungerar. Dessa begrepp ger en solid grund och en bra startpunkt för att förstå mycket av det mer subtila detaljer och nyanser av Bitcoin, som vi kommer att se i de kommande kapitlen.

Vidare läsning

Bitcoin whitepaper:
. Nakamoto, Satoshi Bitcoin En peer-to-peer elektroniska cash system . (2008)
Den ursprungliga ansökan om arbetsbevis:
Back, Adam. Hashcash-a denial of service motåtgärd . (2002)
Paxos-algoritmen för konsensus:
. Lamport, Leslie Paxos på ett enkelt sätt . ACM Sigact News 32,4 (2001): 18-25.

Övningar

1. Varför kör gruvarbetare "fulla noder" som håller reda på hela blockkedjan medan Bob

[2]
handlaren kan komma undan med en "lite nod" som implementerar "förenklad betalningsverifiering", som behöver

att bara undersöka de sista blocken?

2. Om en skadlig internetleverantör helt kontrollerar en användares anslutningar, kan den starta en attack med dubbla utgifter

mot användaren? Hur mycket beräkningsansträngning skulle detta ta?

3. Betrakta Bob handlaren besluta huruvida eller inte att acceptera $C_A \rightarrow B$ transaktion. Vad Bob är

verkligen intresserad av är om den andra kedjan kommer ikapp eller inte. Varför kollar han då helt enkelt

hur många bekräftelser C A

→ B har tagit emot, istället för att beräkna skillnaden i längd mellan

de två kedjorna?

2 Detta gäller bara "solo" gruvarbetare som inte är en del av en gruvpool, men vi har inte diskuterat det ännu.

---

4. Även när alla noder är ärliga, kommer block ibland att bli föräldralösa: om två gruvarbetare Minnie och

Mynie upptäcker block nästan samtidigt, ingen av dem har tid att höra om den andras block

innan hon sänder sin.

4a. Vad avgör vems block som hamnar på konsensusgrenen?

4b. Vilka faktorer påverkar frekvensen av föräldralösa block? Kan du härleda en formel för kursen utifrån

dessa parametrar?

4c. Försök att empiriskt mäta denna kurs på Bitcoin-nätverket.

4d. Om Mynie hör om Minnies block precis innan hon ska upptäcka sitt, betyder det att hon

slösat bort hennes ansträngning?

4e. Får alla gruvarbetare sina block föräldralösa i samma takt, eller är vissa gruvarbetare drabbade

oproportionerligt?

5a. Hur kan en gruvarbetare etablera en identitet på ett sätt som är svårt att fejka? (dvs vem som helst kan säga vilken

block bröts av henne.)

5b. Om en gruvarbetare missköter sig, kan andra gruvarbetare "bojkotta" henne genom att vägra bygga på hennes block på en

löpande?

6a. Om vi antar att nätverkets totala hashkraft förblir konstant, vad är sannolikheten för att a

kommer blocket att hittas inom de närmaste 10 minuterna?

6b. Antag Bob handlaren vill ha en politik som order kommer att levereras inom x minuter efter

kvitto. Vilket värde på x bör Bob väljer så att med 99% konfidens 6 block kommer att vara

inom x minuter?

Kapitel 3: Mekanik för Bitcoin

Det här kapitlet handlar om mekaniken i Bitcoin. Medan vi i de två första kapitlen har pratat om ett relativt hög nivå, nu ska vi fördjupa oss i detalj. Vi ska titta på verkliga datastrukturer, verkliga skript, och försök lära dig detaljerna och språket för Bitcoin på ett exakt sätt för att ställa in allt det vi vill prata om i resten av den här boken. Det här kapitlet kommer att vara utmanande eftersom många detaljer kommer att flyga på dig. Du kommer att lära dig detaljerna och egenheter som gör Bitcoin till vad det är.
För att sammanfatta var vi slutade förra gången, ger Bitcoin-konsensusmekanismen oss ett tillägg ledger, en datastruktur som vi bara kan skriva till. När data väl har skrivits till den finns den där för alltid. Det finns
ett decentraliserat protokoll för att etablera konsensus om värdet av den reskontran, och det finns gruvarbetare som utför det protokollet och validerar transaktioner. Tillsammans ser de till det transaktioner är väl utformade, att de inte redan är förbrukade och att reskontran och nätverket kan fungera som en valuta. Samtidigt antog vi att det fanns en valuta för att motivera dessa

gruvarbetare. I det här kapitlet kommer vi att titta på detaljerna om hur vi faktiskt bygger den valutan, för att motivera
gruvarbetare som får hela denna process att hända.

3.1 Bitcoin-transaktioner

Låt oss börja med transaktioner, Bitcoins grundläggande byggsten. Vi kommer att använda en förenklad modell av en huvudbok för tillfället. Istället för block, låt oss anta att enskilda transaktioner läggs till till huvudboken en i taget.

Figur 3,1 en kontobaserad liggare

Hur kan vi bygga en valuta ovanpå en sådan reskontra? Den första modellen du kanske tänker på, vilket är faktiskt den mentala modellen många människor har för hur Bitcoin fungerar, är att du har en kontobaserat system. Du kan lägga till några transaktioner som skapar nya mynt och kreditera dem någon. Och sedan kan du överföra dem senare. En transaktion skulle säga något i stil med "vi är flyttar 17 mynt från Alice till Bob", och det kommer att signeras av Alice. Det är all information om

---

transaktion som finns i huvudboken. I figur 3.1, efter att Alice får 25 mynt i den första transaktion och sedan överför 17 mynt till Bob i den andra, skulle hon ha 8 Bitcoins kvar på sitt konto. Nackdelen med detta sätt att göra saker är att alla som vill avgöra om en transaktion är valid måste hålla reda på dessa kontosaldon. Ta en ny titt på figur 3.1. Gör Alice har de 15 mynten som hon försöker överföra till David? För att ta reda på detta måste du titta baklänges i tiden för alltid för att se varje transaktion som påverkar Alice, och om hennes nettosaldo eller inte vid den tidpunkt då hon försöker överföra 15 mynt till David är större än 15 mynt. Klart vi kan göra detta är lite mer effektivt med några datastrukturer som spårar Alices balans efter varje transaktion. Men det kommer att kräva mycket extra hushållning förutom själva huvudboken.

På grund av dessa nackdelar använder Bitcoin inte en kontobaserad modell. Istället använder Bitcoin en reskontra som bara håller reda på transaktioner som liknar ScroogeCoin i kapitel 1.

Figur 3,2 en transaktionsbaserad liggaren, som är mycket nära Bitcoin

Transaktioner anger ett antal ingångar och ett antal utgångar (återkalla PayCoins i ScroogeCoin). Du kan tänka på ingångarna som mynt som konsumeras (skapade i en tidigare transaktion) och utdata som mynt som skapas. För transaktioner där ny valuta präglas finns inga mynt som konsumeras (kom ihåg CreateCoins i ScroogeCoin). Varje transaktion har en unik identifierare. Utgångar indexeras med början med 0, så vi kommer att referera till den första utgången som "utgång 0". Låt oss nu arbeta oss igenom figur 3.2. Transaktion 1 har inga indata eftersom denna transaktion är skapa nya mynt, och den har en produktion på 25 mynt som går till Alice. Dessutom eftersom detta är en transaktion
där nya mynt skapas krävs ingen signatur. Låt oss nu säga att Alice vill skicka några av dessa mynt över till Bob. För att göra det skapar hon en ny transaktion, transaktion 2 i vår exempel. I transaktionen måste hon uttryckligen hänvisa till den tidigare transaktionen där dessa mynt kommer från. Här hänvisar hon till utgång 0 av transaktion 1 (i själva verket den enda utgången av transaktion 1), som tilldelade 25 bitcoins till Alice. Hon måste också ange utdataadresserna i transaktionen.

I det här exemplet specificerar Alice två utgångar, 17 mynt till Bob och 8 mynt till Alice. Och, naturligtvis, detta det hela är undertecknat av Alice, så att vi vet att Alice faktiskt godkänner denna transaktion.

Ändra adresser. Varför Alice måste skicka pengar till sig själv i detta exempel? Precis som mynt i ScroogeCoin är oföränderliga, i Bitcoin måste hela en transaktionsutgång konsumeras av annan transaktion, eller inget av det. Alice vill bara betala 17 bitcoins till Bob, men resultatet att hon

äger är värt 25 bitcoins. Så hon måste skapa en ny utgång där 8 bitcoins skickas tillbaka till själv. Det kan vara en annan adress än den som ägde de 25 bitcoinsna, men det måste det ägas av henne. Detta kallas en ***förändring adress*** .

Effektiv verifiering. När en ny transaktion läggs till huvudboken, hur lätt är det att kontrollera om det är giltig? I det här exemplet måste vi slå upp transaktionsutdata som Alice refererade till, se till att den har ett värde av 25 bitcoins och att den inte redan har spenderats. Söker upp transaktionen output är lätt eftersom vi använder hash-pekare. För att säkerställa att det inte har förbrukats måste vi skanna blockkedja mellan den refererade transaktionen och det senaste blocket. Vi behöver inte gå hela vägen tillbaka till början av blockkedjan, och det kräver inte att du behåller några ytterligare datastrukturer (även om, som vi kommer att se, ytterligare datastrukturer kommer att påskynda saker och ting).

Konsolidering medel. Liksom i ScroogeCoin, eftersom transaktioner kan ha många ingångar och många utgångar, att dela och slå samman värde är lätt. Säg till exempel att Bob fick pengar i två olika transaktioner — 17 bitcoins i en och 2 i en annan. Bob kanske säger att jag skulle vilja ha en transaktion som jag kan spendera senare där jag har alla 19 bitcoins. Det är enkelt - han skapar en transaktion med de två ingångarna och en output, där utdataadressen är en som han äger. Det låter honom konsolidera dessa två transaktioner.

Gemensamma betalningar. Likaså gemensamma betalningar är också lätt att göra. Säg att Carol och Bob båda vill betala
David. De kan skapa en transaktion med två ingångar och en utgång, men med de två ingångarna som ägs av två olika personer. Och den enda skillnaden från föregående exempel är att sedan de två utdata från tidigare transaktioner som görs anspråk på här kommer från olika adresser, den transaktionen kommer att behöva två separata signaturer - en av Carol och en av Bob.

Transaktions syntax. Konceptuellt det är egentligen allt som finns att en Bitcoin transaktion. Låt oss nu se hur det är representerat på en låg nivå i Bitcoin. I slutändan är varje datastruktur som skickas på nätverket en sträng bitar. Det som visas i figur 3.3 är på mycket låg nivå, men detta kompileras vidare till en kompakt binärt format som inte är läsbart för människor.

Figur 3,3 **En faktisk Bitcoin transaktion.**

Som du kan se i figur 3.3 finns det tre delar av en transaktion: vissa metadata, en serie indata, och en serie utgångar.

- ***Metadata*** . Det finns en del hushållsinformation - transaktionens storlek, antalet av ingångar och antalet utgångar. Det är hashen för hela transaktionen som fungerar som ett unikt ID för transaktionen. Det är det som gör att vi kan använda hashpekare till referens transaktioner. Äntligen finns det ett "lock_time"-fält, som vi kommer tillbaka till senare.
- ***Ingångar.*** Transaktions ingångar bildar en array, och varje ingång har samma form. En ingång anger en tidigare transaktion, så den innehåller en hash för den transaktionen, som fungerar som en hash pekare till det. Inmatningen innehåller också indexet för den tidigare transaktionens utdata, dvs görs anspråk på. Och så finns det en signatur. Kom ihåg att vi måste skriva under för att visa att vi faktiskt har förmågan att göra anspråk på de tidigare transaktionsutdata.
- ***Utgångar.*** Utgångarna är återigen en array. Varje utgång har bara två fält. De har var och en värde, och summan av alla utdatavärden måste vara mindre än eller lika med summan av alla ingångsvärden. Om summan av utdatavärdena är mindre än summan av ingångsvärdena skillnaden är en transaktionsavgift till gruvarbetaren som publicerar denna transaktion.
Och så finns det en rolig rad som ser ut som vad vi vill ska vara mottagaradressen. Varje utdata är tänkt att gå till en specifik publik nyckel, och det finns faktiskt något i det fältet det ser ut som om det är hash för en offentlig nyckel. Men det finns också en del andra saker som ser ut som en uppsättning kommandon. Det här fältet är faktiskt ett manus, och vi kommer att diskutera detta nu.

3.2 Bitcoin-skript

Varje transaktionsutgång anger inte bara en publik nyckel. Det specificerar faktiskt ett skript. Vad är ett manus, och varför använder vi skript? I det här avsnittet kommer vi att studera Bitcoin-skriptspråket och förstå varför ett skript används istället för att bara tilldela en publik nyckel.

Den vanligaste typen av transaktion i Bitcoin är att lösa in en tidigare transaktionsutgång genom att signera med rätt nyckel. I det här fallet vill vi att transaktionsutdata ska säga, "detta kan lösas in av en underskrift från ägaren av adress X.” Kom ihåg att en adress är en hash av en publik nyckel. Alltså bara att ange adressen X berättar inte för oss vad den publika nyckeln är och ger oss inte något sätt att kontrollera signatur! Så i stället måste den utgående transaktionen säga: ”det här kan lösas genom en publik nyckel som hash till X, tillsammans med en signatur från ägaren av den offentliga nyckeln." Som vi kommer att se är detta exakt

vad den vanligaste typen av skript i Bitcoin säger.

OP_DUP
OP_HASH160
69e02e18...
OP_EQUALVERIFY
OP_CHECKSIG

Figur 3,4. Ett exempel Pay-to-PubkeyHash skript, den vanligaste typen av utgångs skript i Bitcoin

Men vad händer med det här manuset? Vem kör det, och exakt hur fungerar denna sekvens av instruktioner genomdriva uttalandet ovan? Hemligheten är att ingångarna också innehåller skript istället för signaturer. För att verifiera att en transaktion löser in en tidigare transaktionsutgång korrekt, kombinerar vi den nya transaktionens ingångsskript och den tidigare transaktionens utdataskript. Vi sammanfogar dem helt enkelt, och det resulterande skriptet måste köras framgångsrikt för att transaktionen ska vara giltig. Dessa två manus kallas ***scriptPubKey*** och ***scriptSig*** grund i det enklaste fallet, produktionen script anger bara en offentlig nyckel (eller en adress till vilken den publika nyckeln hash) och inmatningsskriptet anger en signatur med den offentliga nyckeln. Det kombinerade skriptet kan ses i figur 3.5.

Bitcoin skriptspråk. Den skriptspråk byggdes speciellt för Bitcoin, och bara kallas "Script" eller Bitcoin-skriptspråket. Det har många likheter med ett språk som heter Forth, vilket är ett gammalt, enkelt, stackbaserat programmeringsspråk. Men du behöver inte förstå Forth förstå Bitcoin-skript. De viktigaste designmålen för Script var att ha något enkelt och kompakt men ändå med inbyggt stöd för kryptografiska operationer. Så det finns till exempel specialinstruktioner för att beräkna hashfunktioner och för att beräkna och verifiera signaturer. Skriptspråket är stackbaserat. Detta innebär att varje instruktion exekveras exakt en gång, i en linjärt sätt. I synnerhet finns det inga loopar i Bitcoin-skriptspråket. Så antalet instruktioner i skriptet ger oss en övre gräns för hur lång tid det kan ta att köra och hur mycket minne den kan använda. Språket är inte Turing-komplett, vilket betyder att det inte har förmåga att beräkna godtyckligt kraftfulla funktioner. Och detta är designat - gruvarbetare måste köra dessa

skript, som skickas in av godtyckliga deltagare i nätverket. Vi vill inte ge dem befogenhet att skicka ett skript som kan ha en oändlig loop.

<sig>
<pubKey>

OP_DUP

OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG

Figur 3.5. För att kontrollera om en transaktion löser en utgång på rätt sätt, skapar vi en kombinerad manus av lägga till scriptPubKey för den refererade utdatatransaktionen (nederst) till scriptSig för inlösande transaktion (överst). Lägg märke till att <pubKeyHash?> innehåller ett '?'. Vi använder denna notation för att indikera att vi senare kommer att kontrollera för att bekräfta att detta är lika med hashen för den publika nyckeln som tillhandahålls i det förlösande manuset.

Det finns bara två möjliga utfall när ett Bitcoin-skript körs. Antingen körs det framgångsrikt utan fel, i vilket fall transaktionen är giltig. Eller om det finns något fel när skriptet är det körs kommer hela transaktionen att vara ogiltig och bör inte accepteras i blockkedjan.

Skriptspråket för Bitcoin är väldigt litet. Det finns bara plats för 256 instruktioner, eftersom var och en representeras av en byte. Av dessa 256 är 15 för närvarande inaktiverade och 75 är reserverade. Den reserverade instruktionskoder har inte tilldelats någon specifik betydelse ännu, men kan vara instruktioner som är det läggs till senare i tiden.

Många av de grundläggande instruktionerna är de du förväntar dig att vara i vilket programmeringsspråk som helst. Det finns grundläggande aritmetik, grundläggande logik — som 'om' och 'då' — , kasta fel, inte kasta fel, och återvänder tidigt. Slutligen finns det kryptoinstruktioner som inkluderar hashfunktioner, instruktioner för signaturverifiering, samt en speciell och viktig instruktion som heter CHECKMULTISIG som låter du kontrollerar flera signaturer med en instruktion. Figur 3.6 listar några av de vanligaste instruktioner i skriptspråket Bitcoin.

Den CHECKMULTISIG instruktion kräver att specificera *n* publika nycklar och en parameter *t* , för en tröskel. För denna instruktion att utföra giltigt, måste det finnas åtminstone *t* signaturer från *t* ur *n* av dem offentliga nycklar som är giltiga. Vi kommer att visa några exempel på vad du skulle använda multisignaturer till i nästa avsnitt, men det borde vara omedelbart klart att detta är en ganska kraftfull primitiv. Vi kan uttrycka i en kompakt sätt konceptet att *t* ur *n* specificerade enheter måste logga in för att transaktionen att vara giltig.

För övrigt finns det en bugg i multisignaturimplementeringen, och den har funnits där hela tiden. De CHECKMULTISIG-instruktionen visar ett extra datavärde från stacken och ignorerar det. Det här är bara en egenhet

---

Bitcoin-språket och man måste hantera det genom att lägga en extra dummyvariabel på stacken. Felet fanns i den ursprungliga implementeringen, och kostnaderna för att fixa det är mycket högre än skada den orsakar, som vi kommer att se senare i avsnitt 3.5. Vid denna tidpunkt anses denna bugg vara en funktion i Bitcoin, eftersom det inte försvinner.

OP_DUP
Duplicerar det översta objektet i stapeln
OP_HASH160
Hashes två gånger: först med SHA-256 och sedan RIPEMD-160
OP_EQUALVERIFY
Returnerar sant om indata är lika. Returnerar falskt och markerar transaktion som ogiltig om de är ojämlika

OP_CHECKSIG
Kontrollerar att ingångssignaturen är en giltig signatur med hjälp av den offentliga ingången nyckel för hash för den aktuella transaktionen
OP_CHECKMULTISIG
Kontrollerar att *k* underskrifter på transaktionen är giltiga signaturer från *k* av de angivna publika nycklar.
Figur 3.6 en lista över vanliga Script instruktioner och deras funktionalitet.
Köra ett skript. För att köra ett skript i ett programmeringsspråk stack-baserade, är allt vi behöver en bunt som vi kan pusha data till och poppa data från. Vi kommer inte att behöva något annat minne eller variabler. Det är vad gör det så beräkningsmässigt enkelt. Det finns två typer av instruktioner: datainstruktioner och op-koder. När en datainstruktion visas i ett skript, skjuts den informationen helt enkelt till toppen av stack. Opcodes, å andra sidan, utför en viss funktion och tar ofta som indata som ligger ovanpå stapeln.
Låt oss nu titta på hur Bitcoin-skriptet i figur 3.5 exekveras. Se figur 3.7, där vi visar stackens tillstånd efter varje instruktion. De två första instruktionerna i det här skriptet är data instruktioner – signaturen och den publika nyckeln som används för att verifiera signaturen – specificerad i scriptSig-komponenten av en transaktionsinmatning i den inlösande transaktionen. Som vi nämnde, när vi se en datainstruktion, vi skjuter den bara på stapeln. Resten av skriptet specificerades i scriptPubKey-komponenten i en transaktionsutdata i den refererade transaktionen.
Först har vi dubblettinstruktionen, OP_DUP, så vi trycker bara upp en kopia av den publika nyckeln överst av stapeln. Nästa instruktion är OP_HASH160, som säger till oss att poppa toppvärdet, beräkna dess kryptografisk hash, och tryck resultatet till toppen av stacken. När denna instruktion är klar körs kommer vi att ha ersatt den publika nyckeln på toppen av stacken med dess hash.

Figur 3,7 **Utförande av en Bitcoin skript.** På botten visar vi instruktionerna i skriptet. Data instruktioner betecknas med omgivande vinkelparenteser, medan opkoder börjar med "OP_". På upp, visar vi stacken precis efter att instruktionen har utförts.
Därefter ska vi göra ytterligare en push av data till stacken. Kom ihåg att dessa uppgifter specificerades av avsändaren av den refererade transaktionen. Det är hashen för en publik nyckel som avsändaren angav; de motsvarande privata nyckel måste användas för att generera signaturen för att lösa in dessa mynt. Vid denna punkt, det finns två värden överst i stacken. Det finns hash för den publika nyckeln, som specificeras av avsändaren och hashen för den publika nyckeln som användes av mottagaren när han försökte göra anspråk på mynt.
Vid det här laget kör vi kommandot EQUALVERIFY, som kontrollerar att de två värdena överst i stack är lika. Om de inte är det kommer ett fel att visas och skriptet slutar köras. Men i vår till exempel antar vi att de är lika, det vill säga att mottagaren av mynten använde rätt offentlig nyckel. Den instruktionen kommer att konsumera de två dataobjekten som är överst i stacken, And stacken innehåller nu två objekt — en signatur och den publika nyckeln.
Vi har redan kontrollerat att den här publika nyckeln faktiskt är den publika nyckeln som den refererade transaktionen till
specificerat, och nu måste vi kontrollera om signaturen är giltig. Detta är ett bra exempel på var Bitcoin skriptspråk är byggt med kryptografi i åtanke. Även om det är ett ganska enkelt språk i termer av logik, det finns några ganska kraftfulla instruktioner där, som denna "OP_CHECKSIG" instruktion. Denna enstaka instruktion tar bort de två värdena från stacken och gör hela signaturverifiering på en gång.
Men vad är detta en signatur för? Vad är ingången till signaturfunktionen? Det visar sig att det bara finns en sak du kan logga in i Bitcoin — en hel transaktion. Så "CHECKSIG"-instruktionen visar de två värden, den offentliga nyckeln och signaturen, utanför stacken, och verifierar att det är en giltig signatur för hela

transaktion med den offentliga nyckeln. Nu har vi kört varje instruktion i skriptet, och det finns inget kvar på traven. Förutsatt att det inte fanns några fel, kommer resultatet av detta skript helt enkelt att vara **sant, vilket** indikerar att transaktionen är giltig.

Vad används i praktiken. I teorin låter Script oss ange, i någon mening, godtyckliga villkor som måste uppfyllas för att spendera mynt. Men från och med idag används denna flexibilitet inte särskilt hårt. Om vi tittar på skript som faktiskt har använts i Bitcoins historia hittills, de allra flesta, 99,9 procent, är

---

exakt samma skript, vilket faktiskt är det skript som vi använde i vårt exempel. Som vi såg, detta manus anger bara en offentlig nyckel och kräver en signatur för den offentliga nyckeln för att spendera mynten.

Det finns några andra instruktioner som du kan använda. MULTISIG vänjer sig lite som en speciell typ av skript som heter Pay-to-Script-Hash som vi kommer att diskutera inom kort. Men förutom det, där har inte varit mycket mångfald när det gäller vilka manus som används. Detta beror på att Bitcoin-noder, som standard,

har en vitlista med standardskript, och de vägrar acceptera skript som inte finns på listan. Detta betyder inte att dessa skript inte kan användas alls; det gör dem bara svårare att använda. I själva verket detta distinktion är en mycket subtil punkt som vi återkommer till om lite när vi pratar om Bitcoin peer-to-peer-nätverk.

Bevis på brännskador. En proof-of-burn är ett skript som aldrig kan lösas. Skickar mynt till en proof-of-burn-skriptet fastställer att de har förstörts eftersom det inte finns något möjligt sätt för dem som ska spenderas. En användning av proof-of-burn är att starta upp ett alternativ till Bitcoin genom att tvinga människor att göra det

förstöra Bitcoin för att få mynt i det nya systemet. Vi kommer att diskutera detta mer i detalj i kapitel 10. Proof-of-burn är ganska enkel att implementera: OP_RETURN-opkoden ger ett fel om det någonsin nådde. Oavsett vilka värden du sätter före OP_RETURN kommer den instruktionen att köras så småningom, i vilket fall detta skript kommer att returnera falskt.

Eftersom felet kastas kommer data i skriptet som kommer efter OP_RETURN inte att bearbetas.

Så det här är en möjlighet för människor att lägga in godtyckliga data i ett skript, och därmed in i blockkedjan. Om,

av någon anledning vill du skriva ditt namn, eller om du vill tidsstämpla och bevisa att du visste vissa data vid en viss tidpunkt kan du skapa en Bitcoin-transaktion med mycket lågt värde. Du kan förstöra en mycket liten mängd valuta, men du får skriva vad du vill i blockkedjan, som

ska finnas kvar för alltid.

Pay-to-script-hash. En konsekvens av det sätt som Bitcoin skript fungerar är att avsändaren av mynt måste ange skriptet exakt. Men det här kan ibland vara ett ganska konstigt sätt att göra saker på. Säga, till exempel, du är en konsument som handlar online och du är på väg att beställa något. Och du säger, "Okej, jag är redo att betala. Säg till mig adressen dit jag ska skicka mina mynt." Säg nu att företaget som du beställer från använder MULTISIG-adresser. Sedan, eftersom den som spenderar coins måste specificera detta, återförsäljaren måste komma tillbaka och säga, "Åh, ja, vi gör något fancy nu. Vi använder MULTISIG. Vi kommer att be dig att skicka mynten till några komplicerade manus." Du kanske säger: "Jag vet inte hur man gör det. Det är för komplicerat. Som konsument, jag bara vill skicka till en enkel adress."

Bitcoin har en smart lösning på detta problem, och den gäller inte bara multi-sig-adresser utan alla komplicerade villkor som styr när mynt kan spenderas. Istället för att säga till avsändaren "skicka din mynt till hashen för denna publika nyckel, kan mottagaren istället säga till avsändaren "skicka dina mynt till hash av detta *script* . Ställ villkoret att för att lösa in dessa mynt är det nödvändigt att avslöja skriptet som har den givna hashen, och vidare tillhandahåller data som gör att skriptet utvärderas till sant." De avsändaren uppnår detta genom att använda transaktionstypen Pay-to-script-hash (P2SH), som har ovanstående semantik.

Sida 84

Specifikt hashar P2SH-skriptet helt enkelt det översta värdet på stacken, kontrollerar om det matchar tillhandahållit hash-värde, exekverar sedan ett speciellt andra steg av validering: det översta datavärdet från stack omtolkas som en sekvens av instruktioner och exekveras en andra gång som ett skript, med resten av stacken som input.

Att få stöd för P2SH var ganska komplicerat eftersom det inte var en del av Bitcoins ursprungliga design Specifikation. Det lades till i efterhand. Detta är förmodligen den mest anmärkningsvärda funktionen som har lagts till

till Bitcoin som inte fanns där i den ursprungliga specifikationen. Och det löser ett par viktiga problem.

Det tar bort komplexiteten från avsändaren, så mottagaren kan bara ange en hash som avsändaren skickar pengar till. I vårt exempel ovan behöver Alice inte oroa sig för att Bob använder multisig; hon bara skickar till Bobs P2SH-adress, och det är Bobs ansvar att ange det snygga manuset när han vill lösa in mynten.

P2SH har också en fin effektivitetsvinst. Gruvarbetare måste spåra uppsättningen av utdataskript som inte har gjorts

inlösta ännu, och med P2SH-utgångar är utdataskripten nu mycket mindre eftersom de bara anger en hash. All komplexitet skjuts till inmatningsskripten.

3.3 Tillämpningar av Bitcoin-skript

Nu när vi förstår hur Bitcoin-skript fungerar, låt oss ta en titt på några av de kraftfulla applikationer som kan realiseras med detta skriptspråk. Det visar sig att vi kan göra många snygga saker som kommer att motivera komplexiteten i att ha skriptspråket istället för att bara ange publika nycklar.

Spärrade transaktioner. Say Alice och Bob vill göra affärer med varandra - Alice vill betala Bob i Bitcoin för att Bob ska skicka några fysiska varor till Alice. Problemet är dock att Alice inte vill att betala förrän efter att hon har tagit emot varorna, men Bob vill inte skicka varorna förrän efter att han har fått det

betalats. Vad kan vi göra åt det? En trevlig lösning i Bitcoin som har använts i praktiken är att introducera en tredje part och göra en depositionstransaktion.

Spärrade transaktioner kan implementeras helt enkelt med MULTISIG. Alice skickar inte pengarna direkt till Bob, men skapar istället en MULTISIG-transaktion som kräver att två av tre personer skriver under för att lösa in mynten. Och de tre personerna kommer att vara Alice, Bob och någon tredje part skiljedomare, Judy, som kommer att träda i spel om det skulle uppstå någon tvist. Så Alice skapar en 2-av-3 MULTISIG transaktion som skickar några mynt hon äger och anger att de kan spenderas om några två av Alice, Bob och Judy sign. Denna transaktion ingår i blockkedjan, och vid denna tidpunkt, dessa mynt hålls i deposition mellan Alice, Bob och Judy, så att två av dem kan ange var mynten ska gå. Vid det här laget är Bob övertygad om att det är säkert att skicka varorna till Alice, så han kommer att posta dem eller leverera dem fysiskt. Nu i normalfallet är Alice och Bob båda ärliga. Så, Bob kommer att skicka över varorna som Alice väntar på, och när Alice tar emot varorna, Alice och Bob både underteckna en transaktion som löser in pengarna från deposition, och skickar dem till Bob. Lägg märke till att i

det här fallet där både Alice och Bob är ärliga behövde Judy aldrig engagera sig alls. Det fanns ingen tvist, och Alices och Bobs underskrifter uppfyllde 2-av-3-kravet för MULTISIG-transaktionen. Så

Sida 85

i normalfallet är detta inte så mycket mindre effektivt än att Alice bara skickar pengarna till Bob. Det kräver bara en extra transaktion i blockkedjan.

Men vad hade hänt om Bob inte faktiskt skickade varorna eller om de tappade vilse med posten? Eller kanske var varorna annorlunda än vad Alice beställde? Alice vill nu inte betala Bob eftersom hon tror att hon blivit lurad och hon vill få tillbaka sina pengar. Så Alice är definitivt kommer inte att underteckna en transaktion som frigör pengarna till Bob. Men Bob kan också förneka någon fel och vägrar att underteckna en transaktion som släpper pengarna tillbaka till Alice. Det är här Judy behöver engagera sig. Judy måste bestämma vem av dessa två personer som förtjänar pengarna.
Om Judy bestämmer sig för att Bob fuskade kommer Judy att vara villig att underteckna en transaktion tillsammans med Alice och skicka
pengar från deposition tillbaka till Alice. Alices och Judys underskrifter uppfyller 2-av-3-kravet MULTISIG transaktion, och Alice kommer att få tillbaka sina pengar. Och, naturligtvis, om Judy tror att Alice är på
fel här, och Alice vägrar helt enkelt att betala när hon borde, kan Judy underteckna en transaktion tillsammans med Bob, skicka pengarna till Bob. Så Judy bestämmer mellan de två möjliga resultaten. Men det fina Saken är att hon inte behöver vara inblandad om det inte finns en tvist.
Gröna adresser. En annan cool applikation är vad som kallas gröna adresser. Säg att Alice vill betala Bob och Bob är offline. Eftersom han är offline kan inte Bob gå och titta på blockkedjan för att se om en transaktionen som Alice skickar finns faktiskt där. Det är också möjligt att Bob är online men inte har tiden att gå och titta på blockkedjan och vänta på att transaktionerna ska bekräftas. Kom ihåg att vi normalt vill att en transaktion ska vara i blockkedjan och bekräftas av sex block, vilket tar upp till en timme, innan vi litar på att det verkligen finns i blockkedjan. Men för vissa varor sådana som mat kan Bob inte vänta en timme innan han levererar. Om Bob var en gatuförsäljare som sålde korv, så är det det
osannolikt att Alice skulle vänta i en timme för att få sin mat. Eller kanske Bob för någon annan reason har ingen anslutning till internet alls och kommer därför inte att kunna kontrollera blockkedja.
För att lösa detta problem med att kunna skicka pengar med hjälp av Bitcoin utan att mottagaren kan komma åt blockkedjan måste vi introducera en annan tredje part, som vi kommer att kalla banken (i praktiken det kan vara en börs eller någon annan finansiell mellanhand). Alice ska prata med sin bank och säga, "Hej, det är jag, Alice. Jag är din lojala kund. Här är mitt kort eller min legitimation. Och det skulle jag verkligen vilja
betala Bob här, kan du hjälpa mig?" Och banken kommer att säga: "Visst. Jag ska dra av lite pengar från ditt konto. Och upprätta en transaktion från en av mina gröna adresser till Bob."
Så lägg märke till att dessa pengar kommer direkt från banken till Bob. En del av pengarna förstås kan vara i en ändrad adress som går tillbaka till banken. Men i huvudsak betalar banken Bob här från en bankkontrollerad adress, som vi kallar en grön adress. Dessutom garanterar banken att det kommer inte dubbelt spendera dessa pengar. Så fort Bob ser att denna transaktion är undertecknad av banken, om han litar på bankens garanti att inte dubbla spendera pengarna, han kan acceptera att de pengarna kommer så småningom bli hans när det bekräftas i blockkedjan.
Observera att detta inte är en Bitcoin-försäkrad garanti. Detta är en verklig garanti, och i ordning för detta system för att fungera, Bob måste lita på att banken, i den verkliga världen, bryr sig om deras rykte,

---

och kommer inte att spendera dubbelt av den anledningen. Och banken kommer att kunna säga, "Du kan titta på min historia.
Jag har använt den här gröna adressen länge, och jag har aldrig spenderat dubbelt. Därför är jag väldigt det är osannolikt att göra det i framtiden." Så Bob behöver inte längre lita på Alice, som han kanske inte känner till handla om. Istället sätter han sitt förtroende till banken att de inte kommer att dubbla spendera pengarna som de skickade honom.
Naturligtvis, om banken någonsin gör dubbelutgifter, kommer folk att sluta lita på dess gröna adress(er). Faktiskt,

de två mest framträdande onlinetjänsterna som implementerade gröna adresser var Instawallet och Mt. Gox, och båda slutade med att kollapsa. Idag används inte gröna adresser särskilt mycket. När tanken var först föreslog, genererade det mycket spänning som ett sätt att göra betalningar snabbare och utan komma åt blockkedjan. Nu har folk dock blivit ganska nervösa inför idén och är det orolig att det sätter för mycket förtroende för banken.

Effektiva mikrobetalningar. Ett tredje exempel på Bitcoin skript är ett sätt att göra effektiva mikrobetalningar. Säg att Alice är en kund som ständigt vill betala Bob små summor pengar för vissa tjänst som Bob tillhandahåller. Till exempel kan Bob vara Alices leverantör av trådlösa tjänster och kräver henne att betala en liten avgift för varje minut hon pratar i telefonen.

Att skapa en Bitcoin-transaktion för varje minut som Alice pratar i telefon fungerar inte. Det kommer skapa för många transaktioner, och transaktionsavgifterna läggs upp. Om värdet av var och en av dessa transaktioner är i storleksordningen vad transaktionsavgifterna är, kommer Alice att betala ganska högt kostnad att göra detta.

Vad vi skulle vilja är att kunna kombinera alla dessa små betalningar till en stor betalning i slutet. Det visar sig att det finns ett bra sätt att göra detta på. Vi börjar med en MULTISIG transaktion som betalar det maximala beloppet som Alice någonsin skulle behöva spendera för en utgång som kräver att både Alice och Bob skriver under att släppa mynten. Nu, efter den första minuten som Alice har använt tjänsten, eller första gången Alice behöver göra en mikrobetalning, undertecknar hon en transaktion som spenderar de mynt som skickades till MULTISIG-adress, skickar en betalningsenhet till Bob och återlämnar resten till Alice. Efter nästa minut av att använda tjänsten, Alice undertecknar en annan transaktion, denna gång betalar två enheter till Bob och skickar resten till sig själv. Observera att dessa endast är signerade av Alice och inte har signerats av Bob ännu, inte heller publiceras de till blockkedjan. Alice kommer att fortsätta skicka dessa transaktioner till Bob varje minut som hon använder tjänsten. Så småningom kommer Alice att sluta använda tjänsten och berättar för Bob,

"Jag är klar, snälla avbryt min tjänst." Vid det här laget kommer Alice att sluta underteckna ytterligare transaktioner. På

När han hör detta kommer Bob att säga "Bra. Jag kopplar bort din tjänst och jag tar den sista transaktionen som du skickade mig, signera det och publicera det till blockkedjan."

Eftersom varje transaktion betalade Bob lite mer, och Alice lite mindre, den slutliga transaktionen som Bob löser in betalar honom i sin helhet för tjänsten han tillhandahållit och återbetalar resten av pengarna till Alice. Alla de transaktioner som Alice skrev under på vägen kommer inte att nå blockkedjan. Guppa behöver inte skriva under dem. De kommer bara att kasseras.

86

Tekniskt sett är alla dessa transaktioner dubbla utgifter. Så till skillnad från fallet med gröna adresser där vi specifikt försökte undvika dubbla utgifter, med en stark garanti, med detta mikrobetalningsprotokoll, genererar vi faktiskt en enorm mängd potentiella dubbla utgifter. I praxis, men om båda parter fungerar normalt, kommer Bob aldrig att underteckna någon transaktion utom den sista, i vilket fall blockkedjan faktiskt inte kommer att se några försök till en dubbel-utgift.

Det finns en annan knepig detalj: tänk om Bob aldrig undertecknar den sista transaktionen? Han kanske bara säger: "Det är jag gärna låta mynten sitta där i spärr för alltid", i så fall kanske mynten inte rör sig, men Alice kommer att förlora hela värdet som hon betalade i början. Det finns ett mycket smart sätt att undvika detta problem med att använda en funktion som vi nämnde kort tidigare och kommer att förklara nu.

Lås tiden. För att undvika detta problem, innan mikrobetalning protokoll kan även starta, Alice och Bob kommer båda att underteckna en transaktion som återbetalar alla Alices pengar till henne, men återbetalningen är "låst"

tills någon gång i framtiden. Så efter Alice signerar, men innan hon sänder, den första MULTISIG transaktion som placerar hennes pengar i deposition, hon vill få denna återbetalningstransaktion från Bob och hålla fast vid det. Som garanterar att om hon gör det då *t* och Bob har inte undertecknat någon av de små transaktioner som Alice har skickat, kan Alice publicera denna transaktion som återbetalar alla pengar direkt till henne.

Vad betyder det att det är låst tills tiden *t* ? Minns när vi tittade på metadata i Bitcoin transaktioner, att det fanns den här parametern lock_time, som vi hade lämnat oförklarad. Hur det är fungerar är att om du anger något annat värde än noll för låstiden, säger det till gruvarbetare att inte publicera transaktionen fram till angiven låstid. Transaktionen kommer att vara ogiltig före antingen en specifik blocknummer, eller en specifik tidpunkt, baserat på de tidsstämplar som sätts i block. Så detta är en sätt att förbereda en transaktion som bara kan spenderas i framtiden om den inte redan är spenderad då. Det fungerar ganska bra i mikrobetalningsprotokollet som en säkerhetsventil för att Alice ska veta det om Bob aldrig tecken, så småningom kommer hon att kunna få tillbaka sina pengar.

Förhoppningsvis har dessa exempel visat dig att vi kan göra några snygga saker med Bitcoin-skript. Vi diskuterade tre enkla och praktiska exempel, men det finns många andra som har undersökts.

En av dem är lotterier för flera spelare, ett mycket komplicerat flerstegsprotokoll med många transaktioner att ha olika låstider och depositioner i fall folk fuskar. Det finns också några snygga protokoll som använda skriptspråket för att låta olika människor få ihop sina mynt och blanda dem, så att det är svårare att spåra vem som äger vilket mynt. Vi kommer att se det i detalj i kapitel 6.

Smarta kontrakt. Den allmänna termen för kontrakt som de som vi såg i det här avsnittet ärsmarta kontrakt. Dessa är kontrakt för vilka vi har en viss grad av teknisk efterlevnad i Bitcoin, medan de traditionellt upprätthålls genom lagar eller skiljedomstolar. Det är en riktigt cool funktion av Bitcoin som vi kan använda skript, gruvarbetare och transaktionsvalidering för att realisera depositionsprotokollet eller
mikrobetalningsprotokollet utan att behöva en centraliserad auktoritet.

Forskning om smarta kontrakt går långt utöver de tillämpningar som vi såg i det här avsnittet. Det finns många typer av smarta kontrakt som folk skulle vilja kunna genomdriva men som inte är det

---

stöds av Bitcoin-skriptspråket idag. Eller åtminstone har ingen kommit på en kreativ sätt att implementera dem. Som vi såg, med lite kreativitet kan du göra ganska mycket med Bitcoin manus som det ser ut för närvarande.

3.4 Bitcoin-block

Hittills i det här kapitlet har vi tittat på hur enskilda transaktioner konstrueras och löses in. Men som vi såg i kapitel 2 är transaktioner grupperade i block. Varför är detta? I grund och botten är det en optimering. Om gruvarbetare var tvungna att komma till konsensus om varje transaktion individuellt, den takt som nya transaktioner skulle kunna accepteras av systemet skulle vara mycket lägre. Dessutom en hashkedja av block är mycket kortare än en hashkedja av transaktioner skulle vara, eftersom ett stort antal transaktioner kan placeras i varje block. Detta kommer att göra det mycket mer effektivt att verifiera blockkedjans datastruktur.

Blockkedjan är en smart kombination av två olika hash-baserade datastrukturer. Den första är en hash kedja av block. Varje block har en blockrubrik, en hash-pekare till vissa transaktionsdata och en hash pekaren till föregående block i sekvensen. Den andra datastrukturen är ett träd per block av alla de transaktioner som ingår i det blocket. Detta är ett Merkle-träd och låter oss ha en sammanfattning av alla transaktioner i blocket på ett effektivt sätt. Som vi såg i kapitel 1, för att bevisa att en transaktion ingår i ett specifikt block, kan vi tillhandahålla en väg genom trädet vars längd är logaritmisk in antalet transaktioner i blocket. För att sammanfatta består ett block av rubrikdata följt av en lista med transaktioner ordnade i en trädstruktur.

Figur 3,8. Den Bitcoin blocket kedjan innehåller två olika hash strukturer. Den första är en hashkedja av block som länkar de olika blocken till varandra. Den andra är intern i varje block och är en Merkle Träd av transaktioner inom blocken.

Rubriken innehåller mest information relaterad till gruvpusslet som vi kort diskuterade i föregående kapitel och kommer att återkomma i kapitel 5. Kom ihåg att hashen för blockhuvudet måste starta med ett stort antal nollor för att blocket ska vara giltigt. Rubriken innehåller också en "nonce" som gruvarbetare kan ändras, en tidsstämpel och "bitar", vilket är en indikation på hur svårt det här blocket var att hitta. De header är det enda som hashas under gruvdrift. Så för att verifiera en kedja av block behöver vi bara göra titta på rubrikerna. Den enda transaktionsdata som ingår i rubriken är roten till transaktionsträd — fältet "mrkl_root".

```
"i":[
{
"prev_out":{
"hash":"000000.....0000000",
"n":4294967295
},
"coinbase":"..."
},
]
"ut":[
{
"value":"25.03371419",
"scriptPubKey":"OPDUPOPHASH160..."
}
]
```

Figur 3,9. **Coinbase transaktion.** En myntbastransaktion skapar nya mynt. Det löser inte en tidigare utdata, och den har en noll-hash-pekare som indikerar detta. Den har en myntbasparameter som kan innehålla godtyckliga uppgifter. Värdet på myntbastransaktionen är blockbelöningen plus alla transaktionsavgifter som ingår i detta block.

En annan intressant sak med block är att de har en speciell transaktion i Merkle-trädet som heter "myntbas"-transaktionen. Detta är analogt med CreateCoins i Scroogecoin. Så det är här skapandet av nya mynt i Bitcoin sker. Det ser mest ut som en normal transaktion men med flera skillnader: (1) den har alltid en enda ingång och en enda utgång, (2) ingången löser inte in en tidigare utdata och innehåller alltså en noll-hash-pekare, eftersom den präglar nya bitcoins och inte spendera befintliga mynt, (3) värdet på utmatningen är för närvarande lite över 25 Bitcoins. Utgången värde är gruvarbetarens inkomst från blocket. Den består av två komponenter: en platt gruvbelöning, som ställs in av systemet och som halveras vart 210 000:e block (cirka 4 år), och transaktionen avgifter som tas ut från varje transaktion som ingår i blocket. (4) Det finns en speciell "myntbas" parameter, vilket är helt godtyckligt — gruvarbetare kan lägga in vad de vill där.

Kända, i det allra första blocket som någonsin utvunnits i Bitcoin, refererade coinbase-parametern till en historia i tidningen Times of London där kanslern räddade banker. Detta har tolkats

---

som politisk kommentar till motivet för att starta Bitcoin. Det fungerar också som ett slags bevis på att första blocket bröts efter att historien kom ut den 3 januari 2009. Ett sätt som myntbasen

parametern har sedan använts är att signalera stöd från gruvarbetare för olika nya funktioner.
För att få en bättre känsla för blockformatet och transaktionsformatet är det bästa sättet att utforska blocket kedja dig själv. Det finns många webbplatser som gör dessa uppgifter tillgängliga, såsom blockchain.info . Du kan titta på grafen över transaktioner, se vilka transaktioner som löser in vilka andra transaktioner, titta för transaktioner med komplicerade skript, och titta på blockstrukturen och se hur block refererar till andra block. Eftersom blockkedjan är en offentlig datastruktur har utvecklare byggt vackra omslag till utforska det grafiskt.

3.5 Bitcoin-nätverket

Hittills har vi pratat om möjligheten för deltagare att publicera en transaktion och få in den blockkedja som om detta händer av magi. I själva verket sker detta genom Bitcoin-nätverket. Det är en peer-to-peer-nätverk, och det ärver många idéer från peer-to-peer-nätverk som har föreslagna för alla möjliga andra ändamål. I Bitcoin-nätverket är alla noder lika. Det finns inget hierarki, och det finns inga speciella noder eller masternoder. Den körs över TCP och har en slumpmässig topologi, där varje nod peers med andra slumpmässiga noder. Nya noder kan gå med när som helst. Faktiskt, du kan ladda ner en Bitcoin-klient idag, snurra upp din dator som en nod, och den kommer att ha lika rättigheter och möjligheter som alla andra noder på Bitcoin-nätverket.
Nätverket förändras över tiden och är ganska dynamiskt på grund av att noder kommer in och ut. Det finns ingen uttryckligt sätt att lämna nätverket. Istället, om en nod inte har hörts från på ett tag — tre timmar är varaktigheten som är hårdkodad i de vanliga klienterna — andra noder börjar glömma det. På det här sättet, nätverket hanterar graciöst noder som går offline.
Kom ihåg att noder ansluter till slumpmässiga kamrater och att det inte finns någon geografisk topologi av något slag. Säg nu
du startar en ny nod och vill gå med i nätverket. Du börjar med ett enkelt meddelande till en nod som du känner till. Detta brukar kallas din ***säd nod*** , och det finns ett par olika sätt du kan slå upp listor över frönoder att försöka ansluta till. Du skickar ett speciellt meddelande och säger: "Berätta för mig adresser till alla andra noder i nätverket som du känner till." Du kan upprepa processen med de nya noderna lär du dig om hur många gånger du vill. Sedan kan du välja vilka du vill peer with, och du kommer att vara en fullt fungerande medlem av Bitcoin-nätverket. Det finns flera steg som involverar slumpmässighet, och det idealiska resultatet är att du är peered med en slumpmässig uppsättning noder. Till
gå med i nätverket, allt du behöver veta är hur du kontaktar en nod som redan finns på nätverket.
Vad är nätverket bra för? För att behålla blockkedjan, förstås. Så för att publicera en transaktion, vi vill få hela nätverket att höra om det. Detta sker genom en enkel ***översvämning*** algoritm ibland kallas en ***skvallerprotokoll*** . Om Alice vill betala Bob lite pengar, skapar hennes klient och henne noden skickar denna transaktion till alla noder som den har peerat med. Var och en av dessa noder kör en serie av kontroller för att avgöra om transaktionen ska accepteras eller inte. Om kontrollerna passerar, noden

---

skickar den i sin tur till alla dess peer-noder. Noder som hör om en transaktion lägger den i en pool av transaktioner som de har hört talas om men som inte finns i blockkedjan ännu. Om en nod hör om en transaktion som redan finns i poolen, sänder den inte den vidare. Detta säkerställer att översvämningen protokollet avslutas och transaktioner går inte runt i nätverket för alltid. Kom ihåg att varje transaktionen identifieras unikt av dess hash, så det är lätt att slå upp en transaktion i poolen.
När noder hör om en ny transaktion, hur avgör de om de ska spridas eller inte
Det? Det finns fyra kontroller. Den första och viktigaste kontrollen är transaktionsvalidering - transaktionen måste vara giltig med den aktuella blockkedjan. Noder kör skriptet för varje tidigare utdata löses in och se till att skripten returnerar sant. För det andra kontrollerar de att utgångarna är

inlösta här har inte redan förbrukats. För det tredje kommer de inte att vidarebefordra en redan sett transaktion, som
nämnde tidigare. För det fjärde, som standard kommer noder endast att acceptera och vidarebefordra "standard"-skript baserat på en
liten vitlista med skript.
Alla dessa kontroller är bara förnuftskontroller. Väluppförande noder implementerar alla dessa för att försöka behålla
nätverket är friskt och fungerar som det ska, men det finns ingen regel som säger att noder måste följa dessa specifika steg. Eftersom det är ett peer-to-peer-nätverk, och vem som helst kan gå med, finns det alltid möjligheten att en nod kan vidarebefordra dubbla utgifter, icke-standardiserade transaktioner eller direkt ogiltiga transaktioner. Det är därför varje nod måste göra kontrollen för sig själv.
Eftersom det finns latens i nätverket är det möjligt att noder kommer att få en annan syn på
väntande transaktionspool. Detta blir särskilt intressant och viktigt när det finns en
försökte dubbelt spendera. Låt oss säga att Alice försöker betala samma bitcoin till både Bob och Charlie,
och skickar ut två transaktioner ungefär samtidigt. Vissa noder kommer att höra om Alice →
Bob transaktionen först medan andra kommer att höra om Alice → Charlie transaktionen först. När en nod
hör någon av dessa transaktioner kommer den att lägga till den i sin transaktionspool, och om den hör om den andra
en senare kommer det att se ut som en dubbelspend. Noden släpper den senare transaktionen och vidarebefordrar den inte
eller lägg till den i transaktionspoolen. Som ett resultat kommer noderna tillfälligt att vara oense om vilka transaktioner
bör placeras i nästa block. Detta kallas ett rastillstånd.
Den goda nyheten är att det här är helt okej. Den som bryter nästa block kommer i princip att bryta
binda och bestäm vilken av dessa två pågående transaktioner som ska hamna permanent i en
blockera. Låt oss säga att Alice → Charlie-transaktionen kommer in i blocket. När noder med Alice →
Bob transaktion hör om detta block, kommer de att ta bort transaktionen från sina minnespooler eftersom det är en dubbel-utgift. När noder med Alice → Charlie-transaktionen hör om detta block kommer de att göra det släpp transaktionen från deras minnespooler eftersom den redan har hamnat i blockkedjan. Så
det kommer inte att bli mer oenighet när detta block sprids till nätverket.
Eftersom standardbeteendet är att noder hänger på vad de hör först, nätverksposition
frågor. Om två motstridiga transaktioner eller block annonseras på två olika positioner i
nätverk, kommer de båda att börja svämma över hela nätverket och vilken transaktion en nod ser först
kommer att bero på var den är i nätverket.
91

Sida 92

Naturligtvis förutsätter detta att varje nod implementerar denna logik där de behåller vad de än hör
först. Men det finns ingen central myndighet som upprätthåller detta, och noder är fria att implementera vilken annan logik som helst
de vill välja vilka transaktioner som ska behållas och om de ska vidarebefordra en transaktion eller inte. Väl titta närmare på incitament för gruvarbetare i kapitel 5.
Sidofält: Noll-bekräftelsetransaktioner och ersätt-med-avgift. I kapitel 2 tittade vi på
nollbekräftelsetransaktioner, där mottagaren accepterar transaktionen så fort den är
sänds på nätet. Detta är inte utformat för att vara säkert mot dubbla utgifter. Men som vi såg,
standardbeteendet för gruvarbetare vid motstridiga transaktioner är att inkludera transaktionen
de fick först, och detta gör dubbelutgifter mot transaktioner med nollbekräftelse
måttligt hårt. Som ett resultat, och på grund av deras bekvämlighet, har nollbekräftelsetransaktioner
bli vanligt.
Sedan 2013 har det funnits intresse för att ändra standard policy att *ersätta-by-avgift* (RBF), varigenom

noder kommer att ersätta en pågående transaktion i sin pool om de hör en motstridig transaktion som inkluderar en högre avgift. Detta är det rationella beteendet för gruvarbetare, åtminstone i en kortsiktig mening, eftersom det
ger dem en bättre avgift. Däremot skulle ersätta-med-avgift göra dubbla utgifter mot nollbekräftelseattacker mycket lättare i praktiken.
Ersätt-med-avgift har därför väckt kontroverser, både när det gäller den tekniska frågan om om det är möjligt att förhindra eller avskräcka dubbelutgifter i en RBF-värld, och den filosofiska frågan om Bitcoin ska försöka stödja nollbekräftelse så gott det går, eller överge det.
Vi kommer inte att dyka in i den långvariga kontroversen här, men Bitcoin har nyligen antagit "opt-in" RBF där transaktioner kan markera sig själva (med hjälp av sekvensnummerfältet) som kvalificerade för ersättas av transaktioner med högre avgifter.
Hittills har vi mest diskuterat spridning av transaktioner. Logiken för att tillkännage nya block, när gruvarbetare hittar ett nytt block, är nästan exakt samma sak som att sprida en ny transaktion och det är alla föremål för samma tävlingsvillkor. Om två giltiga block bryts samtidigt, bara ett av dessa kan inkluderas i den långsiktiga konsensuskedjan. I slutändan, vilket av dessa block kommer att vara inkluderade kommer att bero på vilka block de andra noderna bygger ovanpå, och den som inte får in i konsensuskedjan kommer att bli föräldralös.
Att validera ett block är mer komplext än att validera transaktioner. Förutom att validera rubriken och se till att hashvärdet ligger inom det acceptabla intervallet måste noder validera varje transaktion ingår i blocket. Slutligen kommer en nod att vidarebefordra ett block endast om den bygger på den längsta grenen, baserat
på dess perspektiv på hur blockkedjan (som egentligen är ett träd av block) ser ut. Detta undviker gafflar Bygga upp. Men precis som med transaktioner kan noder implementera olika logik om de vill - de kan vidarebefordra block som inte är giltiga eller block som bygger på en tidigare punkt i blockkedjan. Detta skulle bygga en gaffel, men det är okej. Protokollet är utformat för att motstå det.
92

Sida 93

Figur 3,10 **Block utbredningstid.** Denna graf visar den genomsnittliga tid som det tar ett block till räckvidd olika procentandelar av noderna i nätverket.
Vad är latensen för översvämningsalgoritmen? Grafen i figur 3.10 visar medeltiden för nya block för att spridas till varje nod i nätverket. De tre raderna visar den 25:e, den 50:e och den 75:e percentilblockets utbredningstid. Som du kan se är förökningstiden i princip proportionell till blockets storlek. Detta beror på att nätverkets bandbredd är flaskhalsen. De större blocken tar över 30 sekunder för att spridas till de flesta noder i nätverket. Så det är inte ett särskilt effektivt protokoll.
På Internet är 30 sekunder ganska lång tid. I Bitcoins design, att ha ett enkelt nätverk med liten struktur där noder är lika och kan komma och gå när som helst prioriterades framför effektivitet. Så ett block kan behöva gå igenom många noder innan det når de mest avlägsna noderna i nätverk. Om nätverket istället skulle designas uppifrån och ned för effektivitet skulle vi kunna se till att vägen mellan två valfria noder är kort.
Storlek på nätet. Det är svårt att mäta hur stor nätverket är eftersom det är dynamisk och det finns ingen central myndighet. Ett antal forskare har kommit med uppskattningar. I den övre delen, säger vissa att över en miljon IP-adresser under en given månad någon gång kommer att fungera, åtminstone tillfälligt, som en Bitcoin nod. Å andra sidan verkar det bara finnas cirka 5 000 till 10 000 noder som är permanent ansluten och fullständigt validera varje transaktion de hör. Detta kan verka som en förvånansvärt lågt antal, men när detta skrivs finns det inga bevis för att antalet fullt validerande noder går upp, och det kan faktiskt sjunka.
93

Lagrings krav. Fullt måste validera noder stanna permanent anslutna så att höra om alla uppgifterna. Ju längre en nod är offline, desto mer kommer den att behöva göra när den återansluter till nätverk. Sådana noder måste också lagra hela blockkedjan och behöver en bra nätverksanslutning till kunna höra varje ny transaktion och vidarebefordra den till peers. Lagringskravet finns för närvarande de låga tiotals gigabyte (se figur 3.11), väl inom kapaciteten hos en enskild datordator maskin.

Figur 3,11. **Storlek av blocket kedjan.** Fullt validerande noder måste lagra hela blockkedjan, vilket som i slutet av 2014 är över 26 gigabyte.

Slutligen måste fullständigt validerande noder upprätthålla hela uppsättningen av outnyttjade transaktionsutdata, vilket är de mynt som finns att spendera. Helst bör detta lagras i RAM, så att när du hör en ny föreslagen transaktion på nätverket kan noden snabbt slå upp de transaktionsutgångar den är försöker göra anspråk, köra skripten, se om signaturerna är giltiga och lägg till transaktionen i transaktionspool. I mitten av 2014 finns det över 44 miljoner transaktioner i blockkedjan, varav 12 miljoner är outnyttjade. Lyckligtvis är den fortfarande liten nog för att få plats med mindre än en gigabyte RAM-minne i en effektiv datastruktur.

Lätta noder. I motsats till fullo validera noder, det finns lätta noder, även kallad tunn klienter eller SPV-klienter (Simple Payment Verification). Faktum är att de allra flesta noder på Bitcoin nätverk är lätta noder. Dessa skiljer sig från helt validerande noder genom att de inte lagrar hela blockkedjan. De lagrar bara de bitar som de behöver för att verifiera specifika transaktioner som de bry sig om. Om du använder ett plånboksprogram, skulle det vanligtvis innehålla en SPV-nod. Noden laddar ner blockrubriker och transaktioner som representerar betalningar till dina adresser.

En SPV-nod har inte säkerhetsnivån för en fullständigt validerande nod. Eftersom noden har block headers kan den kontrollera att blocken var svåra att bryta, men den kan inte kontrollera att varje transaktionen som ingår i ett block är faktiskt giltig eftersom den inte har transaktionshistoriken och känner inte till mängden outnyttjade transaktionsutdata. SPV-noder kan endast validera transaktionerna

---

som faktiskt påverkar dem. Så de litar i princip på att de fullständigt validerande noderna har validerat alla de andra transaktionerna som finns där ute. Det här är ingen dålig säkerhetsavvägning. De antar där validerar helt noder där ute som gör det hårda arbetet, och det om gruvarbetare gick igenom problem att bryta detta block, vilket är en riktigt dyr process, de gjorde förmodligen också en del validering för att säkerställa att detta block inte skulle avvisas.

Kostnadsbesparingarna med att vara en SPV-nod är enorma. Blockrubrikerna är bara cirka 1/1 000 av storleken blockkedjan. Så istället för att lagra några tiotals gigabyte, är det bara några tiotals megabyte. Till och med a smartphone kan enkelt fungera som en SPV-nod i Bitcoin-nätverket.

Eftersom Bitcoin vilar på ett öppet protokoll skulle det helst finnas många olika implementeringar som interagerar sömlöst med varandra. På det sättet om det finns en dålig bugg i en, är det inte troligt att den kommer att tappa hela nätverket. Den goda nyheten är att protokollet har implementerats på nytt. där är implementeringar i C ++ och Go, och människor arbetar på en hel del andra. De dåliga nyheterna är att de flesta av noderna på nätverket kör bitcoind-biblioteket, skrivet i C++, underhållet av Bitcoin Core-utvecklarna, och några av dessa noder kör tidigare inaktuella versioner som har inte uppdaterats. I vilket fall som helst, de flesta kör någon variant av denna vanliga klient.

3.6 Begränsningar och förbättringar

Slutligen kommer vi att prata om några inbyggda begränsningar för Bitcoin-protokollet, och varför det är utmanande att
förbättra dem. Det finns många begränsningar hårdkodade i Bitcoin-protokollet, som valdes
när Bitcoin föreslogs 2009, innan någon riktigt hade någon aning om att det kunde växa till ett
globalt viktig valuta. Bland dem finns gränserna för den genomsnittliga tiden per block, storleken på
block, antalet signaturoperationer i ett block och valutans delbarhet, summan
antal Bitcoins och blockbelöningsstrukturen.
Begränsningarna för det totala antalet Bitcoins som finns, såväl som strukturen för gruvdriften
belöningar kommer mycket sannolikt aldrig att ändras eftersom de ekonomiska konsekvenserna av att ändra dem är för bra. Gruvarbetare och investerare har gjort stora satsningar på systemet förutsatt att Bitcoin belönar
struktur och det begränsade utbudet av Bitcoins kommer att förbli som det var planerat. Om det ändras kommer det att göra det
har stora ekonomiska konsekvenser för människor. Så samhället har i princip kommit överens om att dessa aspekter,
huruvida de var klokt utvalda eller inte, kommer inte att förändras.
Det finns andra förändringar som skulle tyckas göra alla bättre, eftersom någon initial design
val verkar inte helt rätt med facit i hand. Främst bland dessa är gränser som påverkar
systemets genomströmning. Hur många transaktioner kan Bitcoin-nätverket bearbeta per sekund?
Denna begränsning kommer från den hårdkodade gränsen för storleken på block. Varje block är begränsat till en megabyte, ungefär en miljon byte. Varje transaktion är minst 250 byte. Vi delar 1 000 000 med 250
se att varje block har en gräns på 4 000 transaktioner, och givet att block hittas ungefär var 10:e
minuter har vi cirka 7 transaktioner per sekund, vilket är allt som Bitcoin-nätverket kan
hantera. Det kan tyckas att att ändra dessa gränser skulle vara en fråga om att justera en konstant i en källa

---

kodfil någonstans. Men det är verkligen svårt att åstadkomma en sådan förändring i praktiken, av skäl som vi kommer att förklara inom kort.
Så hur jämförs sju transaktioner per sekund? Det är ganska lågt jämfört med genomströmningen av
någon större kreditkortsprocessor. Visas nätverk sägs hantera cirka 2 000 transaktioner per sekund
runt om i världen i genomsnitt och kan hantera 10 000 transaktioner per sekund under upptaget
perioder. Även Paypal, som är nyare och mindre än Visa, kan hantera 100 transaktioner per sekund kl
topptider. Det är en storleksordning mer än Bitcoin.
En annan begränsning som människor är oroliga för på lång sikt är att valen av kryptografiska
Algoritmer i Bitcoin är fixerade. Det finns bara ett par hashalgoritmer tillgängliga, och bara en
signaturalgoritm, ECDSA, över en specifik elliptisk kurva som kallas secp256k1. Det finns en viss oro
under Bitcoins livstid – vilket folk hoppas kommer att vara väldigt lång – kan den här algoritmen vara trasig.
Kryptografer kan komma med en smart ny attack som vi inte har förutsett vilket gör det
algoritm osäker. Detsamma gäller för hashfunktionerna; faktiskt, under det senaste decenniet hashfunktioner
har sett stadiga framsteg inom kryptoanalys. SHA-1, som ingår i Bitcoin, har redan några
kända kryptografiska svagheter, om än inte dödliga. För att ändra detta måste vi förlänga
Bitcoin skriptspråk för att stödja nya kryptografiska algoritmer.
Ändra protokollet. Hur kan vi gå om att införa nya funktioner i Bitcoin protokollet? Du
kanske tror att det här är enkelt - släpp bara en ny version av programvaran och säg till alla noder
uppgradering. I verkligheten är detta dock ganska komplicerat. I praktiken är det omöjligt att anta att varje
noden skulle uppgradera. Vissa noder i nätverket skulle misslyckas med att få den nya programvaran eller misslyckas med att få in den
tid. Konsekvenserna av att de flesta noder uppgraderas medan vissa noder kör den gamla versionen
beror mycket på hur förändringarna i programvaran är. Vi kan skilja på två
typer av förändringar: de som skulle orsaka en ***hård gaffel*** och de som skulle orsaka en ***mjuk gaffel*** .

Hårda gafflar. En typ av förändring som vi kan göra introducerar nya funktioner som tidigare anses ogiltig. Det vill säga, den nya versionen av programvaran skulle känna igen block som giltiga som gammal programvara skulle avvisa. Tänk nu på vad som händer när de flesta noder har uppgraderats, men några har inte. Snart kommer den längsta grenen att innehålla block som anses ogiltiga av de gamla noderna. Så de gamla noderna kommer att gå av och arbeta på en gren av blockkedjan som exkluderar block med den nya funktion. Tills de uppgraderar sin mjukvara, kommer de att betrakta sin (kortare) gren som den längsta giltig filial.

Denna typ av förändring kallas en hård gaffelbyte eftersom den gör att blockkedjan splittras. Varje nod i nätverket kommer att vara på den ena eller andra sidan av det baserat på vilken version av protokollet det är löpning. Självklart kommer grenarna aldrig att gå ihop igen. Detta anses oacceptabelt av community eftersom gamla noder effektivt skulle skäras bort från Bitcoin-nätverket om de inte uppgraderar deras programvara.

Mjuka gafflar. En annan typ av förändring som vi kan göra för Bitcoin lägger funktioner som gör valideringsregler strängare. Det vill säga, de begränsar uppsättningen av giltiga transaktioner eller uppsättningen av giltiga block såsom

---

att den gamla versionen skulle acceptera alla block, medan den nya versionen skulle avvisa några. Detta typ av förändring kallas en mjuk gaffel, och den kan undvika den permanenta splittringen som en hård gaffel introducerar.

Tänk på vad som händer när vi introducerar en ny version av programvaran med en mjuk gaffelbyte. Noderna som kör den nya mjukvaran kommer att genomdriva några nya, strängare regler. Förutsatt att majoriteten av noderna byter till den nya mjukvaran, dessa noder kommer att kunna genomdriva den nya regler. Att introducera en mjuk gaffel förlitar sig på att tillräckligt många noder byter till den nya versionen av protokollet som de kommer att kunna tillämpa de nya reglerna, med vetskapen om att de gamla noderna inte kommer att kunna tillämpa de nya regler eftersom de inte har hört talas om dem ännu.

Det finns en risk att gamla gruvarbetare kan bryta ogiltiga block eftersom de inkluderar vissa transaktioner som är ogiltiga enligt de nya, strängare reglerna. Men de gamla noderna kommer åtminstone att räkna ut att några av deras blocken avvisas, även om de inte förstår anledningen. Detta kan föranleda deras operatörer att uppgradera sin programvara. Dessutom, om deras gren blir omkörd av de nya gruvarbetarna, de gamla gruvarbetarna byter till det. Det beror på att block som anses giltiga av nya gruvarbetare också beaktas giltig av gamla gruvarbetare. Således kommer det inte att finnas en hård gaffel; istället blir det många små, tillfälliga gafflar.

Det klassiska exemplet på en förändring som gjordes via soft fork är pay-to-script-hash, som vi diskuterade tidigare i detta kapitel. Pay-to-script-hash fanns inte i den första versionen av Bitcoin-protokollet. Detta är en mjuk gaffel eftersom från de gamla nodernas synvinkel skulle en giltig pay-to-script-hash-transaktion fortfarande verifiera korrekt. Som tolkas av de gamla noderna är skriptet enkelt - det hashhar ett datavärde och kontrollerar om hashen matchar värdet som anges i utdataskriptet. Gamla noder vet inte att göra det (nu krävs) ytterligare steget att köra själva värdet för att se om det är ett giltigt skript. Vi litar på nya noder för att upprätthålla de nya reglerna, dvs att skriptet faktiskt löser in denna transaktion.

Så vad kan vi eventuellt lägga till med en mjuk gaffel? Pay-to-script-hash lyckades. Det är också möjligt att nya kryptografiska system kunde läggas till med en mjuk gaffel. Vi kan också lägga till lite extra metadata i coinbase-parametern som hade någon betydelse. Idag accepteras vilket värde som helst i myntbasparameter. Men vi skulle i framtiden kunna säga att myntbasen måste ha något specifikt formatera. En idé som har föreslagits är att i varje nytt block innehåller myntbasen Merkle

roten av ett träd som innehåller hela uppsättningen outnyttjade transaktioner. Det skulle bara resultera i en mjuk gaffel,
eftersom gamla noder kan bryta ett block som inte hade den nödvändiga nya myntbasparametern som fick avvisades av nätverket, men de skulle komma ikapp och gå med i huvudkedjan som nätverket bryter.
Andra ändringar kan kräva en hård gaffel. Exempel på detta är att lägga till nya opkoder till Bitcoin, ändra gränserna för block- eller transaktionsstorlek, eller olika buggfixar. Fixar felet vi diskuterade tidigare, där MULTISIG-instruktionen poppar upp ett extra värde från stacken, skulle också kräva en hård gaffel. Det förklarar varför, även om det är en irriterande bugg, är det mycket lättare att lämna det i protokollet och få folk att arbeta runt det istället för att byta till Bitcoin. Hårda gaffelbyten, även om de skulle vara trevliga, är det mycket osannolikt att det händer i det nuvarande klimatet för Bitcoin. Men många av dessa idéer har testats och visat sig vara framgångsrika i alternativa kryptovalutor, som börjar om från början. Vi kommer att prata om dem mer i detalj i kapitel 10.
97

Sida 98

Sidofält: Bitcoins gåta med blockstorlek. På grund av Bitcoins växande popularitet, har det i början av 2016 blivit vanligt att 1-megabyte-utrymmet i block fylls upp inom perioden mellan block (särskilt när, på grund av en slumpmässig slump, ett block tar längre tid än 10 minuter att hitta) först, vilket resulterar i att vissa transaktioner måste vänta ett eller flera ytterligare block för att ta sig in blockkedjan. Att öka blockstorleksgränsen kräver en hård gaffel.
Frågan om och hur man adresserar blockkedjans begränsade bandbredd för transaktioner har gripit Bitcoin-gemenskapen. Diskussionen började för flera år sedan, men med små framsteg mot ett samförstånd har det gradvis blivit mer bittert och eskalerat till en cirkus. Vi ska diskutera Bitcoins gemenskap, politik och styrning i kapitel 7.
Beroende på lösningen av blockstorleksproblemet kan några av detaljerna i det här kapitlet blivit lite inaktuell. De tekniska detaljerna för att öka Bitcoins transaktionsbearbetning kapacitet är intressanta, och vi uppmuntrar dig att läsa mer online.
Vid det här laget bör du vara bekant med Bitcoins tekniska mekanik och hur en Bitcoin-nod fungerar. Men människor är inte Bitcoin-noder, och du kommer aldrig att köra en Bitcoin-nod ditt huvud. Så hur interagerar du som människa egentligen med detta nätverk för att få det att vara användbart som ett valuta? Hur hittar du en nod för att informera om din transaktion? Hur får du in Bitcoins byta mot kontanter? Hur lagrar du dina Bitcoins? Alla dessa frågor är avgörande för att bygga en valuta som faktiskt kommer att fungera för människor, i motsats till bara mjukvara, och vi kommer att svara på dessa frågor i nästa kapitel.

Vidare läsning

Online-resurser. I detta kapitel diskuterade vi en hel del tekniska detaljer, och du kan finna det svårt att absorbera dem alla på en gång. För att komplettera materialet i detta kapitel är det användbart att gå online och se några av de saker vi diskuterade i praktiken. Det finns många webbplatser som låter dig undersöka block och transaktioner och se hur de ser ut. En sådan "blockchain explorer" är webbplats blockchain.info .
En utvecklarfokuserad bok om Bitcoin som täcker de tekniska detaljerna väl (särskilt kapitel 5, 6, och 7):
Antonopoulos, Andreas M. Maste Bitcoin: låsa digitala cryptocurrencies . O'Reilly Media, 2014.
98

Sida 99

Övningar

1. **Transaktions validering** : Tänk på stegen i behandlingen Bitcoin transaktioner. Som av dessa steg är beräkningsmässigt dyra? Om du är en enhet som validerar många transaktioner (säg en gruvarbetare) vilken datastruktur kan du bygga för att påskynda verifieringen?
2. **Bitcoin-skript** : För följande frågor är du fri att använda icke-standardiserade transaktioner och op-koder som för närvarande är inaktiverade. Du kan använda <data> som en stenografi för att representera data värden som skjuts upp i stacken. För en snabb referens, se här: https://en.bitcoin.it/wiki/Script .
a. Skriv Bitcoin ScriptPubKey-skriptet för en transaktion som kan lösas in av någon som anger en kvadratrot från 1764.
b. Skriv ett motsvarande ScriptSig-skript för att lösa in din transaktion.
c. Anta att du vill utfärda en ny RSA facto utmaning genom att publicera en transaktion som kan lösas in av alla som kan faktorisera ett 1024-bitars RSA-nummer (RSA-tal är produkten av två stora, hemliga primtal). Vilka svårigheter kan du stöta på?
3. **Bitcoin script II** : Alice backpackar och är orolig för att hennes enheter innehåller privata nycklar blir stulen. Så hon skulle vilja lagra sina bitcoins på ett sådant sätt att de kan vara det lösas in med kunskap om endast ett lösenord. Följaktligen lagrar hon dem i följande ScriptPubKey-adress:
OP_SHA1
<0x084a3501edef6845f2f1e4198ec3a2b81cf5c6bc>
OP_EQUALVERIFY
a. Skriv ett ScriptSig-skript som kommer att lösa in denna transaktion. [Tips: det borde det bara vara en rad lång.]
b. Förklara varför detta inte är ett säkert sätt att skydda Bitcoins med hjälp av ett lösenord.
c. Skulle implementera detta med Pay-to-script-hash (P2SH) fixa säkerhetsproblemet/-en för dig identifierad? Varför eller varför inte?
4. **Bitcoin-skript III.**
a. Skriv en ScriptPubKey som kräver demonstration av en SHA-256-kollision för att lösa in.
b. (Hårt) skriv ett motsvarande ScriptSig som framgångsrikt kommer att lösa in denna transaktion.
5. **Bränning och kodning**
a. Vilka är några sätt att bränna bitcoins, dvs att göra en transaktion olösbar? Vilka av dessa tillåter ett bevis på brännskador, dvs övertygar alla observatörer om att ingen kan lösa in en sådan transaktion?
b. Vilka är några sätt att koda godtycklig data i blockkedjan? Vilka av dessa resultera i brända bitcoins?
[Tips: du har mer kontroll över innehållet i transaktionens "ut"-fält än kan dyka upp först.]
c. En användare kodade en del JavaScript-kod i blockkedjan. Vad kan ha varit en motivation för att göra detta?
6. **Gröna adresser:** Ett problem med gröna adresser är att det inte finns något straff mot dubbla utgifter inom själva Bitcoin-systemet. För att lösa detta bestämmer du dig för att designa ett altcoin

---

kallas "GreenCoin" som har inbyggt stöd för gröna adresser. Alla försök till dubbel utgifter från adresser (eller transaktionsutdata) som har betecknats som "gröna" måste ådra sig en ekonomisk påföljd på ett sätt som kan verkställas av gruvarbetare. Föreslå en möjlig design

för GreenCoin.
7. **SPV-bevis** : Anta att säljaren Bob driver en lättviktsklient och tar emot det aktuella huvudet av blockkedjan från en pålitlig källa.
a. Vilken information ska Bobs kunder tillhandahålla för att bevisa att deras betalning till Bob har inkluderats i blockkedjan? Anta att Bob kräver 6 bekräftelser.
b. Uppskatta hur många byte detta bevis kommer att kräva. Antag att det finns 1024 transaktioner i varje block.
8. **Lägga till nya funktioner** : Bedöm om följande nya funktioner kan läggas till med en hårddisk gaffel eller en mjuk gaffel:
a. Lägger till en ny OP_SHA3 skriptinstruktion
b. Inaktiverar OP_SHA1-instruktionen
c. Ett krav att varje gruvarbetare inkluderar en Merkle-rot av outnyttjade transaktionsutdata (UTXOs) i varje block
d. Ett krav på att alla transaktioner har sina utgångar sorterade efter värde i stigande beställa
9. **Mer gaffel**
a. Den mest framträdande Bitcoin hård gaffel var en övergående en som orsakas av version 0.8 bug . Hur många block övergavs när gaffeln löstes?
b. Den mest framträdande mjuka Bitcoin-gaffeln var tillägget av pay-to-script-hash. Hur många block blev föräldralösa på grund av det?
c. Bitcoin-klienter går in i "säkert läge" när de upptäcker att kedjan har splittrats. Vad heuristik(er) kan du använda för att upptäcka detta?

Kapitel 4: Hur man lagrar och använder Bitcoins

Det här kapitlet handlar om hur vi lagrar och använder bitcoins i praktiken.

4.1 Enkel lokal lagring

Låt oss börja med det enklaste sättet att lagra bitcoins, och det är helt enkelt att placera dem på en lokal enhet. Som en sammanfattning, för att spendera en bitcoin behöver du känna till en del offentlig information och en del hemlighet
information. Den offentliga informationen är vad som händer i blockkedjan - myntets identitet, hur mycket det är värt och så vidare. Den hemliga informationen är den hemliga nyckeln till ägaren av bitcoin, antagligen är det du. Du behöver inte oroa dig för mycket om hur du lagrar den offentliga informationen eftersom du alltid kan få tillbaka den när du behöver. Men den hemliga signeringsnyckeln är något du skulle vilja bättre hålla reda på. Så i praktiken handlar lagring av dina bitcoins om att lagra och hantera dina nycklar.
Att lagra bitcoins handlar egentligen om att lagra och hantera Bitcoins hemliga nycklar.
När man ska ta reda på hur man lagrar och hanterar nycklar finns det tre mål att tänka på. Den första är tillgänglighet: att faktiskt kunna spendera dina mynt när du vill. Det andra är säkerhet: att göra säker på att ingen annan kan spendera dina mynt. Om någon får makten att spendera dina mynt de kunde bara skicka dina mynt till sig själva, och då har du inte mynten längre. Det tredje målet är bekvämlighet, det vill säga nyckelhantering ska vara relativt lätt att göra. Som du kan föreställa dig, att uppnå alla tre samtidigt kan vara en utmaning.
Olika tillvägagångssätt för nyckelhantering erbjuder olika avvägningar mellan tillgänglighet, säkerhet, och bekvämlighet.
Den enklaste metoden för nyckelhantering är att lagra dem på en fil på din egen lokala enhet: din dator, din telefon eller någon annan typ av pryl som du bär, äger eller kontrollerar. Det här är bra för enkelhetens skull: ha en smartphone-app som gör det möjligt att spendera mynt med en enkel knapptryckning är svårt att slå. Men det här är inte bra för tillgänglighet eller säkerhet – om du tappar bort enheten, om enheten

kraschar, och du måste torka av skivan, eller om din fil blir skadad, dina nycklar går förlorade, och så är dina mynt. Likadant för säkerheten: om någon stjäl eller bryter sig in i din enhet, eller den blir infekterad med skadlig programvara kan de kopiera dina nycklar och sedan kan de skicka alla dina mynt till sig själva. Med andra ord, att lagra dina privata nycklar på en lokal enhet, särskilt en mobil enhet, är ungefär som bära runt pengar i plånboken eller i väskan. Det är nyttigt att ha lite pengar, men du vill inte bära med dig dina livsbesparingar eftersom du kan förlora dem, eller så kan någon stjäla dem. Så vad du vanligtvis gör är att lagra lite information/lite pengar i din plånbok och behålla de flesta av dina pengar någon annanstans.

101

Sida 3

Plånböcker . Om du lagrar dina bitcoins lokalt, skulle du vanligtvis använda plånboksmjukvara, som är programvara som håller reda på alla dina mynt, hanterar alla detaljer om dina nycklar och gör det bekvämt med ett trevligt användargränssnitt. Om du vill skicka bitcoins till ett värde av $4,25 till ditt lokala kafé plånboksprogramvara skulle ge dig ett enkelt sätt att göra det. Plånbok programvara är särskilt användbar eftersom du vill vanligtvis använda en hel massa olika adresser med olika nycklar kopplade till dem. Som du kanske minns är det enkelt att skapa ett nytt offentligt/privat nyckelpar, och du kan använda detta till förbättra din anonymitet eller integritet. Plånboksmjukvaran ger dig ett enkelt gränssnitt som berättar hur mycket finns i din plånbok. När du vill spendera bitcoins hanterar den detaljerna om vilka nycklar som ska användas och hur man genererar nya adresser och så vidare.

Kodningsnycklar: bas 58 och QR-koder . För att spendera eller ta emot bitcoins behöver du också ett sätt att byta en adress med den andra parten — adressen dit bitcoins ska skickas. Det finns två huvudsakliga sätt på vilka adresser kodas så att de kan kommuniceras från mottagare till spender: as en textsträng eller som en QR-kod.

För att koda en adress som en textsträng tar vi bitarna av nyckeln och konverterar den från ett binärt tal till ett basnummer 58. Sedan använder vi en uppsättning av 58 tecken för att koda varje siffra som ett tecken; detta är kallas base58 notation. Varför 58? För det är den siffran vi får när vi tar med versaler bokstäver, små bokstäver, såväl som siffror som tecken, men utelämna några som kan vara förvirrande eller kan se ut som en annan karaktär. Till exempel tas både stor bokstav "O" och noll ut eftersom de ser för lika ut. Detta gör att kodade adresser kan läsas upp via telefon eller läsa från tryckt papper och skriva in, om det skulle behövas. Helst sådana manuella metoder för kommunicerande adresser kan undvikas genom metoder som QR-koder, som vi nu diskuterar.

1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa

Adressen som fick den allra första Bitcoin-blockbelöningen i genesis-blocket, base58-kodad.

Figur 4,1 : **en QR-kod som representerar en faktisk Bitcoin adress.** Skicka gärna några bitcoins till oss.

Den andra metoden för att koda en Bitcoin-adress är som en QR-kod, en enkel typ av 2-dimensionell streckkod. Fördelen med en QR-kod är att du kan ta en bild på den med en smartphone och plånbok

102

Sida 4

programvara kan automatiskt förvandla streckkoden till en sekvens av bitar som representerar motsvarande Bitcoin-adress. Detta är användbart i en butik, till exempel: utcheckningssystemet kan visa en QR-kod så kan du betala med din telefon genom att skanna koden och skicka mynt till den adress. Det är också användbart för telefon-till-telefon-överföringar.

Fåfänga adresser . Vissa individer eller handlare vill ha en adress som börjar med några mänskligt meningsfull text. Till exempel har spelwebbplatsen Satoshi Bones användare skickar pengar till

adresser som innehåller strängen "ben" i positionerna 2--6, som t.ex
1bonesEeTcABPjLzAb1VkFgySY6Zqu3sX(alla vanliga adresser börjar med tecknet 1,
anger pay-to-pubkey-hash.)

Vi sa att adresser är utdata från en hashfunktion, som producerar data som ser slumpmässigt ut, så hur kom strängen "ben" in där? Om Satoshi Bones bara gjorde upp dessa adresser, saknas förmågan att invertera hashfunktion, skulle de inte känna till motsvarande privata nycklar och därmed skulle faktiskt inte kontrollera dessa adresser. Istället genererade de flera gånger privata nycklar tills de hade tur och hittade en som hashed till detta mönster. Sådana adresser kallas *fåfänga adresser* och det finns verktyg för att skapa dem.
Hur mycket arbete tar detta? Eftersom det finns 58 möjligheter för varje karaktär, om du vill hitta en adress som börjar med en viss k -tecken sträng, måste du skapa 58^k adresser på genomsnitt tills du har tur. Så det hade krävts att hitta en adress som börjar med "ben". genererar över 600 miljoner adresser! Detta kan göras på en vanlig bärbar dator nuförtiden. Men det blir exponentiellt svårare för varje extra karaktär. Att hitta ett 15-teckens prefix skulle kräva en omöjlig mängd beräkning och (utan att hitta ett avbrott i den underliggande hashfunktionen) borde vara omöjligt.
Sidebar : **Snabbare fåfänga adress generation.** I Bitcoin, om vi kallar den privata nyckeln x , allmänheten Nyckeln är g^x . Exponentieringen representerar vad som kallas skalär multiplikation i en elliptisk kurva grupp. Adressen är $H(g^x)$, hashen av den publika nyckeln. Vi kommer inte in på detaljerna här, men exponentiering är det långsamma steget i adressgenerering.
Den naiva sätt att generera fåfänga adresser skulle vara att välja en pseudoslump x , beräkna $H(g^x)$, och upprepa om den adressen inte fungerar. En mycket snabbare metod är att försöka $x + 1$ Om den första x misslyckas, och
fortsätta uppräkning i stället för att plocka en ny x varje gång. Det beror på $g^{x+1} = xg^x$, och vi har redan beräknade g^x , så vi behöver bara en multiplikation operation för varje adress i stället för exponentiering, och det är mycket snabbare. Faktum är att det påskyndar genereringen av fåfängaadresser med över två
storleksordningar.

4.2 Varm- och kallförvaring

Som vi precis såg är att lagra bitcoins på din dator som att bära runt pengar i din plånbok eller din handväska. Detta kallas "hot storage". Det är bekvämt men också något riskabelt. Å andra sidan, "kallt

---

lagring" är offline. Den är inlåst någonstans. Den är inte ansluten till internet och den är arkiverad. Så det är säkrare och säkrare, men naturligtvis inte lika bekvämt. Det här liknar hur du bär några pengar runt på din person, men placera ditt livs besparingar någonstans säkrare.
För att ha separat varm- och kallförvaring måste du självklart ha separata hemliga nycklar för varje — annars skulle mynten i kylförvaring vara sårbara om varmförvaringen äventyras. Du kommer vill flytta mynt fram och tillbaka mellan den varma sidan och den kalla sidan, så varje sida måste göra det känna till den andres adresser eller offentliga nycklar.
Kylförrådet är inte online, så det går inte att ansluta till varmförrådet och kylförrådet varandra över alla nätverk. Men den goda nyheten är att kylförvaring inte behöver vara online för att ta emot mynt — eftersom varmlagret känner till kyllageradresserna kan det skicka mynt till kallt lagring när som helst. När som helst om summan pengar i din heta plånbok blir obehagligt stor kan du överföra en del av den till kylförvaring, utan att riskera din kylförvaring ansluta till nätverket. Nästa gång kylförrådet ansluter kommer det att kunna ta emot från blockkedjeinformation om dessa överföringar till det och sedan kyllageret kommer att kunna göra vad det

vill ha med de mynten.
Men det finns ett litet problem när det gäller hantering av kyllageradresser. Å ena sidan, som vi såg tidigare, av privatliv och andra skäl vill vi kunna ta emot varje mynt på ett separat adress med olika hemliga nycklar. Så när vi överför ett mynt från den varma sidan till den kalla sidan vi skulle vilja använda en fräsch kall adress för det ändamålet. Men eftersom den kalla sidan inte är online har vi att ha något sätt för den heta sidan att ta reda på om dessa adresser.
Den trubbiga lösningen är att den kalla sidan genererar ett stort antal adresser på en gång och skickar dem över för den varma sidan för att använda dem en efter en. Nackdelen är att vi måste med jämna mellanrum återanslut den kalla sidan för att överföra fler adresser.
Hierarkiska plånböcker . En mer effektiv lösning är att använda en hierarkisk plånbok. Det tillåter den kalla sidan att
använd ett väsentligen obegränsat antal adresser och den heta sidan för att veta om dessa adresser, men med endast en kort engångskommunikation mellan de två sidorna. Men det krävs lite av kryptografiska knep.
För att granska, tidigare när vi pratade om nyckelgenerering och digitala signaturer i kapitel 1, vi tittade på en funktion som heter generKeys som genererar en publik nyckel (som fungerar som en adress) och en hemlig nyckel. I en hierarkisk plånbok fungerar nyckelgenerering annorlunda. Istället för att generera en en enda adress genererar vi vad vi kallar adressgenereringsinformation, och snarare än en privat nyckel generera vad vi kallar information om generering av privat nyckel. Med tanke på adressgenereringsinformationen kan vi
generera en sekvens av adresser: vi tillämpar en adressgenereringsfunktion som tar som indata adressgenerering info och vilket heltal i och genererar i : te adressen i sekvensen. Likaså vi kan generera en sekvens av privata nycklar med hjälp av den privata nyckelgenereringsinformationen.

Den kryptografiska magi som gör detta användbart är att för varje i den i : te adress och *jag* "th hemlig nyckel ”Matcha upp” - det vill säga i "e hemliga nyckelkontroller, och kan användas för att spendera, Bitcoins från i : te adress precis som om paret skapades på gammaldags sätt. Så det är som om vi har en sekvens av vanliga nyckelpar.
Den andra viktiga kryptografiska egenskapen här är säkerhet: informationen om adressgenerering läcker inte all information om de privata nycklarna. Det betyder att det är säkert att ge information om adressgenereringen till vem som helst, och så att vem som helst kan aktiveras att generera den 'i':e nyckeln.
Nu kan inte alla digitala signaturscheman som finns modifieras för att stödja hierarkisk nyckel generation. Vissa kan och vissa kan inte, men den goda nyheten är att det digitala signatursystemet som används av Bitcoin, ECDSA, stöder hierarkisk nyckelgenerering, vilket tillåter detta trick. Det vill säga den kalla sidan genererar godtyckligt många nycklar och den heta sidan genererar motsvarande adresser.
Figur 4.2: Schema för en hierarkisk plånbok . Den kalla sidan skapar och sparar privat nyckelgenerering info och adressgenereringsinfo. Den gör en engångsöverföring av den senare till den heta sidan. Den heta sidan genererar en ny adress sekventiellt varje gång den vill skicka mynt till den kalla sidan. När kalla sidan återansluter, den genererar adresser sekventiellt och kontrollerar blockkedjan för överföringar till dessa adresser tills den når en adress som inte har fått några mynt. Det kan också generera privata nycklar sekventiellt om den vill skicka tillbaka några mynt till den heta sidan eller spendera dem på annat sätt.
Så här fungerar det. Minns att normalt en ECDSA privat nyckel är ett slumptal x och motsvarande offentliga nyckeln är g^x . För hierarkiskt nyckelgenerering, behöver vi två andra slumpmässiga värden k
och y .

Information om generering av privata nyckel:

k , x , y

Jag th privata nyckel:

x_i

$= y + H(k \parallel i)$

Adressgenereringsinformation:

k , g_y

Jag th publika nyckel:

$g_{x\_i} = g^{H(k \parallel i)} \cdot g_y$

i :te adress:

$H(g_{x\_i})$

Detta har alla egenskaper som vi vill ha: varje sida kan generera sin sekvens av nycklar, och motsvarande tangenter matcha upp eftersom (eftersom den publika nyckeln som motsvarar en privat nyckel x säga g_x).

Den har en annan egenskap som vi inte har pratat om: när du ger ut de offentliga nycklarna, de nycklarna kommer inte att kunna kopplas till varandra, det vill säga det går inte att sluta sig till att de kommer från samma plånbok. Halmmanslösningen att låta den kalla sidan generera ett stort antal adresser har denna egenskap, men vi var tvungna att se till att bevara den med den nya tekniken med tanke på det nycklarna är faktiskt inte oberoende genererade. Den här egenskapen är viktig för integritet och anonymitet, som kommer att vara ämnet för kapitel 6.

Här har vi två säkerhetsnivåer, där den heta sidan ligger på en lägre nivå. Om den heta sidan är komprometteras, kommer egenskapen för upplösning av länkbarhet som vi just diskuterade att gå förlorad, men de privata nycklarna (och bitcoins) är fortfarande säkra. I allmänhet stöder detta schema godtyckligt många säkerhetsnivåer --- därför "hierarkisk" --- även om vi inte har sett detaljerna. Detta kan vara användbart, till exempel när det finns är flera nivåer av delegering inom ett företag.

Låt oss nu prata om de olika sätten på vilka kall information — om en eller flera nycklar, eller information om nyckelgenerering — kan lagras. Det första sättet är att lagra den i någon form av enhet och lägga den enheten i ett kassaskåp. Det kan vara en bärbar dator, en mobiltelefon eller surfplatta eller ett minne. De Det viktiga är att stänga av enheten och låsa den, så att om någon vill stjäla den att bryta sig in i det låsta förrådet.

Hjärnplånbok . Den andra metoden vi kan använda kallas hjärnplånbok. Detta är ett sätt att kontrollera åtkomsten till bitcoins som inte använder något annat än en hemlig lösenfras. Detta undviker behovet av hårddiskar, papper eller liknande annan långtidslagringsmekanism. Den här egenskapen kan vara särskilt användbar i situationer där du har dålig fysisk säkerhet, kanske när du reser internationellt.

Nyckeltricket bakom en hjärnplånbok är att ha en förutsägbar algoritm för att förvandla en lösenfras till en offentlig och privat nyckel. Du kan till exempel hasha lösenfrasen med en lämplig hashfunktion till härleda den privata nyckeln, och givet den privata nyckeln kan den publika nyckeln härledas på ett standard sätt. Vidare, genom att kombinera detta med den hierarkiska plånbokstekniken vi såg tidigare, kan vi generera en hela sekvensen av adresser och privata nycklar från en lösenfras, vilket möjliggör en komplett plånbok.

Men en motståndare kan också få alla privata nycklar i en hjärnplånbok om de kan gissa det lösenordsfras. Som alltid inom datasäkerhet måste vi utgå från att motståndaren känner till proceduren du använde för att generera nycklar, och endast din lösenfras ger säkerhet. Så motståndaren kan försöka olika lösenfraser och generera adresser med hjälp av dem; om han hittar några outnyttjade transaktioner på blockchain på någon av dessa adresser kan han omedelbart överföra dem till sig själv. Motståndaren

kanske aldrig vet (eller bryr sig) vem mynten tillhörde och attacken kräver inte inbrott i någon maskiner. Att gissa lösenfraser för hjärnplånbok är inte riktade mot specifika användare, och vidare, lämnar inget spår.

Till skillnad från uppgiften att gissa din e-lösenord som kan vara *hastighetsbegränsad* av e-post server (kallas *på nätet gissa*), med hjärnan plånböcker angriparen kan hämta listan över adresser med oinlösta mynt och prova så många potentiella lösenordsfraser som de har beräkningskapacitet för kolla upp. Observera att angriparen inte behöver veta vilka adresser som motsvarar hjärnplånböcker. Detta kallas *offline gissa* eller *lösenord sprickbildning* . Det är mycket mer utmanande att komma på lösenordsfraser som är lätta att memorera och som ändå inte är sårbara för att gissa på det här sättet. Ett säkert sätt att generera en lösenfras är att ha en automatisk procedur för att välja en slumpmässig 80-bitars nummer och omvandla det numret till en lösenordsfras på ett sådant sätt att olika nummer resulterar i olika lösenfraser.

Sidebar: genererar minnesvärda lösenfraser. En procedur för generering av lösenordsfraser som ger cirka 80 bitar av entropi är att välja en slumpmässig sekvens av 6 ord bland de 10 000 mest gemensam engelska ord ($6 \times \log_2$ (10 000) är ungefär 80). Många tycker att dessa är lättare att memorera än en slumpmässig sträng av tecken. Här är ett par lösenfraser som skapats på detta sätt.

sliten legering fokusering okej reducerande
jordnländska faketireddototillfällen

I praktiken är det också klokt att använda en medvetet långsam funktion för att härleda den privata nyckeln från lösenordsfras för att säkerställa att det tar så lång tid som möjligt för angriparen att prova alla möjligheter. Detta är känt som *nyckel streching* . För att skapa en medvetet långsam nyckelhärledningsfunktion kan vi ta en fasta kryptografisk hashfunktion som SHA-256 och compute säga 2^{20} iterationer av det, att multiplicera angripare arbetsbelastning med en faktor 2^{20} . Självklart, om vi gör det för långsamt kommer det att börja bli irriterande för användaren eftersom deras enhet måste räkna om den här funktionen när de vill spendera mynt från sin hjärnplånbok.

Om en lösenfras för hjärnplånbok är otillgänglig — säg att den har glömts, inte har skrivits ner, och går inte att gissa — då är mynten förlorade för alltid.

Pappersplånbok . Det tredje alternativet är vad som kallas en pappersplånbok. Vi kan skriva ut nyckelmaterialet på papper och lägg sedan papperet på en säker plats. Självklart är säkerheten för denna metod precis som bra eller dåligt som den fysiska säkerheten för det papper som vi använder. Typiska pappersplånböcker kodar båda den offentliga och privata nyckeln på två sätt: som en 2D-streckkod och i bas 58-notation. Precis som med en hjärna plånbok är det tillräckligt att lagra en liten mängd nyckelmaterial för att återskapa en plånbok. [107]

Figur 4.3: En Bitcoin-pappersplånbok med den publika nyckeln kodad både som en 2D-streckkod och i bas 58 notation.Observera att den privata nyckeln ligger bakom en manipuleringssäker försegling.

Åtgärdssäker enhet. Det fjärde sättet att lagra offlineinformation är att lägga in den i en del typ av manipuleringssäker enhet. Antingen sätter vi nyckeln i enheten eller så genererar enheten nyckeln; Hur som helst, enheten är utformad så att den inte kan mata ut eller avslöja nyckeln. Enheten undertecknar istället uttalanden med nyckeln, och gör det när vi, säg, trycker på en knapp eller ger den någon form av lösenord. En fördel är att om enheten tappas bort eller blir stulen kommer vi att veta det, och det enda sättet nyckel kan bli stulen är om enheten blir stulen. Detta skiljer sig från att förvara din nyckel på en bärbar dator.

I allmänhet kan folk använda en kombination av fyra av dessa metoder för att säkra sina nycklar. För

varm lagring, och speciellt för varm lagring som innehåller stora mängder bitcoins, är folk villiga att arbeta ganska hårt och komma på nya säkerhetssystem för att skydda dem, så pratar vi en lite om ett av dessa mer avancerade scheman i nästa avsnitt.

4.3 Dela och dela nycklar

Hittills har vi tittat på olika sätt att lagra och hantera de hemliga nycklar som styr bitcoins, men vi har alltid lagt en nyckel på ett enda ställe – oavsett om det är inlåst i ett kassaskåp eller i programvara eller på papper. Detta lämnar oss med en enda punkt av misslyckande. Om något går fel med den enda lagringen plats då har vi problem. Vi skulle kunna skapa och lagra säkerhetskopior av nyckelmaterialet, men medan detta minskar risken för nyckeln att gå vilse eller korrupt (tillgänglighet), det *ökar* risken för stöld (säkerhet). Denna avvägning verkar grundläggande. Kan vi ta en bit data och lagra den på ett sådant sätt att tillgängligheten och säkerheten ökar samtidigt? Anmärkningsvärt nog är svaret ja, och det är det en gång återigen ett trick som använder kryptering, kallad *hemlig delning* .

Här är idén: vi vill dela upp vår hemliga nyckel i ett antal N bitar. Vi vill göra det i på ett sådant sätt att om vi får någon K av dessa bitar så kommer vi att kunna rekonstruera originalet hemlighet, men om vi får färre än K bitar kommer vi inte att kunna lära oss något om ursprungliga hemlighet.

Sida 10

Med tanke på detta stränga krav kommer det inte att fungera att helt enkelt "klippa upp" hemligheten i bitar eftersom till och med en enda bit ger lite information om hemligheten. Vi behöver något smartare. Och eftersom vi är Om vi inte skär upp hemligheten, kallar vi de enskilda komponenterna för "andelar" istället för bitar.

Låt oss säga att vi har N=2 och K=2. Det betyder att vi genererar 2 aktier baserat på hemligheten, och vi behöver båda aktierna för att kunna rekonstruera hemligheten. Låt oss kalla vår hemliga S, som bara är en stor (säg 128-bitars) nummer. Vi skulle kunna generera ett 128-bitars slumptal R och få de två delarna att vara R och S $\oplus$R. ($\oplus$representerar bitvis XOR). I huvudsak har vi "krypterat" S med en engångsplatta, och vi lagrar nyckeln (R) och chiffertexten (S $\oplus$R) på separata ställen. Varken nyckeln eller chiffertexten i sig berättar något om hemligheten. Men med tanke på de två aktierna, XOR vi dem helt enkelt tillsammans rekonstruera hemligheten.

Det här tricket fungerar så länge som N och K är samma — vi skulle bara behöva generera N-1 olika slumpmässiga numren för de första N-1 aktierna, och den sista andelen skulle vara den hemliga XOR'd med alla andra N-1 aktier. Men om N är mer än K, fungerar det inte längre, och vi behöver lite algebra.

Figur 4.4: Geometrisk illustration av 2-av-N hemlig delning. S representerar hemligheten, kodad som en (stort) heltal. Den gröna linjen har en lutning vald slumpmässigt. De orange punkterna (specifikt deras Y-koordinater S+R, S+2R, ...) motsvarar andelar. Varje två orange punkter är tillräckliga för att rekonstruera den röda punkten, och därav hemligheten. All aritmetik görs modulo ett stort primtal.

Ta en titt på figur 4.4. Vad vi har gjort här är att först generera punkten (0, S) på Y-axeln, och dra sedan en linje med en slumpmässig lutning genom den punkten. Därefter genererar vi ett gäng poäng på det linje, så många vi vill. Det visar sig att detta är en hemlig delning av S där N är antalet poäng vi genererade och K=2.

Sida 11

Varför fungerar detta? Först, om du får två av de genererade punkterna kan du dra en linje igenom dem och se var den möter Y-axeln. Det skulle ge dig S. Å andra sidan, om du bara får en enda punkt säger den ingenting om S, eftersom linjens lutning är slumpmässig. Varje rad genom

din punkt är lika trolig, och de skulle alla skära Y-axeln vid olika punkter.
Det finns bara en annan subtilitet: för att få matematiken att fungera måste vi göra alla våra aritmetiska modulo ett stort primtal P. Det behöver inte vara hemligt eller så, bara riktigt stort. Och den hemlighet S har att vara mellan 0 och P-1, inklusive. Så när vi säger att vi genererar poäng på linjen, menar vi det att vi genererar ett slumpmässigt värde R, också mellan 0 och P-1, och de punkter vi genererar är
x=1, y=(S+R) mod P
x=2, y=(S+2R) mod P
x=3, y=(S+3R) mod P
och så vidare. Hemligheten motsvarar punkten x=0, y=(S+0*R) mod P, som bara är x=0, y=S.
Det vi har sett är ett sätt att göra hemlig delning med K=2 och valfritt värde på N. Detta är redan ganska bra — om N=4, säg, kan du dela upp din hemliga nyckel i fyra delar och placera dem på fyra olika enheter så att om någon stjäl någon av dessa enheter lär de sig ingenting om din nyckel. Å andra sidan, även om två av dessa enheter förstörs i en brand, kan du rekonstruera nyckeln med de andra två. Så som utlovat har vi ökat både tillgänglighet och säkerhet.
Men vi kan göra bättre: vi kan göra hemlig delning med vilket N och K som helst så länge K inte är mer än N. Till se hur, låt oss gå tillbaka till figuren. Anledningen till att vi använde en linje istället för någon annan form är att en linje är, algebraiskt sett, ett polynom av grad 1. Det betyder att för att rekonstruera en linje behöver vi två poäng och inte mindre än två. Om vi ville ha K=3 skulle vi ha använt en parabel, som är en andragradspolynom, eller ett polynom med grad 2. Exakt tre punkter behövs för att konstruera en kvadratisk funktion. Vi kan använda tabellen nedan för att förstå vad som händer.

Ekvation	**Grad**	**Form**	**Slumpmässiga parametrar**	**Antal poäng (K)** **behövs för att återställa S**
$(S + RX) \bmod P$	1	Linje	R	2
$(S + R_1 X + R_2 X_2) \bmod P$	2	Parabel	R_1, R_2	3
$(S + R_1 X + R_2 X_2 + R_3 X_3) \bmod P$	3	Kubisk	R_1, R_2, R_3	4

Tabell 4.1: Matematiken bakom hemlighetsdelning. Representera en hemlighet via en serie punkter på en slumpmässig
polynomkurva av grad K-1 gör att hemligheten kan rekonstrueras om, och endast om, åtminstone K av

poäng ("andelar") är tillgängliga.
110

Sida 12

Det finns en formel som heter Lagrange-interpolation som låter dig rekonstruera ett gradpolynom
K-1 från alla K-punkter på dess kurva. Det är en algebraisk version (och en generalisering) av det geometriska
intuition att dra en rak linje genom två punkter med en linjal. Som ett resultat av allt detta har vi en
sätt att lagra hemligheter som N delar så att vi är säkra även om en motståndare lär sig upp till K-1 av dem,
och samtidigt kan vi tolerera förlusten av upp till NK av dem.
Inget av detta är specifikt för Bitcoin, förresten. Du kan hemligt dela dina lösenord just nu och ge
delar med dina vänner eller placera dem på olika enheter. Men ingen gör egentligen det här med hemligheter som
lösenord. Bekvämlighet är en anledning; en annan är att det finns andra säkerhetsmekanismer tillgängliga
för viktiga onlinekonton, som tvåfaktorssäkerhet med SMS-verifiering. Men med Bitcoin, om
du lagrar dina nycklar lokalt, du har inte de andra säkerhetsalternativen. Det finns inget sätt att göra
kontrollen av en Bitcoin-adress beroende på mottagandet av ett SMS-meddelande. Situationen är annorlunda
med onlineplånböcker, som vi ska titta på i nästa avsnitt. Men inte alltför annorlunda - det skiftar bara
problem till en annan plats. När allt kommer omkring kommer leverantören av plånbok online att behöva något sätt att undvika en singel
point of failure vid förvaring *sina* nycklar.
Tröskelkryptering . Men det finns fortfarande ett problem med hemlig delning: om vi tar en nyckel och vi delar den
upp på det här sättet och vi sedan vill gå tillbaka och använda nyckeln för att signera något, vi behöver fortfarande ta med
aktierna tillsammans och räkna om den initiala hemligheten för att kunna signera med den nyckeln. De
punkt där vi samlar alla aktier är fortfarande en enda sårbarhet där en motståndare
kanske kan stjäla nyckeln.
Kryptografi kan också lösa detta problem: om andelarna lagras i olika enheter finns det en
sätt att producera Bitcoin-signaturer på ett decentraliserat sätt utan att någonsin rekonstruera det privata
knapp på valfri enhet. Detta kallas en "tröskelsignatur". Det bästa användningsfallet är en plånbok med
tvåfaktorssäkerhet, vilket motsvarar fallet N=2 och K=2. Säg att du har konfigurerat din plånbok till
dela upp nyckelmaterialet mellan ditt skrivbord och din telefon. Då kan du initiera en betalning den
ditt skrivbord, vilket skulle skapa en partiell signatur och skicka den till din telefon. Din telefon skulle göra det
varna dig sedan med betalningsinformationen – mottagare, belopp etc. – och begär din bekräftelse. Om
detaljerna checkar ut, du skulle bekräfta, och din telefon skulle slutföra signaturen med sin andel av
den privata nyckeln och sända transaktionen till blockkedjan. Om det fanns skadlig programvara på din
skrivbord som försökte stjäla dina bitcoins, kan det initiera en transaktion som skickade pengarna till
hackarens adress, men sedan skulle du få en varning på din telefon för en transaktion som du inte godkände,
och du skulle veta att något var på gång. De matematiska detaljerna bakom tröskelsignaturer är
komplexa och vi kommer inte att diskutera dem här.
Multisignaturer . Det finns ett helt annat alternativ för att undvika en enda punkt av misslyckande:
multisignaturer, som vi såg tidigare i kapitel 3. Istället för att ta en enda nyckel och dela upp den,
Bitcoin-skript låter dig direkt bestämma att kontrollen över en adress ska delas mellan olika
nycklar. Dessa nycklar kan sedan lagras på olika platser och signaturerna produceras separat. Av
naturligtvis kommer den avslutade, undertecknade transaktionen att konstrueras på någon enhet, men även om
motståndaren kontrollerar den här enheten, allt han kan göra är att förhindra att den sänds till nätverket.
111

Sida 13

Han kan inte producera giltiga multisignaturer för någon annan transaktion utan inblandning av

andra enheter.
Anta som ett exempel att Andrew, Arvind, Ed, Joseph och Steven, författarna till den här boken, är det medgrundare av ett företag — vi kanske startade det med de rikliga royalties från försäljningen av detta gratis bok — och företaget har många bitcoins. Vi kanske använder multi-sig för att skydda vår stora butik av bitcoins. Var och en av oss fem kommer att generera ett nyckelpar, och vi kommer att skydda vår kylförvaring med hjälp av
3-av-5 multi-sig, vilket innebär att tre av oss måste signera för att skapa en giltig transaktion.
Som ett resultat vet vi att vi är relativt säkra om vi fem förvarar våra nycklar separat och säkert dem annorlunda. En motståndare skulle behöva kompromissa med tre av de fem nycklarna. Om en eller till och med
två av oss är oseriösa, de kan inte stjäla företagets mynt eftersom du behöver minst tre nycklar att göra den där. Samtidigt, om någon av oss tappar bort vår nyckel eller blir överkörd av en buss och vår hjärnplånbok tappas bort,
de andra kan fortfarande få tillbaka mynten och överföra dem till en ny adress och återsäkra nycklarna.
Med andra ord, multi-sig hjälper dig att hantera stora mängder kalllagrade mynt på ett sätt som relativt säker och kräver åtgärder av flera personer innan något drastiskt händer.
Sidofältet . Tröskelsignaturer är en kryptografisk teknik för att ta en enda nyckel, dela upp den i aktier, lagra dem separat och signera transaktioner utan att rekonstruera nyckeln. Multisignaturer är en funktion i Bitcoin-skript som du kan ange att kontrollen av en adress delas mellan flera oberoende nycklar. Även om det finns vissa skillnader mellan dem ökar de båda säkerhet genom att undvika enskilda felpunkter.
I vår presentation ovan motiverade vi tröskelsignaturer genom att förklara hur det kan bidra till att uppnå tvåfaktors (eller multifaktor) säkerhet och multisignaturer genom att förklara hur det kan hjälpa en uppsättning individer delar kontrollen över gemensamma fonder. Men båda teknikerna är tillämpliga på båda situationerna.

4.4 Onlineplånböcker och börser

Hittills har vi pratat om sätt på vilka du kan lagra och hantera dina bitcoins själv. Nu ska vi prata om hur du kan använda andras tjänster för att hjälpa dig göra det. Det första du kan göra är att använda en onlineplånbok.
Online plånböcker . En onlineplånbok är ungefär som en lokal plånbok som du kanske hanterar själv, förutom informationen lagras i molnet, och du kommer åt den via ett webbgränssnitt på din dator eller med en app på din smartphone. Vissa onlineplånbokstjänster som är populära i början av 2015 är Coinbase och blockchain.info.
Det som är avgörande ur säkerhetssynpunkt är att sajten levererar koden som körs på din webbläsaren eller appen, och den lagrar även dina nycklar. Den kommer åtminstone att ha möjlighet att komma åt dina nycklar.

Helst kommer sajten att kryptera dessa nycklar under ett lösenord som bara du känner till, men självklart har du att lita på att de gör det. Du måste lita på deras kod för att inte läcka dina nycklar eller ditt lösenord.
En onlineplånbok har vissa avvägningar till att göra saker själv. En stor fördel är att det är bekvämt.
Du behöver inte installera något på din dator för att kunna använda en onlineplånbok i din webbläsare. På din telefon kanske du bara behöver installera en app en gång, och den behöver inte laddas ner blockkedjan. Det kommer att fungera på flera enheter - du kan ha en enda plånbok som du kommer åt på ditt skrivbord och på din telefon och det kommer bara att fungera eftersom den riktiga plånboken bor i molnet.
Å andra sidan finns det säkerhetsbekymmer. Om sajten eller personerna som driver sajten visar sig för att vara skadlig eller på något sätt äventyras, är dina bitcoins i trubbel. Webbplatsen tillhandahåller koden som har sina fula fingrar på dina bitcoins, och saker kan gå fel om det finns en kompromiss eller illvilja hos tjänsteleverantören.

Helst drivs sajten eller tjänsten av säkerhetspersonal som är bättre utbildade, eller kanske fler flitig än du för att upprätthålla säkerheten. Så du kanske hoppas att de gör ett bättre jobb och att din mynt är faktiskt säkrare än om du förvarade dem själv. Men i slutet av dagen måste du lita på dem och du måste lita på att de inte äventyras.

Bitcoin-utbyten . För att förstå Bitcoin-utbyten, låt oss först prata om hur banker eller banker gillar tjänster verkar i den traditionella ekonomin. Du ger banken lite pengar — en insättning — och Banken lovar att ge dig tillbaka pengarna senare. Naturligtvis, avgörande, banken faktiskt inte bara ta dina pengar och lägg dem i en låda i det bakre rummet. Allt banken gör är att lova det om du dyker upp för pengarna ger de tillbaka det. Banken tar vanligtvis pengarna och lägger dem någon annanstans, det vill säga investera det. Banken kommer förmodligen att ha lite pengar i reserv för att vara säker att de kan betala ut efterfrågan på uttag som de kommer att möta en vanlig dag, eller kanske till och med en ovanlig dag. Många banker använder vanligtvis något som kallas ***fractional reserve*** där de håller en viss bråkdel av alla avsatta insättningar på reserv för säkerhets skull.

Nu är Bitcoin-börser företag som åtminstone ur användargränssnittssynpunkt fungerar i en liknande sätt som banker. De accepterar insättningar av bitcoins och kommer, precis som en bank, att lova att ge dem
tillbaka på begäran senare. Du kan också överföra fiat-valuta - traditionell valuta som dollar och euro — till ett utbyte genom att göra en överföring från ditt bankkonto. Börsen lovar att betala tillbaka ena eller båda typerna av valuta på begäran. Exchange låter dig göra olika bankliknande saker. Du kan göra och ta emot Bitcoin-betalningar. Det vill säga, du kan styra utbytet att betala ut några bitcoins till en viss part, eller så kan du be någon annan att sätta in pengar på den specifika börsen på dina vägnar – sätt in på ditt konto. De låter dig också byta bitcoins mot fiatvaluta eller vice versa. Vanligtvis gör de detta genom att hitta någon kund som vill köpa bitcoins med dollar och någon annan kund som vill sälja bitcoins för dollar och matcha dem. Med andra ord, de försöka hitta kunder som är villiga att ta motsatta positioner i en transaktion. Om det finns ett ömsesidigt acceptabelt pris, kommer de att fullfölja den transaktionen.

Anta att mitt konto på någon börs innehåller 5000 dollar och tre bitcoins och jag använder utbytet, jag beställer att köpa 2 bitcoins för 580 dollar styck, och börsen hittar någon som är villig för att ta den andra sidan av den transaktionen och transaktionen sker. Nu har jag fem bitcoins i min konto istället för tre och 3840 dollar istället för 5000.

Det viktiga att notera här är att när den här transaktionen hände involverade mig och en annan kund på samma börs, inträffade faktiskt ingen transaktion på Bitcoin-blockkedjan. De utbyte behöver inte gå till blockkedjan för att överföra bitcoins eller dollar från en konto till en annan. Allt som händer i den här transaktionen är att utbytet nu gör ett annat lova mig än vad de gjorde tidigare. Innan de sa, "vi ger dig 5000 USD och 3 BTC" och nu säger de "vi ger dig 3840 USD och 5 BTC." Det är bara en förändring i deras löfte - nej faktiska rörelser av pengar genom dollarekonomin eller genom blockkedjan. Naturligtvis annan person har fått sina löften till dem ändrade på motsatt sätt.

Det finns för- och nackdelar med att använda utbyten. Ett av de stora fördelarna är att utbyten hjälper till att koppla ihop
Bitcoin-ekonomin och flödena av bitcoins med fiat-valutaekonomin så att det är lätt att överföra värde fram och tillbaka. Om jag har dollar och bitcoins på mitt konto kan jag handla fram och tillbaka mellan dem ganska lätt, och det är verkligen användbart.

Nackdelen är risken. Du har samma typ av risk som du möter med banker, och de riskerna delas in i tre kategorier.

Tre typer av risker . Den första risken är risken för en ***uttagsanstormning*** . En löpning är vad som händer när ett gäng

folk dyker upp på en gång och vill ha tillbaka sina pengar. Eftersom banken upprätthåller endast bråkdelar reserver, kan den kanske inte klara av de samtidiga uttagen. Faran är en slags panik beteende där när ryktet börjar spridas att en bank eller börs kan vara i trubbel och de kanske närmar sig att inte hedra uttag, sedan stampade folk in för att försöka dra sig ur deras pengar före publiken, och du får en sorts lavin.

Den andra risken är att ägarna till bankerna bara kan vara skurkar som driver ett Ponzi-system. Det här är en system där någon får folk att ge dem pengar i utbyte mot vinster i framtiden, men tar sedan faktiskt sina pengar och använder dem för att betala ut vinsten till personer som köpt tidigare. Ett sådant system är dömt att så småningom misslyckas och förlora många människor en massa pengar. Bernie Madoff mest känd drog av detta i senare minne.

Den tredje risken är risken för ett hack, risken att någon – kanske till och med en anställd på börsen – kommer att lyckas penetrera börsens säkerhet. Eftersom utbyten lagra nyckelinformation som kontrollerar stora mängder bitcoins måste de vara riktigt försiktiga med sin mjukvarusäkerhet och deras rutiner — hur de hanterar sin kalla och varma förvaring och allt det där. Om något går fel, dina pengar kan bli stulna från börsen.

Alla dessa saker har hänt. Vi har sett utbyten som misslyckades på grund av motsvarigheten till en bankkörning. Vi har sett utbyten misslyckas på grund av att operatörerna av börsen är skurkar, och det har vi

---

sett byten som misslyckas på grund av inbrott. Faktum är att statistiken inte är uppmuntrande. En studie 2013 fann att 18 av 40 Bitcoin-börser slutade med att stänga på grund av något misslyckande eller oförmåga att betala ut pengarna som börsen hade lovat att betala ut.

Det mest kända exemplet på detta är naturligtvis Mt. Gox. Mt Gox var en gång den största Bitcoin utbyte, och det befann sig så småningom insolvent, oförmöget att betala ut pengarna som det var skyldigt. Mount Gox var ett japanskt företag och det slutade med att det gick i konkurs och fick många att undra vart deras pengar tagit vägen. Just nu är Mt Gox konkurs trasslat in i japanska och amerikanska domstolar, och det kommer att dröja ett tag innan vi vet exakt var pengarna tog vägen. Den rätta sak vi vet är att det finns mycket av det och Mt. Gox har det inte längre. Så detta är en varning berättelse om användningen av utbyten.

Om vi kopplar tillbaka detta till banker ser vi inte en 45 % felfrekvens för banker i de flesta utvecklade länder, och det beror delvis på reglering. Regeringar reglerar traditionella banker på olika sätt.

Bankreglering . Det första som regeringar gör är att de ofta inför en minimireserv krav. I USA är den andel av avisanställda inlåning som banker måste ha i likvida medel formen är vanligtvis 3-10%, så att de kan hantera en ökning av uttag om det händer. Andra, regeringar reglerar ofta vilka typer av investeringar och penninghanteringsmetoder som banker kan använda. Målet är att säkerställa att bankernas tillgångar placeras på platser med relativt låg risk, eftersom det verkligen är insättarnas tillgångar i någon mening.

Nu, i utbyte mot dessa former av reglering, gör regeringar vanligtvis saker för att hjälpa banker eller hjälpa deras insättare. Först kommer regeringar att utfärda insättningsförsäkring. Det vill säga regeringen lovar insättare att om en bank som följer dessa regler går under, kommer regeringen att gottgöra sig på minst en del av dessa insättningar. Regeringar agerar också ibland som en "långivare till sista utväg". Om en bank hamnar i en tuff plats, men det är i grunden solvent, regeringen kan gå in och låna banken pengar för att föra över det tills det kan flytta runt pengar efter behov för att ta sig ut ur skogen.

Så traditionella banker regleras på detta sätt. Bitcoin-utbyten är det inte. Frågan om eller hur Bitcoin-utbyten eller annan Bitcoin-affär ska regleras är ett ämne som vi återkommer till i kapitel 7.

Reservbevis . En Bitcoin-börs eller någon annan som har bitcoins kan använda ett kryptografiskt trick

kallas ett reservbevis för att ge kunderna tröst med pengarna som de satt in. De
Målet är att börsen eller företaget som innehar bitcoins ska bevisa att det har en delreserv — det
de behåller kontrollen över kanske 25 % eller kanske till och med 100 % av de insättningar som folk har gjort.
Vi kan dela upp proof-of-reserve-problemet i två delar. Den första är att bevisa hur mycket reserv
du håller — det är den relativt enkla delen. Företaget publicerar helt enkelt en giltig betalning till sig själv
transaktion av det yrkade förbehållsbeloppet. Det vill säga, om de påstår sig ha 100 000 bitcoins skapar de
en transaktion där de betalar 100 000 bitcoins till sig själva och visar att den transaktionen är giltig.
Sedan signerar de en utmaningssträng — en slumpmässig sträng av bitar som genereras av någon opartisk part — med

---

samma privata nyckel som användes för att signera betalningen till sig själv. Detta bevisar att någon
som visste att privat nyckel deltog i reservbeviset.
Vi bör notera två varningar. Det är strängt taget inte ett bevis på att den part som påstår sig äga
reservatet äger det, men bara att den som äger dessa 100 000 bitcoins är villig att samarbeta i
denna process. Ändå ser detta ut som ett bevis på att någon kontrollerar eller känner någon som
kontrollerar den givna summan pengar. Observera också att du alltid kan göra underkrav på: organisationen
kan ha 150 000 bitcoins men väljer att göra en betalning till sig själv på endast 100 000. Så detta bevis på
reserv bevisar inte att detta är allt du har, men det bevisar att du har åtminstone så mycket.
Bevis på skulder. Den andra biten är att bevisa hur många depositioner du har, vilket är
svår del. Om du kan bevisa dina reserver och dina depositioner kan vem som helst helt enkelt dela
dessa två siffror och det är vad din bråkdelsreserv är. Vi kommer att presentera ett schema som tillåter dig
till *överkrav* , men inte under-anspråk på dina avistainlåning. Så om du kan bevisa att dina reserver är på
minst ett visst belopp och dina skulder är högst ett visst belopp, sammantaget har du
visat sig vara en nedre gräns för din bråkreserv.
Om du inte brydde dig alls om dina användares integritet, kunde du helt enkelt publicera dina register —
närmare bestämt användarnamnet och beloppet för varje kund med en insättning. Nu kan vem som helst
beräkna dina totala skulder, och om du utelämnade någon kund eller ljugit om värdet på deras insättning,
du riskerar att den kunden exponerar dig. Du kan hitta på falska användare, men du kan bara
öka värdet av dina påstådda totala skulder på detta sätt. Så länge det inte finns kunder
klagomål låter detta dig bevisa en lägre gräns för dina insättningar. Tricket är förstås att göra allt detta
samtidigt som du respekterar dina användares integritet.
För att göra detta använder vi Merkle-träd, som vi såg i kapitel 1. Kom ihåg att ett merkle-träd är ett binärt träd
som är byggd med hash-pekare så att var och en av pekarna inte bara säger var vi kan få tag i en bit av
information, men också vad den kryptografiska hashen för informationen är. Utbytet utför
bevis genom att konstruera ett Merkle-träd där varje blad motsvarar en användare, och publicera dess rot
hash. I likhet med det naiva protokollet ovan är det varje användares ansvar att se till att de är det
ingår i trädet. Dessutom finns det ett sätt för användare att kollektivt kontrollera den anspråkade summan av
inlåning. Låt oss gräva i detalj nu.

Figur 4.5: Bevis på skulder. Börsen publicerar roten till ett Merkle-träd som innehåller alla användare
vid löven, inklusive pantbelopp. Alla användare kan begära ett bevis på inkludering i trädet, och
verifiera att insättningsbeloppen förökas korrekt till trädets rot.
Nu ska vi lägga till ett annat fält eller attribut till var och en av dessa hashpekare. Detta attribut
är ett tal som representerar det totala monetära värdet i bitcoins av alla insättningar som finns i underträdet
under hash-pekaren i trädet. För att detta ska vara sant, värdet som motsvarar varje hash

pekaren ska vara summan av värdena för de två hashpekarna under den.
Exchange konstruerar detta träd, kryptografiskt signerar rotpekaren tillsammans med roten attributvärde och publicerar det. Rotvärdet är naturligtvis de totala skulderna, antalet vi är intresserad av. Utbytet gör påståendet att alla användare är representerade i trädets löv, deras insättningsvärden representeras korrekt, och att värdena sprids korrekt uppåt träd så att rotvärdet är summan av alla användares insättningsbelopp.
Nu kan varje kund gå till organisationen och be om ett bevis på korrekt inkludering. Utbytet måste sedan visa kunden delträdet från användarens blad upp till roten, som visas i figuren 4.6. Kunden verifierar sedan att:
1. Rothashpekaren och rotvärdet är samma som vad börsen signerade och publiceras.
2. Hashpekarna är konsekventa hela vägen ner, det vill säga varje hashvärde är verkligen det kryptografisk hash för noden den pekar på.
3. Bladet innehåller korrekt användarkontoinformation (t.ex. användarnamn/användar-ID och insättningsbelopp).
4. Varje värde är summan av värdena av de två värdena under det.

---

5. Inget av värdena är ett negativt tal.

Figur 4.6: Bevis på inkludering i ett Merkle-träd. Bladnoden avslöjas, liksom syskonen till noder på vägen från bladet till roten.

Den goda nyheten är att om varje kund gör detta, kommer varje gren av detta träd att utforskas, och någon kommer att verifiera att för varje hash-pekare är dess associerade värde lika med summan av värdena av dess två barn. Avgörande är att utbytet inte kan presentera olika värden i någon del av trädet olika kunder. Det beror på att det antingen skulle innebära möjligheten att hitta en hashkollision, eller presentera olika grundvärden för olika kunder, vilket vi antar är omöjligt.
Låt oss sammanfatta. Först bevisar utbytet att de har minst X mängd reservvaluta genom att göra a självtransaktion på X-belopp. Då bevisar de att deras kunder som mest har ett belopp Y deponeras. Detta visar att deras reservfraktion är minst X/Y. Vad det betyder är att om en Bitcoin Exchange vill bevisa att de har 25 % reserver på alla insättningar – eller 100 % – de kan göra det i ett sätt som kan verifieras oberoende av vem som helst, och ingen central regulator krävs.
Du kanske märker att de två bevisen som presenteras här (beviset på reserver genom att underteckna en utmaning sträng och bevis på skulder via ett Merkle-träd) avslöjar mycket privat information. Specifikt, de avslöjar alla adresser som används av börsen, det totala värdet av reserverna och skulder, och även viss information om de enskilda kundernas saldon. Verkliga utbyten är tveksam till att publicera detta, och som ett resultat av detta har kryptografiska reservbevis varit sällsynta.
Ett nyligen föreslagit protokoll kallat Provisions möjliggör samma bevis på solvens, men utan avslöjar de totala skulderna eller reserverna eller adresserna som används. Detta protokoll använder mer avancerade

---

krypto och vi kommer inte att täcka det här, men det är ett annat exempel som visar hur kryptografi kan användas för att
säkerställa integritet.
Solvens är en aspekt av reglering som Bitcoin-börser kan bevisa frivilligt, men andra aspekter av reglering är svårare att garantera, som vi kommer att se i kapitel 7.

4.5 Betaltjänster

Hittills har vi pratat om hur du kan lagra och hantera dina bitcoins. Låt oss nu överväga hur en

handlare – oavsett om det är en onlinehandlare eller en lokal återförsäljare – kan acceptera betalningar i bitcoins på ett praktiskt sätt. Handlare stöder i allmänhet Bitcoin-betalningar eftersom deras kunder vill kunna betala med bitcoins. Köpmannen kanske inte vill hålla fast vid bitcoins, utan helt enkelt ta emot dollar eller vad som är den lokala fiatvalutan i slutet av dagen. De vill ha ett enkelt sätt att gör detta utan att oroa dig för mycket om teknik, ändra sin webbplats eller bygga någon typ av teknik för försäljningsställen.

Köpmannen vill också ha låg risk. Det finns olika möjliga risker: användning av ny teknik kan orsaka deras hemsida försvinner, vilket kostar dem pengar. Det finns säkerhetsrisken med att hantera bitcoins — någon kan bryta sig in i deras heta plånbok eller så kommer någon anställd göra av med sina bitcoins. Till sist det finns växelkursrisken: värdet på bitcoins i dollar kan fluktuera från tid till annan. De handlare som kanske vill sälja en pizza för tolv dollar vill veta att de kommer att få tolv dollar eller något i närheten av det, och att värdet på bitcoins som de får in utbytet mot den pizzan kommer inte att sjunka drastiskt innan de kan byta ut dessa bitcoins mot dollar. Betaltjänster finns för att låta både kunden och handlaren få vad de vill ha, överbryggande gapet mellan dessa olika önskningar.

119

Sida 21

Figur 4.7: Exempel på betalningstjänstgränssnitt för att generera en betal-med-Bitcoin-knapp. En köpman kan använda det här gränssnittet för att generera ett HTML-kodavsnitt att bädda in på sin webbplats.

Processen att ta emot Bitcoin-betalningar via en betaltjänst kan se ut så här för handlare:

1. Säljaren går till betaltjänstens webbplats och fyller i ett formulär som beskriver varan, priset, och presentation av betalningswidgeten och så vidare. Figur 4.7 visar ett illustrativt exempel av en blankett från Coinbase.
2. Betaltjänsten genererar HTML-kod som handlaren kan släppa in på sin webbplats.
3. När kunden klickar på betalningsknappen händer olika saker i bakgrunden och så småningom får handlaren en bekräftelse som säger, "en betalning gjordes med kund-ID [kund-id] för artikel [varu-id] i belopp [värde]."

Även om den här manuella processen är vettig för en liten webbplats som säljer en eller två artiklar, eller en webbplats som vill ta emot donationer, kopiera och klistra in HTML-kod för tusentals föremål är naturligtvis omöjligt. Betalning alltså tjänster tillhandahåller också programmatiska gränssnitt för att lägga till en betalningsknapp till dynamiskt genererad webbsidor.

120

Sida 22

Figur 4.8: Betalningsprocess som involverar en användare, handlare och betaltjänst.

Låt oss nu titta på betalningsprocessen mer i detalj för att se vad som händer när kunden gör ett köp med Bitcoin. Stegen nedan illustreras i figur 4.8.

1. Användaren väljer ut en vara att köpa på handlarens webbplats och när det är dags att betala, handlaren kommer att leverera en webbsida som innehåller knappen Betala med Bitcoin, vilket är HTML-kodavsnittet som tillhandahålls av betaltjänsten. Sidan kommer också att innehålla ett transaktions-ID — som är en identifierare som är meningsfull för handlaren och som låter dem hitta en post i sitt eget bokföringssystem — tillsammans med ett belopp som handlaren vill få betalt.
2. Om användaren vill betala med bitcoins klickar de på den knappen. Det kommer att utlösa en HTTPS förfrågan till betaltjänsten med besked om att knappen klickades på och vidarebefordra identiteten

handlaren, handlarens transaktions-ID och beloppet.
3. Nu vet betaltjänsten att denna kund – vem de än är – vill betala en viss mängd bitcoins, och så kommer betaltjänsten att dyka upp någon form av en låda, eller initiera någon form av interaktion med användaren. Detta ger användaren information om hur att betala, och användaren kommer sedan att initiera en bitcoinöverföring till betaltjänsten genom sin föredragen plånbok.
4. När användaren har skapat betalningen kommer betaltjänsten att omdirigera webbläsaren till handlare, förmedlar meddelandet från betaltjänsten att det ser okej ut än så länge. Detta kan till exempel innebära att betaltjänsten har observerat transaktionen som sänds till peer-to-peer-nätverket, men transaktionen har inte fått tillräckligt många (eller några) bekräftelser än så länge. Detta slutför betalningen för användaren, med handlarens frakt av varor i avvaktan på en slutlig bekräftelse från betaltjänsten.
121

---

5. Betaltjänsten skickar senare direkt en bekräftelse till handlaren som innehåller transaktions-ID och belopp. Genom att göra detta berättar betaltjänsten för handlaren att tjänsten är skyldig handlaren pengar i slutet av dagen. Köpmannen skickar sedan varorna till användaren.
Det sista steget är det där betaltjänsten faktiskt skickar pengar till handlaren, i dollar eller någon fiat-valuta, via en insättning till handlarens bankkonto. Detta händer i slutet av fix avräkningsperioder, kanske en gång om dagen, snarare än en gång för varje köp. Betaltjänsten behåller en liten procentandel som avgift; det är så de gör sina intäkter. Vissa av dessa detaljer kanske variera beroende på betaltjänst, men detta är det allmänna systemet.
För att sammanfatta, i slutet av denna process betalar kunden bitcoins och handlaren får dollar, minus en liten andel, och alla är nöjda. Kom ihåg att handlaren vill sälja föremål för en viss antal dollar eller vad som är den lokala fiat-valutan. Betaltjänsten sköter allt annat — ta emot bitcoins från kunder och göra insättningar i slutet av dagen.
Avgörande är att betaltjänsten absorberar all risk. Den absorberar säkerhetsrisken, så den måste ha bra säkerhetsrutiner för att hantera sina bitcoins. Det absorberar växelkursrisken eftersom det är det ta emot bitcoins och betala ut dollar. Om priset på dollar mot bitcoins fluktuerar vilt, kan betaltjänst kan förlora pengar. Men sedan om det svänger vilt åt andra hållet tjänsten kan tjäna pengar, men det är en risk. Att ta till sig det är en del av betaltjänstens verksamhet.
Observera att betaltjänsten troligen verkar i stor skala, så den tar emot ett stort antal bitcoins och betalar ut ett stort antal dollar. det kommer att ha ett konstant behov av att byta bitcoins den tar emot för mer dollar så att den kan hålla cykeln igång. Därför måste en betaltjänst vara en aktiv deltagare på valutamarknaderna som länkar samman fiat-valutor och Bitcoin ekonomi. Så tjänsten behöver inte bara oroa sig för vad växelkursen är, utan också hur växla valuta i stora volymer.
Som sagt, om det kan lösa dessa problem gör avgiften som tjänsten får för varje transaktion det en potentiellt lukrativ verksamhet eftersom den löser bristen på överensstämmelse mellan kundernas vilja att betala bitcoins och köpmäns önskan att bara få dollar och koncentrera sig på att sälja varor.

4.6 Transaktionsavgifter

Ämnet transaktionsavgifter har kommit upp i tidigare kapitel och det kommer upp igen senare kapitel. Här kommer vi att diskutera de praktiska detaljerna om hur transaktionsavgifter sätts i Bitcoin idag.
Närhelst en transaktion läggs in i Bitcoin-blockkedjan kan den transaktionen innehålla en transaktionsavgift. Kom ihåg från ett tidigare kapitel att en transaktionsavgift bara definieras som skillnaden mellan det totala värdet av mynt som går in i en transaktion minus det totala värdet av mynt som kommer ut. Ingångarna måste alltid vara minst lika stora som utgångarna eftersom en vanlig

transaktion kan inte skapa mynt, men om ingångarna är större än utgångarna är skillnaden det anses vara en transaktionsavgift, och den avgiften går till gruvarbetaren som gör blocket som inkluderar detta transaktion.

Ekonomin med transaktionsavgifter är intressant och komplex, men vi begränsar oss till hur transaktionsavgifter är faktiskt satta i Bitcoin eftersom det fungerar från början av 2015. Dessa detaljer ändras då och då, men vi ger dig en ögonblicksbild av det aktuella tillståndet.

Varför existerar transaktionsavgifter överhuvudtaget? Anledningen är att det finns en viss kostnad som någon måste ta på sig för att förmedla din transaktion. Bitcoin-noderna behöver förmedla din transaktion och i slutändan en miner behöver bygga din transaktion till ett block, och det kostar dem lite att göra det. För till exempel, om en gruvarbetares block är något större eftersom det innehåller din transaktion, kommer det att ta något längre för att spridas till resten av nätverket och det finns en något större chans att blockeringen gör det bli föräldralös om ett annat block hittades nästan samtidigt av en annan gruvarbetare.

Så det finns en kostnad - både för peer-to-peer-nätverket och för gruvarbetarna - av att införliva ditt transaktion. Tanken med en transaktionsavgift är att kompensera gruvarbetare för de kostnader de ådrar sig behandla din transaktion. Noder får inte monetär kompensation i det nuvarande systemet, även om det naturligtvis är mycket billigare att köra en nod än att vara gruvarbetare. I allmänhet är du fri att ställ in transaktionsavgiften till vad du vill att den ska vara. Du kan inte betala någon avgift, eller om du vill kan du ställa in avgiften ganska hög. I allmänhet, om du betalar en högre transaktionsavgift är det naturligt att din transaktionen kommer att vidarebefordras och registreras snabbare och mer tillförlitligt.

Aktuella standardtransaktionsavgifter. De nuvarande transaktionsavgifterna som de flesta gruvarbetare förväntar sig är följande:

För det första tas ingen avgift ut om en transaktion uppfyller alla dessa tre villkor:

1. transaktionen är mindre än 1 000 byte stor,
2. alla utgångar är 0,01 BTC eller större
3. Prioriteten är tillräckligt stor

Prioritet definieras som: (summan av inmatad ålder * ingångsvärde) / (transaktionsstorlek). Med andra ord, se överhuvudtaget av indata till transaktionen, och för var och en beräkna produkten av den ingångens ålder och dess värde i bitcoins och summera alla dessa produkter. Observera att ju längre en transaktionsutgång sitter outnyttjad, desto mer åldras den och desto mer kommer den att bidra till prioritet när den slutligen är förbrukad.

Om du uppfyller dessa tre krav kommer din transaktion att vidarebefordras och den kommer att registreras blockkedjan utan avgift. Annars tas en avgift ut och den är cirka 0,0001 BTC per 1000 byte, och från och med 2015 är det en bråkdel av ett amerikanskt öre per 1000 byte. Den ungefärliga storleken på en transaktion är 148 byte för varje ingång plus, 34 byte för varje utgång och tio byte för andra information. Så en transaktion med två ingångar och två utgångar skulle vara cirka 400 byte.

Det nuvarande status quo är att de flesta gruvarbetare tillämpar ovanstående avgiftsstruktur, vilket innebär att de kommer antingen inte att betjäna eller kommer att betjäna de senaste transaktionerna som inte ger den nödvändiga transaktionen avgifter. Men det finns andra gruvarbetare som inte tillämpar dessa regler, och som kommer att spela in och operera på en transaktion även om den betalar en mindre avgift eller ingen avgift alls.

Om du gör en transaktion som inte uppfyller avgiftskraven kommer den förmodligen att hitta vägen till blockchain ändå, men sättet att få din transaktion registrerad snabbare och mer tillförlitligt är att betala standardavgiften, och det är därför de flesta plånboksmjukvara och de flesta betaltjänster inkluderar standardavgiftsstrukturen i de betalningar som pågår, så att du kommer att se en liten bit av pengar som tas ut för transaktionsavgifter när du ägnar dig åt vardagliga Bitcoin-affärer.

4.7 Valutaväxlingsmarknader

Med valutaväxling menar vi handel med bitcoins mot fiatvalutor som dollar och euro. Det har vi pratade tidigare om tjänster som låter dig göra detta, men nu vill vi se på det här som en marknad — dess storlek, omfattning, hur den fungerar och lite om ekonomin på denna marknad.

Det första att förstå är att det fungerar på många sätt som marknaden mellan två fiat valutor som dollar och euro. Priset kommer att fluktuera fram och tillbaka beroende på hur mycket folk vill köpa euro kontra hur gärna folk vill köpa dollar en viss dag. I den Bitcoin-världen det finns sajter som bitcoincharts.com som visar växelkursen med olika fiat valutor på ett antal olika börser.

Som du kommer att se om du utforskar webbplatsen pågår det mycket handel och priserna rör sig i realtid när affärer görs. Det är en likvid marknad och det finns gott om ställen som du kan gå till för att köpa eller sälja bitcoins. I mars 2015 var volymen på Bitfinex, den största Bitcoin-USD-börsen, ungefär 70 000 bitcoins eller cirka 21 miljoner dollar under en 24-timmarsperiod.

Ett annat alternativ är att träffa människor för att handla bitcoins i verkligheten. Det finns sajter som hjälper dig att göra detta. På localbitcoins.com, till exempel, kan du ange din plats och som du vill köpa bitcoins med kontanter. Du kommer att få ett gäng resultat av personer som vid tidpunkten för din sökning är villiga att sälja bitcoins på den platsen, och i varje fall talar den om vilket pris och hur många bitcoins de erbjuder. Du kan då kontakta någon av dem och träffas på ett kafé eller i en park eller var som helst, ge dem dollar och få bitcoins i utbyte. För små transaktioner kan det räcka att vänta på en eller två bekräftelser på blockkedjan.

Slutligen, på vissa ställen finns det regelbundna träffar där folk går för att handla bitcoins, och så kan du gå till en viss park eller gathörn eller café på en schemalagd dag och tid så kommer det att finnas ett gäng människor som vill köpa eller sälja bitcoins och du kan göra affärer med dem. En anledning till att någon kanske föredrar att skaffa bitcoins personligen framför att göra det online är att det är anonymt, i den utsträckning som en transaktion på allmän plats kan anses vara anonym. Å andra sidan, öppna ett konto med ett utbyte kräver generellt att man tillhandahåller statligt utfärdat ID på grund av bankreglering. Väl diskutera detta närmare i kapitel 7.

Tillgång och efterfrågan. Som vilken marknad som helst, matchar Bitcoin-växlingsmarknaden köpare som vill göra en sak med säljare som är villiga att göra det motsatta. Det är en relativt stor marknad — miljoner

---

av amerikanska dollar per dag passerar genom den. Det är inte i skalan av New York Stock Exchange eller dollar–euro-marknaden, som är mycket större, men den är tillräckligt stor för att det finns en uppfattning om konsensus pris. En person som vill komma in på denna marknad kan köpa eller sälja åtminstone en blygsam summa och vilja alltid kunna hitta en motpart.

Priset på denna marknad, detta konsensuspris, som priset på allt på en likvid marknad kommer att fastställas efter utbud och efterfrågan. Med det menar vi utbudet av bitcoins som potentiellt kan säljas och

efterfrågan på bitcoins av människor som har dollar. Priset genom denna marknadsmekanism kommer att fastställas
till den nivå som matchar utbud och efterfrågan. Låt oss gräva in det här lite mer detaljerat.
Vad är utbudet av bitcoins? Detta är antalet bitcoins som du kan tänkas köpa i en av
dessa marknader, och det är lika med utbudet av bitcoins som är i omlopp för närvarande. Det finns en fast
antal bitcoins i omlopp. I oktober 2015 handlar det om 13,9 miljoner, och reglerna för Bitcoin som
de står för närvarande säger att detta antal sakta kommer att stiga och så småningom nå en gräns på 21 miljoner.
Du kan också inkludera insättningar på bitcoins på begäran. Det vill säga om någon har lagt pengar på sina
konto i en Bitcoin-börs, och börsen har inte en full reserv för att möta varenda en
insättning, då kommer det att finnas insättningar på den börsen som är större än antalet mynt
att utbytet håller.
Beroende på vilken fråga du ställer om marknaden kan det vara korrekt eller inte
inkludera efterfrågan i utbudet. I grund och botten bör du inkludera anfordringsinsättningar på en marknad
analys när insatta pengar kan säljas på den marknaden. Till exempel om du har handlat
dollar för en anfordran av bitcoins, och utbytet gör det möjligt att på begäran insätta bitcoins
lösas in mot dollar, då räknas de.
Det är också värt att notera att när ekonomer konventionellt talar om utbudet av fiatvaluta
de inkluderar vanligtvis inte bara den valuta som är i omlopp i penningmängden – det vill säga papper
och metallpengar — men också det totala beloppet av anfordringsinsättningar, och det är av den logiska anledningen
att folk faktiskt kan spendera sina insatta pengar för att köpa grejer. Så även om det är frestande
att säga att utbudet av bitcoins är fixerat till 13,9 miljoner för närvarande eller 21 miljoner så småningom, för vissa
syften som vi måste inkludera anfordringsinsättningar där dessa anfordringsinsättningar fungerar som pengar,
och så utbudet kanske inte är fixat som vissa Bitcoin-förespråkare kan hävda. Vi måste titta på
omständigheterna på den specifika marknaden vi pratar om för att förstå vad
rätt penningmängd är. Men låt oss anta att vi har kommit överens om vilket utbud vi använder baserat på vad
marknaden vi analyserar.
Låt oss nu titta på efterfrågan. Det finns egentligen två huvudkällor för efterfrågan på bitcoins. Det finns en
efterfrågan på bitcoins som ett sätt att förmedla fiat-valutatransaktioner och det finns en efterfrågan på bitcoins
som en investering.
Låt oss först titta på förmedling av fiat-valutatransaktioner. Föreställ dig att Alice vill köpa något av
Bob och vill betala lite pengar till Bob, och Alice och Bob vill överföra låt oss säga en viss

---

mängd dollar, men de tycker att det är bekvämt att använda Bitcoin för att göra denna överföring. Låt oss anta det här
varken Alice eller Bob är intresserade av att behålla bitcoins på lång sikt. Vi återkommer till den möjligheten om en
ögonblick. Så Alice skulle köpa bitcoins för dollar och överföra dem, och när de väl fick tillräckligt
bekräftelser till Bobs belåtenhet kommer han att sälja dessa bitcoins för dollar. Det viktigaste här från
synvinkel efterfrågan på bitcoins är att bitcoins som förmedlar denna transaktion måste tas
ur omlopp under den tid som transaktionen pågår. Detta skapar en efterfrågan på bitcoins.
Den andra källan till efterfrågan är att Bitcoin ibland efterfrågas som en investering. Det är om
någon vill köpa bitcoins och hålla dem i hopp om att priset på bitcoins kommer att gå upp i
framtiden och att de kommer att kunna sälja dem. När människor köper och håller, är dessa bitcoins slut
omlopp. När priset på Bitcoin är lågt kan du förvänta dig att många människor vill köpa bitcoins
som en investering, men om priset går upp mycket högt så är efterfrågan på bitcoins som en investering
blir inte lika hög.
En enkel modell för marknadsbeteende. Nu kan vi göra några enkla ekonomiska modeller för att förstå

hur dessa marknader kommer att bete sig. Vi kommer inte att göra en fullständig modell här även om det är en intressant övning.
Låt oss titta specifikt på efterfrågan på transaktionsförmedling och vilken effekt det kan ha på priset på bitcoins.
Vi börjar med att anta några parametrar. T är det totala transaktionsvärdet som förmedlas via Bitcoin av alla som deltar på marknaden. Detta värde mäts i dollar per sekund. Det är för att vi anta för enkelhetens skull att de personer som vill förmedla dessa transaktioner har i åtanke en viss dollarvärdet på transaktionerna, eller någon annan fiatvaluta som vi översätter till dollar. Så det finns en viss mängd dollar per sekund av transaktioner som behöver förmedlas. D är varaktigheten av tid som bitcoins behöver hållas ur cirkulation för att förmedla en transaktion. Det är dags från när betalaren köper bitcoins till när mottagaren kan sälja tillbaka dem till marknaden, och vi mäter det på några sekunder. S är det totala utbudet av bitcoins som är tillgängliga för detta köp, och så det kommer att vara alla bitcoins i hårdvaluta som finns - för närvarande cirka 14 miljoner eller så småningom upp till 21 miljoner — minus de som hålls ut av människor som långsiktiga investeringar. I med andra ord, vi pratar om bitcoins som skvalpar runt och är tillgängliga i syfte att förmedla transaktioner. Slutligen är P priset på Bitcoin, mätt i dollar per bitcoin.
Nu kan vi göra några beräkningar. Först kommer vi att beräkna hur många bitcoins som blir tillgängliga i ordning att betjäna transaktioner varje sekund. Det finns S bitcoins tillgängliga totalt och eftersom de är tagna ur cirkulation under en tid av D sekunder, varje sekund i genomsnitt en S/D-bråkdel av dessa bitcoins kommer att bli nytillgängliga eftersom de kommer att komma ur det ur-cirkulerande tillståndet och bli tillgänglig för att förmedla transaktioner varje sekund. Det är utbudssidan.
På efterfrågesidan - antalet bitcoins per sekund som behövs för att förmedla transaktioner - vi har transaktioner värda T dollar att förmedla och för att förmedla en dollar värda transaktioner vi behöver 1/P bitcoins. Så T/P är antalet bitcoins per sekund som behövs i för att betjäna alla transaktioner som människor vill betjäna.

Sida 28

Om du nu tittar på en viss sekund av tiden, för den sekunden finns det ett utbud av S/D och en efterfrågan av T/P. På denna marknad, liksom de flesta marknader, kommer priset att fluktuera för att få utbudet i linje med efterfrågan. Om utbudet är högre än efterfrågan så finns det bitcoins som inte såldes, så folk sälja bitcoins kommer att vara villiga att sänka sitt utropspris för att sälja dem. Och enligt vår formel T/P för efterfrågan, när priset sjunker ökar efterfrågan, och utbud och efterfrågan kommer nå jämvikt.
Å andra sidan, om utbudet är mindre än efterfrågan betyder det att det finns människor som vill få bitcoins för att förmedla en transaktion men kan inte få dem eftersom det inte finns tillräckligt med bitcoins runt omkring. Dessa personer kommer då att behöva bjuda mer för att få sina bitcoins eftersom det kommer att finnas en
stor konkurrens om ett begränsat utbud av bitcoins. Detta driver upp priset, och hänvisar till vår formel igen, det betyder att efterfrågan kommer att minska tills det är jämvikt. I jämvikt är utbudet måste vara lika med efterfrågan, så vi har

$$\frac{S}{D} = \frac{T}{P}$$

vilket ger oss en formel för priset:

$$P = \frac{S}{TD}$$

Vad säger denna ekvation oss? Vi kan förenkla det lite ytterligare: vi kan anta att D, varaktigheten för vilket du behöver hålla en bitcoin för att förmedla en transaktion, ändras inte. Det totala utbudet S också

förändras inte, eller åtminstone långsamt över tiden. Det betyder att priset är proportionellt mot efterfrågan på medling mätt i dollar. Så om kravet på medling i dollar fördubblas då priset på bitcoins bör fördubblas. Vi skulle faktiskt kunna rita priset mot någon uppskattning av efterfrågan på transaktionsförmedling och se om de matchar eller inte. När ekonomer gör detta, de två tenderar att matcha ganska bra.

Observera att det totala utbudet S endast inkluderar de bitcoins som inte hålls som investeringar. Så om fler människor köper bitcoins som en investering, S kommer att gå ner, och vår formel säger oss att P kommer att göra det
gå upp. Detta är vettigt - om det finns mer efterfrågan på investeringssidan så priset som du måste betala för att förmedla en transaktion kommer att gå upp.

Nu är detta inte en fullständig modell av marknaden. För att ha en fullständig modell måste vi ta hänsyn till investerarnas aktivitet. Det vill säga investerare kommer att kräva bitcoins när de tror att priset kommer att bli högre
i framtiden, och därför måste vi tänka på investerarnas förväntningar. Dessa förväntningar, naturligtvis, har något att göra med den förväntade efterfrågan i framtiden. Vi skulle kunna bygga en modell som är mer komplex och tar hänsyn till det, men det kommer vi inte att göra här.

Summan av kardemumman här är att det finns en marknad mellan bitcoins och dollar, och mellan bitcoins och andra fiat-valutor. Den marknaden har tillräckligt med likviditet som du kan köpa eller sälja i blygsamma kvantiteter
ett tillförlitligt sätt, även om priset går upp och ner. Äntligen är det möjligt att göra ekonomisk modellering och har en uppfattning om hur utbud och efterfrågan interagerar på denna marknad och förutsäga vad marknaden kan göra, så länge du har ett sätt att uppskatta okända saker som hur mycket folk kostar

---

kommer att vilja använda Bitcoin för att förmedla transaktioner i framtiden. Den typen av ekonomisk modellering är
viktigt att göra och mycket informativt, och det finns säkert människor som gör det i detalj idag, men en detaljerad ekonomisk modell av denna marknad ligger utanför denna texts omfattning.

Vidare läsning

Att säkra bitcoins har vissa likheter, såväl som viktiga skillnader, till hur banker säkrar pengar. Kapitel 10 i Ross Andersons lärobok om säkerhet, med titeln "Banking and bookkeeping", är en stor läsa. Hela boken är fritt tillgänglig online.

Anderson, Ross. *Säkerhetssystem* . John Wiley & Sons, 2008.

Studien som analyserar stängningar av Bitcoin-börser som vi refererade till:

Moore, Tyler och Nicolas Christin. *Akta mellanhanden: empirisk analys av Bitcoin-utbyte risk*. Finansiella Kryptering och datasäkerhet 2013.

Adi Shamirs papper om hemlig delning:

Shamir, Adi. *Så att dela en hemlighet* . Communications of the ACM 22.11 (1979).

Papper som beskriver bestämmelser, ett protokoll för att bevara integriteten solvensbevis:

Dagher, Gaby och Benedikt Bunz och Joseph Bonneau och Jeremy Clark och Dan Boneh. Avsättningar: Privacy bevarande bevis på solvens för Bitcoin utbyte . I ACM CCS, 2015.

Det är svårt för användare att välja minnesvärda men svåra att gissa lösenord eftersom det är modernt Lösenordsknäckningstekniker är ganska smarta och effektiva. Detta dokument presenterar en sådan teknik:

Weir, Matt, Sudhir Aggarwal, Breno De Medeiros, och Bill Glodek. Lösenord sprickbildning hjälp probabilistiska kontextfria grammatiker . I Säkerhet och integritet, 2009.

En undersökning av transaktionsavgifter i praktiken fram till 2014:

. Moser, Malte och Böhme, Rainer Trends, Tips, vägtull: en longitudinell studie av Bitcoin Transaktions Avgifter . 2nd Workshop om Bitcoin Research, 2015.

Övningar

1.

Reservbevis. TransparentExchange hävdar att det kontrollerar minst 500 000 BTC och vill att bevisa detta för sina kunder. För att göra detta publicerar den en lista över adresser som har en summa

---

saldo på 500 000 BTC. Det signerar sedan påståendet "TransparentExchange kontroller åtminstone 500 000 BTC" med var och en av motsvarande privata nycklar, och presenterar dessa signaturer som bevis.

Vilka är några sätt på vilka TransparentExchange skulle kunna producera ett sådant bevis även om det faktiskt för närvarande inte styr 500 000 BTC? Hur skulle du ändra beviset till göra det svårare för börsen att fuska?

2.

Bevis på skulder.

TransparentExchange implementerar ett Merkle Tree-baserat protokoll för att bevisa en övre gräns dess totala insättningar. (Kombinerat med ett reservbevis, bevisar detta att utbytet är lösningsmedel.) Varje kund tilldelas en bladnod som innehåller ett ID som är hashen för hennes användarnamn och ett värde som är hennes BTC-saldo. Protokollet anger att TransparentExchange ska sprida ID:n och värden upp i trädet genom följande rekursiva definition - för alla interna nod:

nod.värde = nod.vänster_barn.värde + nod.höger_barn.värde

node.id = Hash (node.left_child.id ‖ node.right_child.id ‖ node.value)

Börsen publicerar rot-ID och värde och lovar att bevisa det för alla kunder hennes nod är inkluderad i trädet (av standard Merkle träd bevis på inkludering). Tanken är att om börsen försöker göra anspråk på en lägre summa än den faktiska summan av insättningar genom att lämna vissa kunder ur trädet eller genom att göra deras nodvärde mindre än deras saldo, kommer det att göra det fastna när någon av dessa kunder kräver ett bevis på inkludering.

2.1.

Varför kan inte börsen inkludera falska kunder med negativa värden för att sänka total?

2.2.

Visa en attack på detta schema som skulle tillåta utbytet att göra anspråk på totalt mindre än den faktiska summan av insättningar.

2.3.

Fixa det här schemat så att det inte är sårbart för attacken du identifierade.

2.4.

Helst bör beviset som utbytet ger en kund inte läcka information om andra kunder. Har detta schema denna egenskap? Om inte, hur kan du fixa det Det?

3.

Transaktionsavgifter.

3.1.

Alice har ett stort antal mynt vardera av litet värde v , som hon skulle vilja kombinera till ett mynt. Hon konstruerar en transaktion för att göra detta, men finner att transaktionsavgiften hon skulle behöva spendera är lika med summan av hennes myntvärden. Baserat på det här information (och standardtransaktionsavgift policy som anges i slide 50), uppskatta v .

3.2.
Kan Alice på något sätt konsolidera sina mynt utan att ådra sig någon transaktionsavgift enligt standardpolicyn?
3.3.
Jämfört med en avgiftsstruktur som inte tar hänsyn till åldern på indata i transaktionsavgift, vilken effekt kan den nuvarande standardavgiftsstrukturen ha på användares och tjänsters beteende?
4.
Plånbok med flera signaturer
4.1.
BitCorp har precis märkt att Mallory har äventyrat en av deras servrar deras privata Bitcoin-nycklar. Lyckligtvis använder de en 2-av-3 multisignaturplånbok, så Mallory har bara lärt sig en av de tre uppsättningarna nycklar. De andra två uppsättningarna nycklar är på

---

olika servrar som Mallory inte kan komma åt. Hur återsäkrar de sin plånbok och effektivt återkalla informationen som Mallory har lärt sig?
4.2.
Om BitCorp använder en 2-av-2 istället för en 2-av-3 plånbok, vilka steg kan de ta i avancera så att de kan återhämta sig även i händelse av att en av deras servrar får brutit sig in i (och Mallory lär sig inte bara utan också tar bort nyckeln material på den servern)?
5.
Växlingskurs
5.1.
Spekulera om varför det i allmänhet är dyrare att köpa bitcoins personligen än köpa från en onlinebörs.
5.2.
Moore och Christin observera att säkerhetsöverträdelser och andra misslyckanden börser har liten inverkan på Bitcoin-växelkursen. Spekulera om varför detta kan vara.
6.
Betalningar. En Bitcoin-betalningstjänst kan ta emot tusentals betalningar från olika användare nästan samtidigt. Hur kan det se om en viss användare Alice som loggat in på betaltjänst webbplats och initierade betalningsprotokollet faktiskt gjort en betalning eller inte?
7.
BitcoinLotto: Anta att nationen Bitcoinia har beslutat att konvertera sitt nationella lotteri till användning Bitcoin. En pålitlig fabrik för skraplottsutskrift finns och kommer inte att föra register över några värden utskrivna. Bitcoinia föreslår en enkel design: en vecka med biljetter skrivs ut med en adress som har jackpotten på varje lott. Detta gör att alla kan verifiera att jackpotten finns. Den vinnande lotten innehåller den korrekta privata nyckeln under skrapmaterialet.
7.1.
Vad kan hända om vinnaren hittar biljetten på måndag och omedelbart gör anspråk jackpotten? Kan du ändra din design för att säkerställa att detta inte blir ett problem?
7.2.
Vissa biljetter går oundvikligen bort eller förstörs. Så du skulle vilja ändra designen så att den rullar vidarebefordra varje outtagna jackpot från vecka n till vinnaren i vecka $n + 1$. Kan du föreslå en design som fungerar, utan att låta lotteriadministratörerna förskingra medel? Kontrollera också att veckan n vinnaren inte helt enkelt vänta tills början av Vecka $n + 1$ att försöka att fördubbla sina vinster.

Kapitel 5: Bitcoin Mining

Det här kapitlet handlar om gruvdrift. Vi har redan sett en hel del om gruvarbetare och hur Bitcoin förlitar sig på dem — de validerar varje transaktion, de bygger och lagrar alla block, och de når en konsensus om vilka block som ska inkluderas i blockkedjan. Vi har också redan sett att gruvarbetare tjänar någon belöning för att göra detta, men vi har fortfarande lämnat många frågor obesvarade. Vilka är gruvarbetarna? Hur kom de in på det här? Hur fungerar de? Hur är affärsmodellen för gruvarbetare? Vad påverkar de miljön? I det här kapitlet kommer vi att svara på alla dessa frågor.

5.1 Bitcoin-gruvarbetarnas uppgift

Vill du komma in i Bitcoin-gruvdrift? Om du gör det kommer vi inte att helt avskräcka dig, men akta dig för att Bitcoin-brytning har många likheter med guldrusher. Historiska guldrusher är fulla av berättelser om unga människor som rusar iväg för att hitta förmögenhet och oundvikligen många av dem förlorar allt de ha. Ett fåtal slår det rikt, men även de som gör det utstår i allmänhet mycket svårigheter på vägen. Vi kommer att se i det här avsnittet varför Bitcoin-gruvdrift delar många av samma utmaningar och risker som traditionella guldrusher och andra bli rik-snabb-planer.

Men låt oss först titta på de tekniska detaljerna. För att vara en Bitcoin-gruvarbetare måste du gå med i Bitcoin-nätverket och anslut till andra noder. När du är ansluten finns det sex uppgifter att utföra:

1. *Lyssna efter transaktioner.* Först du lyssnar för transaktioner på nätet och validera dem genom kontrollera att signaturerna är korrekta och att utdata som spenderas inte har förbrukats innan.
2. *Behåll blockera kedja och lyssna efter nya block.* Du måste hålla blocket kedjan. Du startar genom att be andra noder att ge dig alla historiska block som redan är en del av blocket kedjan innan du gick med i nätverket. Du lyssnar sedan efter nya block som sänds till nätverket. Du måste validera varje block som du får — genom att validera varje block transaktion i blocket och kontrollera att blocket innehåller en giltig nonce. Vi återkommer till detaljer om icke-kontroll längre fram i detta avsnitt.
3. *Montera en kandidat block.* När du har en up-to-date kopia av blocket kedjan, kan du börja bygga dina egna block. För att göra detta grupperar du transaktioner som du hört talas om i en nytt block som utökar det senaste blocket du känner till. Du måste se till att varje transaktionen som ingår i ditt block är giltig.
4. *Hitta en nonce som gör din blocket giltigt.* Detta steg kräver mest arbete och det är där alla den verkliga svårigheten händer för gruvarbetare. Vi kommer att se detta i detalj inom kort.
5. *Hoppas din blocket accepteras.* Även om du hittar ett block, det finns ingen garanti för att din blocket bli en del av konsensuskedjan. Det är lite tur här; du får hoppas att andra gruvarbetare accepterar ditt block och börjar bryta ovanpå det, istället för någon konkurrents block.
6. *Resultat.* Om alla andra gruvarbetare accepterar ditt block, då du vinst! När detta skrivs in i början av 2015 är blockbelöningen 25 bitcoins som för närvarande är värt över $6 000. Dessutom, om

någon av transaktionerna i blocket innehöll transaktionsavgifter, gruvarbetaren samlar in dessa också. Hittills har transaktionsavgifter varit en blygsam källa till extra inkomst, endast cirka 1 % av blockera belöningar.

Vi kan klassificera stegen som en gruvarbetare måste ta i två kategorier. Vissa uppgifter — validering transaktioner och blockeringar — hjälper Bitcoin-nätverket och är grundläggande för dess existens. Dessa uppgifter
är anledningen till att Bitcoin-protokollet kräver gruvarbetare i första hand. Andra uppgifter — loppet mot hitta block och vinst —- är inte nödvändiga för själva Bitcoin-nätverket men är avsedda att stimulera gruvarbetare att utföra de väsentliga stegen. Naturligtvis är båda dessa nödvändiga för att Bitcoin ska fungera som en valuta, eftersom gruvarbetare behöver ett incitament för att utföra de kritiska stegen.

Att hitta ett giltigt block. Låt oss återvända till frågan om att hitta en nonce som gör din blocket giltigt. I Kapitel 3 såg vi att det finns två huvudsakliga hashbaserade strukturer. Det finns blockkedjan där var och en blockrubrik pekar på föregående blockrubrik i kedjan, och sedan inom varje block finns ett Merkle-trädet för alla transaktioner som ingår i det blocket.

Det första du gör som gruvarbetare är att sammanställa en uppsättning giltiga transaktioner som du har från din pågående transaktion pool i ett Merkle-träd. Naturligtvis kan du välja hur många transaktioner du vill inkludera upp till gränsen för blockets totala storlek. Du skapar sedan ett block med en rubrik som pekar till föregående block. I blockhuvudet finns ett 32-bitars nonce-fält, och du fortsätter att försöka annorlunda nonces letar efter en som gör att blockets hash hamnar under målet - ungefär, till att börja med det erforderliga antalet nollor. En gruvarbetare kan börja med en nonce på 0 och successivt öka den med en på jakt efter en nonce som gör blocket giltigt. Se figur 5.1.

Figur 5.1: Hitta ett giltigt block. I det här exemplet försöker gruvarbetaren en nonce av alla nollor. Det producerar inte
en giltig hash-utgång, så gruvarbetaren skulle sedan fortsätta att prova en annan nonce.

I de flesta fall försöker du varenda möjliga 32-bitars värde för nonce och ingen av dem kommer att producera en giltig hash. Vid det här laget kommer du att behöva göra ytterligare ändringar. Lägg märke till i figur 5.1 att det finns en extra nonce i myntbastransaktionen som du också kan ändra. Efter att du har gjort det uttömt alla möjliga nonces för blockhuvudet, kommer du att ändra den extra nonce i myntbasen transaktion — säg genom att öka den med en — och sedan börjar du söka efter nonces i blocket header ännu en gång.

När du ändrar nonce-parametern i myntbastransaktionen, kommer hela Merkle-trädet på transaktioner måste ändras (se figur 5.2). Eftersom förändringen av myntbasen kommer nonce att spridas alla vägen upp i trädet är det mycket dyrare att ändra den extra nonce i myntbastransaktionen operation än att ändra nonce i blockhuvudet. Av denna anledning spenderar gruvarbetare det mesta tid att ändra nonce i blockhuvudet och bara ändra myntbas nonce när de har utmattad alla av 2^{32} möjliga nonces i blockhuvudet utan att hitta ett giltigt block.

Figur 5.2: Att ändra en nonce i myntbastransaktionen sprider sig hela vägen upp i Merkle-trädet.

Den stora, stora majoriteten av nonces som du försöker kommer inte att fungera, men om du stannar kvar tillräckligt länge
du kommer så småningom att hitta den rätta kombinationen av extra nonce i myntbastransaktionen och inget i blockhuvudet som producerar ett block med en hash under målet. När du hittar detta, du vill meddela det så snabbt du kan och hoppas att du kan dra nytta av det.

Löser alla samma pussel? Du kanske undrar: om varje gruvarbetare bara ökar nonces som vi beskrev, löser inte alla gruvarbetare exakt samma pussel? Kommer inte den snabbaste gruvarbetaren
vinner alltid? Svaret är nej! För det första är det osannolikt att gruvarbetare kommer att arbeta med exakt samma sak
block eftersom varje gruvarbetare sannolikt kommer att inkludera en något annorlunda uppsättning transaktioner och i olika
beställa. Men ännu viktigare, även om två olika gruvarbetare arbetade på ett block med identiska transaktioner, skulle blocken fortfarande skilja sig åt. Kom ihåg att i myntbastransaktionen anger gruvarbetare sin egen adress som ägare till de nypräglade mynten. Denna adress i sig kommer att orsaka ändringar som fortplantar sig upp till roten av Merkle-trädet, vilket säkerställer att inga två gruvarbetare arbetar på exakt samma pussel om de inte delar en offentlig nyckel. Detta skulle bara hända om de två gruvarbetarna är en del av samma gruvpool (som vi kommer att diskutera inom kort), i vilket fall de kommer att kommunicera till
se till att de inkluderar en tydlig nonce i myntbastransaktionen för att undvika dubbelarbete.

Svårigheter. Exakt hur svårt är det att hitta en giltig block? Från och med mars 2015, målet för gruvsvårigheter (i hexadecimal) är:

0000000000000000172EC0000000000000000000000000000000000000000000

så hashen för alla giltiga block måste vara under detta värde. Med andra ord bara en i cirka 2^{67} nonces
att du försöker kommer att fungera, vilket är ett riktigt stort antal. En uppskattning är att den är större än mänsklig befolkning på jorden i kvadrat. Så, om varje person på jorden var sin egen planet Jorden med sju miljarder människor på det, skulle det totala antalet personer vara nära 2^{67}.

Fastställande av svårigheter. Gruv svårighet ändras var 2016 block, som finns om en gång varannan vecka. Den justeras utifrån hur effektiva gruvarbetarna var under perioden

tidigare 2016 block enligt denna formel:

next_difficulty = (föregående_svårighet * 2016 * 10 minuter) / (tid att ta bort de senaste 2016 blocken)

Observera att 2016*10 minuter är exakt två veckor, så 2016 block skulle ta två veckor att bryta 2016 block om ett block skapades exakt var tionde minut. Så effekten av denna formel är att skala svårigheter att underhålla egenskapen att block bör hittas av nätet i genomsnitt ungefär en gång var tionde minut. Det är inget speciellt med 2 veckor, men det är en bra avvägning. Om perioden var mycket kortare, svårigheten kan fluktuera på grund av slumpmässiga variationer i antalet block som finns i varje period. Om perioden var mycket högre, kan nätverkets hashkraft komma för långt ur balansera med svårigheten.

Varje Bitcoin-gruvarbetare beräknar självständigt svårigheten och accepterar bara block som uppfyller svårighet som de beräknade. Gruvarbetare som är på olika grenar kanske inte beräknar samma sak svårighetsvärde, men två gruvarbetare som bryter ovanpå samma block kommer överens om vad svårigheten är borde vara. Detta gör det möjligt att nå konsensus.

Du kan se i figur 5.3 att gruvsvårigheten med tiden fortsätter att öka. Det är inte nödvändigtvis en stadig linjär ökning eller en exponentiell ökning, men det beror på aktiviteten på marknaden. Brytning svårighetsgraden påverkas av faktorer som hur många nya gruvarbetare som ansluter sig, vilket i sin tur kan påverkas
med den nuvarande växelkursen för Bitcoin. I allmänhet, eftersom fler gruvarbetare kommer online och gruvhårdvara
blir effektivare, block hittas snabbare och svårighetsgraden ökar så att det alltid tar ca
tio minuter för att hitta ett kvarter.

I figur 5.3 kan du se att i den röda linjen på grafen finns en stegfunktion av svårighetsgrad även
även om den övergripande hashhastigheten för nätverket växer smidigt. Det diskreta steget är resultatet av det faktum att
svårighetsgraden justeras endast varje 2016 block.

Ett annat sätt att se nätverkets tillväxttakt är att överväga hur lång tid det tar att hitta en blockering på genomsnitt. Figur 5.4 (a) visar hur många sekunder som går mellan på varandra följande block i blocket kedja. Du kan se att detta gradvis går ner, hoppar upp och sedan gradvis går ner igen. Av vad som händer är att varje 2016 blockerar svårighetsåterställningen och den genomsnittliga blockeringstiden går tillbaka till cirka tio minuter. Under nästa period förblir svårigheten oförändrad, men mer och fler gruvarbetare kommer online. Eftersom hashkraften har ökat men svårigheten inte har blockerats hittas snabbare tills svårigheten återigen justeras efter 2016 block, eller cirka två veckor.

Figur 5.3: Gruvsvårigheter över tid (mitten-2014). Observera att y-axeln börjar vid 80 000 TH/s.

Figur 5.4 (a) : Dags att hitta ett block (början av 2014). Observera att y-axeln börjar vid 460 sekunder. På grund av
fortsatt snabb tillväxt av gruvkraft under denna tid minskade tiden för att hitta ett block stadigt
inom varje tvåveckorsfönster. Källa: bitcoinwisdom.com

Figur 5.4 (b): Dags att hitta ett block (tidigt 2015). Observera att y-axeln börjar vid 540 sekunder. Som den tillväxten av nätverket har avtagit, tiden för att hitta varje block är mycket närmare 10 minuter och är ibland över under perioder då nätverkets hashkraft faktiskt krymper. Källa: bitcoinwisdom.com

Även om målet var att ett block skulle hittas var tionde minut i snitt, under större delen av 2013 och 2014 var det närmare cirka nio minuter i snitt och skulle närma sig 8 minuter i slutet av varje tvåveckorscykel. Snabba beräkningar visar att detta kräver en häpnadsväckande tillväxt på 25 % varje två veckor, eller flera hundra gånger per år.

Föga överraskande var detta inte hållbart för alltid och 2015 har tillväxttakten varit mycket långsammare (och ibland negativt). I figur 5.4(b) kan vi se att när gruvkraften är närmare a

---

steady-state förblir perioden för att hitta varje block mycket närmare 10 minuter. Det kan till och med ta längre tid än 10 minuter, i vilket fall det kommer att finnas en svårighet *minskning* . En gång ansetts otänkbart, detta har hänt ganska regelbundet under 2015.

Även om det inte har skett några katastrofala minskningar av nätverkets gruvkraft hittills, finns det ingen inneboende orsak till att det inte kan hända. Ett föreslaget scenario för Bitcoins kollaps är en "död spiral” där en fallande växelkurs gör gruvdrift olönsam för vissa gruvarbetare, vilket orsakar en exodus, vilket i sin tur får priset att sjunka ytterligare.

5.2 Hårdvara för gruvdrift

Vi har nämnt att beräkningen som gruvarbetare måste göra är mycket svår. I det här avsnittet ska vi diskutera varför det är så beräkningsmässigt svårt och ta en titt på hårdvaran som gruvarbetare använder utföra denna beräkning.

Kärnan i de svåra beräkningar som gruvarbetare arbetar med är hashfunktionen SHA-256. Vi diskuterade hashfunktioner abstrakt i kapitel 1. SHA-256 är en kryptografisk hash för allmänt bruk funktion som är en del av en större familj av funktioner som standardiserades 2001 (SHA står för Secure Hash Algorithm). SHA-256 var ett rimligt val eftersom detta var den starkaste kryptografiska hashen funktion tillgänglig vid den tidpunkt då Bitcoin designades. Det är möjligt att det blir mindre säkert under Bitcoins livstid, men för närvarande är det säkert. Dess design kom från NSA (US National Security Agency), vilket har lett till vissa konspirationsteorier, men det anses allmänt vara det en mycket stark hashfunktion.

En närmare titt på SHA-256. Figur 5.5 visar mer i detalj om vad som faktiskt händer i en SHA-256 beräkning. Även om vi inte behöver känna till alla detaljer för att förstå hur Bitcoin fungerar, så är det

bra att ha en allmän uppfattning om uppgiften som gruvarbetare löser.

SHA-256 upprätthåller 256 bitar av tillstånd. Tillståndet är uppdelat i åtta 32-bitars ord vilket gör det mycket optimerad för 32-bitars hårdvara. I varje omgång tas ett antal ord i staten — några med små bitvisa justeringar tillämpas — och läggs ihop mod 32. Hela tillståndet flyttas sedan över med resultatet av att tillägget blev statens nya ord längst till vänster. Designen är löst inspirerad genom enklare bitvis linjära återkopplingsskiftregister (LFSR).

Sidofält: SHA-familjen. "256" i SHA-256 kommer från dess 256-bitars tillstånd och utdata. Tekniskt SHA-256 är en av flera närbesläktade funktioner i SHA-2-familjen, inklusive SHA-512 (som har ett större tillstånd och är därför säkrare). Det finns även SHA-1, en tidigare generation med 160-bitars utdata som nu anses osäker men är fortfarande implementerad i Bitcoin-skript.

Även om SHA-2-familjen, inklusive SHA-256, fortfarande anses vara kryptografiskt säker, nästa generations SHA-3-familj har nu blivit utvald av en tävling. SHA-3 är i slutskedet av standardisering idag, men det var inte tillgängligt när Bitcoin designades.

Sida 39

Figur 5.5 visar bara en omgång av SHA-256-kompressionsfunktionen. En komplett beräkning av SHA-256 gör detta i 64 iterationer. Under varje omgång tillämpas lite olika konstanter så att ingen iteration är exakt densamma.

Figur 5.5 : Strukturen för SHA-256. Detta är en omgång av komprimeringsfunktionen.

Uppgiften för gruvarbetare är att beräkna denna funktion så snabbt som möjligt. Kom ihåg att gruvarbetare är det tävla mot varandra så ju snabbare de gör detta, desto mer tjänar de. För att göra detta måste de kunna manipulera 32-bitars ord, gör 32-bitars modulär addition och gör även lite bitvis logik.

Som vi kommer att se inom kort kräver Bitcoin faktiskt att SHA-256 appliceras två gånger på ett block för att få hashen som används av noderna. Detta är en egenhet med Bitcoin. Orsakerna till dubbelberäkningen är inte helt specificerade, men vid det här laget är det bara något som gruvarbetare måste ta itu med.

. *CPU gruv* Den första generationen av gruv blev klar i datorer - det är generella centrala processorenheter (CPU). Faktum är att CPU mining var lika enkelt som att köra kod som visas i figur 5.6. Det vill säga, gruvarbetare sökte helt enkelt över nonces på ett linjärt sätt, beräknade SHA 256 i programvaran och kontrollerade om resultatet var ett giltigt block. Lägg också märke till i koden att som vi
nämnt, SHA-256 appliceras två gånger.

```
MÅL = (65535 << 208) / SVÅRHET;
coinbase_nonce = 0;
medan (1) {
header = makeBlockHeader(transaktioner, coinbase_nonce);
för (header_nonce = 0; header_nonce < (1 << 32); header_nonce++){
if (SHA256(SHA256(makeBlock(header, header_nonce))) <
MÅL)
ha sönder; //block hittades!
}
coinbase_nonce++;
}
```

Figur 5.6: CPU mining pseudokod.

Hur snabbt kommer detta att köras på en allmändator? På en avancerad stationär PC kan du förvänta dig det beräkna cirka 20 miljoner hash per sekund (MH/s). Med den hastigheten skulle det ta dig flera hundra tusen år i genomsnitt på tidigt 2015 svårighetsgrad (2^{67}) för att hitta ett giltigt block. Vi skojade inte när vi sa att gruvdrift skulle bli svårt!

Om du gruvdrift på en allmän PC idag, är CPU-utvinning inte längre lönsamt med strömmen svårighet. Under de senaste åren har den som försöker bryta på en CPU förmodligen inte förstår hur Bitcoin fungerar och var förmodligen ganska besviken över att de aldrig tjänade några pengar på det.

GPU gruvdrift. Den andra generationen började när folk började bli frustrerad med hur långsamt deras CPU:er var och använde istället sitt grafikkort, eller grafikprocessorenhet (GPU).

Nästan alla moderna datorer har en inbyggd GPU för att stödja högpresterande grafik. De är designade att ha hög genomströmning och även hög parallellitet, som båda är mycket användbara för Bitcoin-brytning. Bitcoin-brytning kan lätt parallelliseras eftersom du kan prova att beräkna flera hash samtidigt tid med olika nonces. 2010 släpptes ett språk som heter OpenCL. OpenCL är en general syfte språk för att göra andra saker än grafik på en GPU. Det är ett språk på hög nivå och över tid människor har använt det för att köra många typer av beräkningar snabbare på grafikkort. Detta banade sätt för Bitcoin-brytning på GPU:er.

Gruvdrift med grafikkort hade flera attraktiva egenskaper vid den tiden. Dels är de lätta tillgänglig och lätt för amatörer att ställa in. Du kan beställa grafikkort online eller köpa dem som mest stora hemelektronikbutiker. De är den mest tillgängliga avancerade hårdvaran som är tillgänglig för allmänheten. De har också några egenskaper som gör dem särskilt bra för Bitcoin-brytning. De är designade för parallellism så de har många aritmetiska logiska enheter (ALU) som kan användas för samtidiga SHA-256-beräkningar. Vissa GPU:er har också specifika instruktioner att göra bitvis operationer som är ganska användbara för SHA-256.

De flesta grafikkort kan också vara ***överklockat*** , vilket innebär att du kan köra dem snabbare än de är faktiskt designad för om du vill ta risken att de kan överhettas eller inte fungerar. Det här är en fastighet

spelare har efterfrågat i åratal. Med Bitcoin-brytning kan det vara lönsamt att köra chipet mycket snabbare än den var designad för även om du framkallar några fel genom att göra det.

Säg till exempel att du kan köra ditt grafikkort 50 procent snabbare men att göra det kommer att orsaka fel i SHA-256-beräkning till 30 procent av tiden. Om en ogiltig lösning felaktigt förklaras giltig av grafikkortet - något som skulle hända sällan - du kan alltid dubbelkolla det på din CPU. Å andra sidan, om en giltig lösning missas av misstag, skulle du aldrig veta. Men om din hastighet ökning från överklockning kan övervinna minskningen i produktionen på grund av fel, skulle du fortfarande komma ut
ett huvud. I exemplet ovan är genomströmningen 1,5x jämfört med att inte överklocka, medan framgångsfrekvensen är 0,7x. Produkten är 1,05, vilket innebär att överklockning ökar dina förväntade vinster med
5 %. Människor har lagt ner mycket tid på att optimera exakt hur mycket de borde överklocka en given chip för att maximera vinsten.

Slutligen kan du köra många grafikkort från ett moderkort och CPU. Så du kan ta din dator, som kommer att köra din faktiska Bitcoin-nod som samlar transaktioner från nätverk och sätter ihop block, och anslut flera grafikkort till det för att försöka hitta rätt nonces för att göra SHA-256 för blocket giltigt. Många människor skapade några riktigt intressanta hembryggda inställningar som den här som visas i figur 5.7 för att driva många, många GPU:er från en enda CPU. Detta var fortfarande inne
Bitcoins tidiga dagar när gruvarbetare fortfarande mestadels var hobbyister utan mycket erfarenhet av att springa servrar, men de kom fram till några ganska geniala konstruktioner för hur man packar många grafikkort på en liten plats och förvara dem tillräckligt svala för att fungera.

Figur 5.7: En hembyggda rack GPU som används för Bitcoin mining . Du kan också se fans som de brukade bygga ett primitivt kylsystem. Källa: LeonardH, cryptocurrenciestalk.com.
140

Nackdelar med GPU gruvdrift. GPU mining har vissa nackdelar. GPU:er har mycket hårdvara inbyggd i dem för att göra videobearbetning som inte kan användas för gruvdrift. Specifikt har de en stort antal flyttalsenheter som inte används alls i SHA-256. GPU:er har inte heller bästa kylningsegenskaper när du lägger många av dem bredvid varandra. De är inte designade att springa sida vid sida som de är på bilden; de är designade för att vara i en enda låda som gör grafik för en dator.

Miners vs. Gamers. Enligt folklore köpte Bitcoin-gruvarbetare 2011 tillräckligt många GPU:er för att stör den normala marknaden. Detta orsakade friktion med spelgemenskapen som hittade det alltmer svårt att hitta vissa populära grafikprocessorer i lokala elektronikbutiker. Intressant nog kan det dock ha gjort det ökat intresse för Bitcoin-brytning eftersom många av dessa frustrerade spelare lärde sig om valutan för att förstå vart alla GPU:er var på väg, med några av spelarna som själva blev gruvarbetare!

GPU:er kan också dra en ganska stor mängd ström, så det går åt mycket el i förhållande till en dator. En annan nackdel från början var att man var tvungen att antingen bygga sin egen bräda eller köpa dyra kort som rymmer flera grafikkort.

På en riktigt high-end grafikkort med aggressiv inställning du kan få så högt som 200 MH / s
eller 200
miljoner hash per sekund, en storleksordning bättre än vad du skulle göra med en CPU. Men
även med den förbättrade prestandan, och även om du är riktigt företagsam och använd hundra
GPU:er tillsammans skulle det fortfarande ta dig över 300 år i genomsnitt att hitta ett block i början av 2015
svårighetsgrad. Som ett resultat är GPU-brytning i princip död för Bitcoin idag, även om det fortfarande dyker upp
ibland i ett tidigt skede av altcoins.

FPGA gruvdrift. Omkring 2011 några gruvarbetare började byta från GPU till FPGA eller Field
Programmerbara Gate Arrays, efter att den första implementeringen av Bitcoin-gruvdrift kom ut i Verilog, en
hårdvarudesignspråk som används för att programmera FPGA:er. Den allmänna motiveringen bakom FPGA är att
försöka
för att komma så nära prestandan för anpassad hårdvara som möjligt samtidigt som ägaren till den
kort för att anpassa det eller omkonfigurera det "i fältet". Däremot är anpassade hårdvaruchips designade
i en fabrik och göra samma sak för alltid.

FPGA:er erbjuder bättre prestanda än grafikkort, särskilt vid "bitfiddling"-operationer som
är triviala att specificera på en FPGA. Kylning är också lättare med FPGA:er och till skillnad från GPU:er kan du
det
Använd teoretiskt nästan alla transistorer på kortet för gruvdrift. Precis som med GPU:er kan du packa
många FPGA:er tillsammans och kör dem från en central enhet, vilket är precis vad folk började göra
(se figur 5.8). Sammantaget var det möjligt att bygga ett stort utbud av FPGA:er snyggare och renare än du
kunde med grafikkort.

Genom att använda en FPGA med en noggrann implementering kan du få upp till en GH/s, eller en miljard hash
per
andra. Detta är förvisso en stor prestandavinst jämfört med CPU:er och GPU:er, men även om du hade hundra

---

kort tillsammans, var och en med en kapacitet på 1 GH/s, skulle det fortfarande ta dig cirka 50 år i genomsnitt
hitta ett Bitcoin-block vid tidig-2015 svårighetsgrad.

Figur 5.8: a. Hembyggda rack FPGA Även om du inte ser kylnings installation bilden här, en
rack som detta skulle behöva ett kylsystem.

Trots prestandavinsten var dagarna med FPGA-brytning ganska begränsade. För det första var de
drivs hårdare för Bitcoin-brytning - genom att vara på hela tiden och överklockad - än konsumentklass
FPGA var verkligen designade för. På grund av detta såg många människor fel och funktionsfel i deras
FPGA när de bröt. Det visade sig också vara svårt att optimera 32-bitars additionssteget som
är avgörande för att göra SHA-256. FPGA:er är också mindre tillgängliga - du kan inte köpa dem i de flesta butiker
och
det är färre som vet hur man programmerar och ställer in en FPGA än en GPU.

Men viktigast av allt, även om FPGA:er förbättrade prestanda var kostnadsprestanda
endast marginellt förbättrad jämfört med GPU:er. Detta gjorde att FPGA-brytning var ett ganska kortlivat
fenomen.

Medan GPU-brytning dominerade i ungefär ett år, var FPGA-brytningens dagar mycket längre begränsad — varar bara några månader innan anpassade ASIC:er kom.

ASIC gruvdrift. Mining i dag domineras av Bitcoin *ASIC* , eller ***applikationsspecifika integrerade kretsar*** . Dessa är marker som designades, byggdes och optimerades för det enda syftet att bryta Bitcoins. Det finns några få stora leverantörer som säljer dessa till konsumenter med en hel del variation: du kan välja mellan lite större och dyrare modeller, mer kompakta modeller, samt modeller med varierande prestanda och energiförbrukningspåståenden.

Att designa ASIC:er kräver avsevärd expertis och deras ledtid är också ganska lång. Ändå, Bitcoin ASICs designades och producerades förvånansvärt snabbt. I själva verket har analytiker sagt att detta
142

---

kan vara den snabbaste omloppstiden i historien om integrerade kretsar från att specificera ett problem och att ha ett fungerande chip i människors händer. Delvis som ett resultat av detta, de första generationerna av Bitcoin ASICs var ganska buggiga och de flesta av dem levererade inte riktigt den utlovade prestandan tal. Bitcoin ASICs har sedan mognat och det finns nu ganska pålitliga ASICs tillgängliga.

Fram till 2014 var livslängden för ASIC:er ganska kort på grund av den snabbt ökande nätverkshashhastigheten, med de flesta kort i den tidiga ASIC-eran som blev föråldrade på ungefär sex månader. Inom denna tid har huvuddelen av vinsten görs i förväg. Ofta kommer gruvarbetare att göra hälften av den förväntade vinsten för ASIC:s livslängd under bara de första sex veckorna. Detta innebar att frakthastigheten kan bli avgörande faktor för att göra en vinst. På grund av branschens omogenhet upplevde dock konsumenterna ofta leveransförseningar med brädor är ofta nästan föråldrade när de anlände. Som tillväxttakten för Bitcoins hashkraft har stabiliserats, gruvutrustning har en längre livslängd, men den tidiga eran såg många frustrerade kunder och anklagelser om bedrägeri från leverantörer.

Under mycket av Bitcoins historia har ekonomin i gruvdrift inte varit gynnsam för den lilla gruvarbetaren som vill gå online, beställa gruvutrustning och börja tjäna pengar. Faktum är att i de flesta fall människor som har lagt beställningar på gruvhårdvara skulle ha förlorat pengar baserat på beräkningen att de gjorde på den tiden. Fram till 2013 steg dock växelkursen för Bitcoin tillräckligt för att lösa de flesta kunder ute av att förlora pengar direkt. I själva verket har gruvdrift varit ett dyrt sätt att helt enkelt satsa att priset på Bitcoin skulle stiga, och många gruvarbetare – även om de har tjänat pengar på att bryta Bitcoins — skulle ha varit bättre om de bara hade tagit pengarna som de skulle spendera på gruvutrustning, investerade den i bitcoins och sålde dem så småningom med vinst.

Du kan fortfarande beställa Bitcoin gruvutrustning idag och vi skulle inte vilja avråda från det som ett sätt för att lära dig om Bitcoin och kryptovalutor. Vi noterar dock igen att detta inte är tillrådligt sätt att tjäna pengar. De flesta ASIC:er som säljs kommersiellt idag kommer sannolikt inte att betala för sig själva i gruvdrift
belöningar när du räknar in priset på el och kyla.

Idag. Professional gruv Idag mining har mest rört sig bort från individer och mot professionella gruvcentra. Exakta detaljer om hur dessa centra fungerar är inte särskilt välkända eftersom företag vill skydda sina installationer för att behålla en konkurrensfördel. Förmodligen, dessa verksamheter upprätthåller lönsamheten genom att köpa något nyare och mer effektiva ASIC än vad som är tillgänglig för allmän försäljning med massrabatt. I figur 5.9 ser vi en bild av en professionell gruvdrift centrum i Republiken Georgien.

Sida 45

Bild 5.9: BitFury gruvcenter, ett professionellt gruvcenter i republiken Georgien.

När man bestämmer var man ska etablera ett gruvcenter är de tre största övervägandena: klimat, kostnad för el och nätverkshastighet. I synnerhet vill du ha ett kallt klimat för att hålla kylkostnaderna låga. Kylning är särskilt utmanande med Bitcoin-brytning, som beräknas använda en order på magnitud mer el per kvadratfot än traditionella datacenter (och därmed avge en order av storleksordningen mer värme). Du vill uppenbarligen ha billig el. Du vill också ha ett snabbt nätverk anslutning för att vara väl ansluten till andra noder i Bitcoin peer-to-peer-nätverket så att du kan hör om nya block så snabbt som möjligt efter att de har annonserats. Georgien och Island har enligt uppgift varit populära destinationer för Bitcoin gruvdatacenter.

Likheter med guldgruvor. Medan "Minings kan tyckas vara bara en söt namn, om vi steg tillbaka och tänka om utvecklingen av gruvdrift kan vi se intressanta paralleller mellan Bitcoin-brytning och guld brytning. Till att börja med såg båda en liknande guldrush-mentalitet med många unga amatörer ivriga att komma in i verksamheten så snart som möjligt.

Medan vi med Bitcoin-brytning har sett en långsam utveckling från CPU:er till GPU:er till FPGA:er, till nu ASIC:er,
guldbrytningen såg en utveckling från individer med guldpannor till små grupper av människor med sluss lådor, till placer gruvdrift — bestående av stora gruvgrupper som spränger bort sluttningar med vatten — till modern guldbrytning som ofta utnyttjar gigantiska dagbrott för att utvinna tonvis med råmaterial ur jorden (se figur 5.10). Både med Bitcoin och med guld, vänligheten och tillgängligheten till individer har gått ner över tiden och stora företag har så småningom konsoliderat det mesta verksamhet (och vinster). Ett annat mönster som framkommit på båda ställena är att större delen av vinsten har tjänats in av dem som säljer utrustning, vare sig det gäller guldpannor eller ASIC:er för gruvdrift, på bekostnad av
individer som hoppas att bli rika.

Sida 46

Figur 5,10: Utveckling av gruvdrift. Vi kan se en tydlig parallell mellan utvecklingen av Bitcoin gruvdrift och utvecklingen av guldbrytning. Båda var från början vänliga mot individer och över tid blev massiv verksamhet kontrollerad av stora företag.

Framtiden. För närvarande ASIC mining är det enda realistiska sättet att vara lönsam i Bitcoin och det är inte mycket vänlig mot små gruvarbetare. Detta väcker några frågor om vad som kommer att hända framöver. Är små gruvarbetare ur Bitcoin-brytning för alltid, eller finns det ett sätt att återinföra dem? Dessutom gör det ASIC gruvdrift och utveckling av professionella gruvcenter bryter mot den ursprungliga visionen om Bitcoin som skulle ha ett helt decentraliserat system där varje individ i nätverket minerade

på sin egen dator?

Dessutom, om detta verkligen är ett brott mot Satoshi Nakamotos ursprungliga vision för Bitcoin, skulle vi vara bättre med ett system där det enda sättet att bryta var med processorer? I kapitel 8 kommer vi att överväga dessa frågor och titta på idéer för alternativa former som kan vara mindre vänliga för ASIC.

Cykeln upprepas. Det är också värt att notera här att flera mindre altcoins faktiskt har använt en annat pussel än SHA-256, men har sett en liknande bana inom gruvdrift som Bitcoin. Vi ska diskutera dessa altcoins mer i kapitel 9 men minns att det för ASICs fortfarande är lång ledtid mellan dem designa ett chip och skicka det, så om ett nytt altcoin använder ett nytt pussel (även bara en modifierad version av SHA-256), kommer detta att kosta en tid då ASIC:er ännu inte är tillgängliga. Normalt kommer gruvdrift att fortsätta
precis vid Bitcoin gjorde från CPU:er till GPU:er och/eller FPGA:er till ASIC:er (om altcoin är mycket framgångsrik, som
LiteCoin).

Således kan en strategi för mindre gruvarbetare vara att försöka skapa nya altcoins som ännu inte är värdefulla tillräckligt för stora gruvkoncerner att investera i – precis som små guldgruvarbetare som har drivits ut ur

---

beprövade guldfält kan försöka prospektera oprövade nya områden. Det betyder naturligtvis att pionjärerna är det står inför en betydande risk att altcoin aldrig kommer att lyckas.

5.3 Energiförbrukning och ekologi

Vi såg hur stora professionella gruvdatacenter har tagit över verksamheten inom Bitcoin gruvdrift, och hur detta liknar rörelsen till gropbrytning i guldbrytning. Du kanske är medveten om att gruvor har varit en stor källa till oro under åren på grund av skadorna de orsakar
miljö. Bitcoin är inte riktigt på den nivån ännu, men det börjar använda en betydande mängd energi som har blivit ett diskussionsämne. I det här avsnittet kommer vi att se hur mycket energi Bitcoin gruvdrift använder och vilka konsekvenserna är för både valutan och för vår planet.

Termodynamiska gränser. Det finns en fysisk lag som kallas ***Landauer princip*** som utvecklats av Ralph Landauer på 1960-talet som säger att varje icke-reversibel beräkning måste använda ett minimibelopp av energi. Logiskt irreversibla beräkningar kan ses som sådana som förlorar information.
Specifikt de princip anger att radera någon bit måste konsumera ett minimum av (kT ln 2) joule,
där k är Boltzmanns konstant (ca 1,38 x 10^{-23}
J / K), T är temperaturen hos kretsen
i kelvin, och ln 2 är den naturliga logaritmen av 2, ungefär 0,69. Detta är en liten mängd energi per bit, men detta ger en hård nedre gräns för energianvändning från grundläggande fysik.

Vi ska inte gå igenom härledningen här, men idén på hög nivå är att varje gång du vänder en bit på ett icke-reversibelt sätt finns ett minsta antal joule som du måste använda. Energi är aldrig förstört; den omvandlas från en form till en annan. I fallet med beräkning är energin mestadels omvandlas från elektricitet, som är användbar, högkvalitativ energi, till värme som försvinner in i miljön.

Som en kryptografisk hashfunktion är SHA-256 inte en reversibel beräkning. Vi kan minnas från Kapitel 1 att detta är ett grundläggande krav för kryptografiska hashfunktioner. Så, eftersom icke-reversibel beräkning måste använda lite energi och SHA-256 – grunden för Bitcoin-brytning – är inte reversibel, energiförbrukning är ett oundvikligt resultat av Bitcoin-brytning. Som sagt, de gränser som Landauers sätter principen är långt, långt under den mängd el som används idag. Vi är ingenstans i närheten den teoretiskt optimala förbrukningen av beräkningar, men även om vi nådde det teoretiska optimala vi skulle fortfarande använda energi för att utföra Bitcoin-brytning.

Hur använder Bitcoin gruvdrift energi? Det finns tre steg i processen som kräver energi, några varav kanske inte är så självklart:

1. förkroppsligad energi. Första, Bitcoin gruvutrustning behöver tillverkas. Detta kräver fysisk brytning av råvaror samt att förvandla dessa råvaror till en Bitcoin gruv ASIC, som båda kräver energi. Detta är den förkroppsligade energin. Så snart du får en Bitcoin-brytning ASIC på posten, du har redan förbrukat mycket energi — inklusive fraktenergin, förstås – innan du ens har slagit på den!

Förhoppningsvis kommer den förkroppsligade energin med tiden att sjunka när mindre och mindre ny kapacitet kommer online. Som färre människor går ut för att köpa nya ASIC:er för gruvdrift, de kommer att bli föråldrade mindre snabbt, och den förkroppsligade energin kommer att amorteras över år och år av gruvdrift.

2. el. När ASIC är påslagen och gruvdrift, förbrukar den el. Detta är steget som vi vet måste förbruka energi på grund av Landauers princip. När gruvriggar blir effektivare, kostnaden för el kommer att sjunka. Men på grund av Landauers princip vet vi att det aldrig kommer att ske försvinna; elektrisk energiförbrukning kommer att vara ett faktum för Bitcoin-gruvarbetare för alltid.

3. kylning. En tredje viktig komponent i gruvdrift som förbrukar energi kylning av din utrustning för att säkerställa att den inte fungerar fel. Om du arbetar i liten skala i en mycket kyla klimat kan din kylkostnad vara trivial, men även i kallt klimat när du väl får tillräckligt med ASIC i en litet utrymme måste du betala extra för att kyla av din utrustning från all spillvärme som det genererar. I allmänhet kommer energin som används för att kyla av gruvutrustning också vara i form av el.

Gruvdrift i stor skala. Både förkroppsligad energi och elektricitet minskning (per enhet av gruvarbete avslutats) när man arbetar i stor skala. Det är billigare att bygga chips som är designade för att köras i stora data center, och du kan leverera strömmen mer effektivt eftersom du inte behöver så många nätaggregat.

När det kommer till kylning är det dock oftast motsatsen: kylkostnaderna tenderar att öka större din skala är. Om du vill driva en mycket stor verksamhet och ha mycket Bitcoin-brytning utrustning allt på ett ställe, det finns mindre luft för värmen att avledas i området runt din Utrustning. Din kylbudget kommer därför att öka i skala (per slutfört gruvarbete) såvida du inte skalar ditt fysiska område tillsammans med antalet marker du har i bruk.

Uppskatta energianvändningen. Hur mycket energi hela Bitcoin systemet använder? Självklart kan vi inte beräkna detta just för att det är ett decentraliserat nätverk med gruvarbetare som arbetar överallt

utan att dokumentera exakt vad de gör. Men det finns två grundläggande tillvägagångssätt för att uppskatta hur mycket energi Bitcoin-gruvarbetare använder tillsammans. Vi ska göra lite baksidan av kuvertet beräkningar här baserade på tidiga 2015 års värden. Vi måste betona att dessa siffror är mycket grova, både för att vissa av parametrarna är svåra att uppskatta och för att de ändras snabbt. I bästa fall de bör behandlas som uppskattningar av storleksordning.

Top-down-strategi. Den första metoden är en top-down-strategi. Vi börjar med det enkla faktum att varje gång ett block hittas idag ges 25 bitcoins, värda cirka 6 500 amerikanska dollar, till gruvarbetarna. Det är cirka 11 dollar varje sekund, skapas ur tomma intet i Bitcoin-ekonomin och ges till gruvarbetarna.

Låt oss nu ställa den här frågan: om gruvarbetarna förvandlar alla dessa 11 dollar per sekund till elektricitet, hur mycket kan de köpa? Naturligtvis spenderar inte gruvarbetarna alla intäkter på el, men detta kommer att ge en övre gräns för den elektricitet som används. Elpriserna varierar mycket, men

---

vi kan använda som en uppskattning att el kostar cirka 10 cent per kilowattimme (kWh) vid en industriränta i USA, eller motsvarande 3 cent per megajoule (MJ). Om Bitcoin-gruvarbetare spenderade alla 11 dollar per sekund av inkomster som köper el, kunde de köpa 367 megajoule per andra, förbrukar stadiga 367 megawatt (MW).

Enheter av energi och kraft. I internationella enhetssystemet (SI), är energi mäts i ***joule*** . En ***watt*** är en enhet för effekt, där en ***watt*** definieras som en joule per sekund.

Nedifrån och upp. Ett annat sätt att beräkna kostnaden är att använda en bottom-up-strategi. I denna tillvägagångssätt tittar vi på antalet hash som gruvarbetarna faktiskt beräknar, vilket vi känner till observera svårigheten för varje block. Om vi då antar att alla gruvarbetare använder det mest effektiva hårdvara kan vi härleda en nedre gräns för elförbrukningen.

För närvarande är den bästa effektivitetssiffran bland kommersiellt tillgängliga gruvriggar cirka 3 GH/s/W. Det vill säga, de mest avancerade ASIC:erna hävdar att de utför tre miljarder hash per sekund medan förbrukar 1 watt ström. Den totala nätverkshashratet är cirka 350 000 000 GH/s, eller motsvarande 350 petahashes per sekund (PH/s). Multiplicerar vi dessa två tillsammans ser vi att det tar cirka 117 MW för att producera så många hash per sekund med den effektiviteten. Naturligtvis exkluderar denna siffra alla kylningsenergin och all förkroppsligande energi som finns i dessa chips, men vi gör det optimala beräkning och härledning av en nedre gräns så det är okej.

Genom att kombinera "top down" och "bottom up" tillvägagångssätt kan vi härleda en uppskattning av beloppet av ström som används för Bitcoin-gruvarbetare är förmodligen i storleksordningen några hundra MW.

Hur mycket är en megawatt? För att bygga upp intuitionen kan vi se hur mycket stora kraftverk producerar. Three Gorges Dam i Kina är ett av de största kraftverken i världen och har en kraft på 10 000 MW växt. Ett typiskt stort vattenkraftverk producerar cirka 1 000 MW. Den största nukleära kraftverk i världen, Kashiwazaki-Kariwa i Japan, är en 7000 MW anläggning, medan den genomsnittliga kärnkraftverket är cirka 4 000 MW. En stor koleldad anläggning producerar cirka 2 000 MW.

Enligt våra uppskattningar då, förbrukar hela Bitcoin-nätverket kanske 10% av ett stort kraftverkets värde av el. Även om detta är en betydande mängd kraft, är den fortfarande liten

jämfört med alla andra saker som människor använder el till på planeten.

Är Bitcoin mining slösaktig? Det sägs ofta Bitcoin ”avfall” energi eftersom den energi som förbrukas på SHA-256-beräkningar som inte tjänar något annat användbart syfte. Det är viktigt att inse, dock att alla betalningssystem kräver energi och el. Med traditionell valuta, avsevärd energi förbrukas vid utskrift av valuta och drift av bankomater, myntsortering maskiner, kassaregister och betalningshanteringstjänster, samt transport av pengar och guld ädelmetaller i pansarbilar. Du kan lika gärna hävda att all denna energi är "slösad" eftersom den inte gör det tjäna något syfte förutom att upprätthålla valutasystemet. Så, om vi värderar Bitcoin som en användbar valutasystemet, så slösas inte den energi som krävs för att stödja det verkligen bort.
148

Ändå, om vi kunde ersätta Bitcoin-gruvdrift med ett mindre energikrävande pussel och fortfarande ha en säker valuta, skulle detta vara en positiv förändring. Vi kommer dock att se i kapitel 8 att vi inte vet om det är faktiskt möjligt

Återanvända energi. En annan idé att göra Bitcoin mer miljövänlig är att fånga den värme som alstras från Bitcoin mining gör något användbart med det istället för att bara värma upp atmosfären. Den här modellen att fånga spillvärme från beräknings kallas tillvägagångssätt kallas *uppgifter ugnen* tillvägagångssätt. Konceptet är att istället för att köpa en traditionell elvärmare för att värma ditt hem, eller för att värma vatten i ditt hem kunde du köpa en värmare som fungerade som en Bitcoin gruvrigg, bryta bitcoins och värma upp ditt hem som en biprodukt av den beräkningen. Det visar sig att effektiviteten av att göra detta är inte mycket värre än att köpa en elvärmare, och det här skulle kanske inte vara mer komplicerat för en hemkonsument än att koppla in sin värmare till sin internetanslutning samt sin el utlopp.

Det finns några nackdelar med detta tillvägagångssätt. Även om det är ungefär lika effektivt som att använda en elektrisk värmare är elektriska värmare i sig mycket mindre effektiva än gasvärmare. Dessutom, vad händer när alla stänger av sin Bitcoin gruvrigg under sommaren (eller åtminstone alla i Norra halvklotet)? Mining hash power kan minska säsongsmässigt baserat på hur mycket värme människor behöver. Det kan till och med gå ner på dagar som råkar vara varmare än genomsnittet! Detta skulle orsakade många intressanta effekter för Bitcoin-konsensus om dataugnsmodellen faktiskt höll på.

Ägandefrågan är inte heller klar. Om du köper en Bitcoin-dataugn, äger du Bitcoin gruvbelöningar som du får, eller gör företaget som sålde dem till dig? De flesta människor har inte något intresse för Bitcoin-brytning – och kommer förmodligen aldrig att göra det – så det kan vara mer meningsfullt att köpa det som en apparat och låt företaget som sålde den till dig behålla belöningarna. Detta kan betyda värmaren säljs med en liten förlust då, i vilket fall vissa företagsamma användare kanske vill köpa dem och ändra dem att behålla gruvbelöningarna för sig själva, vilket leder till en potentiellt ful DRM (Digital Rights Management) strid.

Turning el i kontanter. En annan långsiktig fråga som Bitcoin är att det kan ge det mest effektiva sättet att förvandla el till kontanter. Föreställ dig en värld där Bitcoin-gruvdrift ASIC:er finns en lättillgänglig vara och den dominerande kostnaden för gruvdrift är elektricitet. I själva verket skulle detta innebär att tillhandahållande av gratis eller lågkostnadselektricitet är öppet för nya former av missbruk.

I många länder runt om i världen subventionerar regeringar el, särskilt industri
elektricitet. Bland annat gör de det ofta för att uppmuntra industrin att placeras i deras
Land. Men Bitcoin ger ett bra sätt att förvandla el till kontanter, vilket kan orsaka regeringar
att tänka om den modellen om deras subventionerade el omvandlas en masse till bitcoins. Elektricitet
Subventionerna är avsedda att attrahera företag som ska bidra till landets ekonomi och arbetskraft
marknadsföra och subventionera Bitcoin-brytning kanske inte har den avsedda effekten.

Sida 51

Ett ännu större problem är de miljarder fritt tillgängliga eluttag runt om i världen
människors hem, universitet, hotell, flygplatser, kontorsbyggnader och så vidare. Folk kanske försöker koppla in
gruvutrustning så att de kan tjäna medan någon annan betalar elräkningen. Faktum är att de
kan använda föråldrad hårdvara och inte bry sig om att uppgradera, med tanke på att de inte kommer att betala
elräkning. Det är ganska skrämmande att överväga möjligheten att övervaka varje eluttag i
värld av för potentiella obehöriga använde en elkälla för Bitcoin-brytning.

5.4 Gruvbassänger

Tänk på ekonomin med att vara en liten gruvarbetare. Anta att du är en individ som spenderade 6 000 $ av
dina surt förvärvade pengar för att köpa en fin, glänsande, ny Bitcoin gruvrigg. Säg att föreställningen är
så att du förväntar dig att hitta ett block var 14:e månad (och kom ihåg att ett block är värt ungefär
6 500 dollar i början av 2015).

Amorterat är den förväntade inkomsten för din gruvarbetare kanske 400 USD per månad när du räknar in
el och andra driftskostnader. Om du faktiskt fick en check med posten varje månad för $400, det
skulle vara mycket vettigt att köpa gruvriggen. Men kom ihåg att gruvdrift är en slumpmässig process. Du
vet inte när du kommer att hitta nästa block, och tills det händer kommer du inte att tjäna något.

Hög varians. Om vi tittar på fördelningen av hur många block du sannolikt att hitta i det första året,
variansen är ganska hög och det förväntade antalet är ganska lågt. Eftersom du hittar block på en fast,
låg hastighet som är oberoende av tiden sedan det senaste blocket du hittade, ditt förväntade antal
block är mycket väl approximeras med en ***Poisson-fördelning*** . En Poissonfördelning uppstår om du har N
oberoende försök var och en med en chans λ / N av framgång som N närmar sig oändligheten. Med Bitcoin
gruvdrift,
varje individuell nonce försök är i själva verket en slumpmässig studie med en liten chans att lyckas, så N är
verkligen mycket stor även för små gruvarbetare och approximationen är mycket bra.

Om du förväntar dig att hitta på ett kvarter per 14 månader (en Poisson-fördelning med λ = 6/7 block / år),
det finns en större än 40 % chans att du inte hittar några block inom det första året. För en
enskild gruvarbetare kan detta vara förödande. Du spenderade tusentals dollar på gruvarbetaren, betalade in
massor
el för att driva den, och fick ingenting i gengäld. Det finns ungefär 36 % chans att du hittar en
blockera inom det första året vilket betyder att du kanske knappt skrapar förbi, förutsatt din el
kostnaderna var inte för höga. Slutligen, det finns en mindre chans att du hittar två eller flera block, i vilka
i fall du kan göra ut med en bra vinst.

Figur 5.11: Illustration av osäkerhet i gruvdrift. Förutsatt att den globala hashhastigheten är konstant och medeltiden för att hitta ett block är 14 månader, variansen för en liten gruvarbetare är ganska hög.

Dessa siffror är bara ungefärliga, men huvudpoängen här är att även om du förväntar dig kanske går bra - det vill säga tjänar tillräckligt för att göra en avkastning på din investering - variansen är tillräckligt hög för att det finns en stor chans att du inte tjänar något alls. För en liten gruvarbetare betyder detta gruvdrift är en stor chansning.

Gruv pooler. Historiskt sett när små företagare står inför en hel del risk, bildade de ömsesidiga försäkringsbolag för att minska den risken. Bönder skulle till exempel gå samman och komma överens om att om varje enskild bondes lada som brann ner, de andra skulle dela sin vinst med den bonden. Skulle kunna har vi en ömsesidig försäkringsmodell som fungerar för små Bitcoin-gruvarbetare?

En gruvpool är precis det - ömsesidig försäkring för Bitcoin-gruvarbetare. En grupp gruvarbetare kommer att bilda en pool och alla försöker bryta ett block med en utsedd myntbasmottagare. Den mottagaren kallas poolchef. Så, oavsett vem som faktiskt hittar blocket, kommer poolchefen att få belöningarna. Poolchefen kommer att ta den inkomsten och fördela den till alla deltagare i poolen baserat på hur mycket arbete varje deltagare faktiskt utförde. Naturligtvis kommer poolchefen också förmodligen ta någon form av nedskärning för deras tjänst att hantera poolen.

Förutsatt att alla litar på poolchefen fungerar detta utmärkt för att minska gruvarbetarnas varians. Men hur vet en poolchef hur mycket arbete varje medlem i poolen faktiskt utför? Hur kan poolchefen dela intäkterna i proportion till hur mycket arbete varje gruvarbetare utför? Uppenbarligen vill poolchefen inte bara ta allas ord för det eftersom folk kanske hävda att de har gjort mer än de faktiskt gjorde.

Gruvaktier. Det finns en elegant lösning på detta problem. Gruvarbetare kan sannolikt bevisa hur mycket arbete de gör genom att mata ut *aktier* , eller nästan giltiga block. Säg att målet är ett nummer börjar med 67 nollor. Ett blocks hash måste vara lägre än målet för att blocket ska vara giltigt. I den process för att söka efter ett sådant block, kommer gruvarbetare att hitta några block med hash som börjar med mycket

nollor, men inte riktigt 67. Gruvarbetare kan visa dessa nästan giltiga block för att bevisa att de verkligen är arbetssätt. En andel kan kräva säg 40 eller 50 nollor, beroende på vilken typ av gruvarbetare poolen är inriktad på för.

Figur 5.12. Gruvaktier Miners försöker ständigt hitta block med en hash under målet. I den process, kommer de att hitta andra block vars hash innehåller färre nollor — men som fortfarande är sällsynta nog bevisa att de har arbetat hårt. I den här figuren är de tråkiga gröna hasharna aktier, medan ljusgrön hash är från ett giltigt block (som också är en giltig andel).

Poolförvaltaren kommer också att köra en Bitcoin-nod på uppdrag av deltagarna, samla in transaktioner och sätt ihop dem till ett block. Chefen kommer att inkludera sin egen adress i myntbastransaktionen och skicka blocket till alla deltagare i poolen. Alla pooldeltagare arbetar på detta block, och de bevisar att de har arbetat med det genom att skicka in aktier.

När en medlem i poolen hittar ett giltigt block skickar de det till poolchefen som distribuerar belöning i proportion till mängden utfört arbete. Gruvarbetaren som faktiskt hittar blocket är det inte tilldelas en speciell bonus, så om en annan gruvarbetare gjorde mer arbete än, kommer den andra gruvarbetaren att få mer betalt
även om det inte var de som hittade ett giltigt block. Se figur 5.13.
152

Figur 5.13: Gruvbelöningar. Tre deltagare på bilden här arbetar alla på samma block. De tilldelas i proportion till mängden utfört arbete. Även om gruvarbetaren till höger var det den som hittar det giltiga blocket, gruvarbetaren till vänster får mer betalt eftersom denna gruvarbetare gjorde mer arbete.
Det är (vanligtvis) ingen bonus utbetald till gruvarbetaren som faktiskt hittar blocket.

Det finns några alternativ för exakt hur poolchefen beräknar hur mycket den ska betala var och en gruvarbetare baserat på de aktier de lämnar in. Vi ska titta på två av de vanliga, enklare. Det finns många andra system som också används, men dessa kommer att illustrera avvägningarna mellan belöning system.

Pay-per-aktie. I pay per aktie modell betalar poolansvarige en fast avgift för varje aktie över en viss svårighet för blocket som poolen arbetar på. I den här modellen kan gruvarbetare skicka sina aktier till poolchefen direkt och få betalt utan att vänta på att poolen ska hitta ett block.

På vissa sätt är pay-per-share-modellen den bästa för gruvarbetare. De är garanterade ett visst belopp pengar varje gång de hittar en andel. Poolchefen absorberar i princip all risk eftersom de betala belöningar även om ett block inte hittas. Naturligtvis, som ett resultat av den ökade risken, i betala-per-aktie-modellen kommer poolförvaltaren förmodligen att ta ut högre avgifter jämfört med andra modeller.

Ett problem med pay-per-share-modellen är att gruvarbetare faktiskt inte har några incitament att skicka giltiga block till poolchefen. Det vill säga att de kan kassera giltiga block men ändå få samma betalt belöningar, vilket kommer att orsaka en stor förlust för poolen. En illvillig poolchef kan attackera en konkurrerande
slå samman på detta sätt för att försöka driva dem i konkurs.
153

Proportionella. I den proportionella modellen istället för att betala en fast avgift per aktie, mängden

betalningen beror på om poolen faktiskt hittat ett giltigt block eller inte. Varje gång ett giltigt block är fann att belöningarna från det blocket delas ut till medlemmarna i proportion till hur mycket arbete det gjorde de faktiskt.

I den proportionella modellen bär gruvarbetarna fortfarande en viss risk proportionell mot risken för poolen i allmän. Men om poolen är tillräckligt stor, kommer variansen av hur ofta poolen hittar block att vara ganska låg. Proportionella utbetalningar ger lägre risk för poolförvaltaren eftersom de bara betalar ut när giltiga block har hittats. Detta kommer också runt problemet som vi nämnde med pay-per-share modell, eftersom gruvarbetare uppmuntras att skicka in de giltiga block som de hittar eftersom det utlöser inkomsterna kommer tillbaka till dem.

Den proportionella modellen kräver lite mer arbete på uppdrag av poolcheferna för att verifiera, beräkna och dela ut belöningar jämfört med den platta betal-per-aktie-modellen.

Pool hoppande. Ävenmed bara dessa två typer av pooler, kan vi se att gruvarbetarna kan stimuleras att växla mellan bassängerna vid olika tidpunkter. För att se detta, anser att en rent proportionell pool kommer effektivt att betala ut ett större belopp per aktie om ett block hittas snabbt, eftersom det alltid betalar en blockera belöning oavsett hur lång tid det har gått sedan det senaste blocket hittades.

En smart gruvarbetare kan försöka bryta i en proportionell pool tidigt i cykeln (strax efter föregående block hittades) medan belöningarna per aktie är relativt höga, bara för att byta ("hoppa") till en pay-per-share pool senare i cykeln, när de förväntade belöningarna från gruvdrift i den proportionella poolen är relativt låg. Som ett resultat av detta är proportionella pooler inte riktigt praktiska. Mer komplicerade upplägg, som t.ex ”Pay per sista N aktier inlämnade” är vanligare, men även dessa är föremål för subtila pool hoppbeteende. Det är fortfarande öppet hur man utformar ett belöningssystem för gruvpooler som inte är sårbart till denna typ av manipulation.

Historia och standardisering. Gruv pooler började runt 2010 GPU era Bitcoin mining. De blev omedelbart mycket populära av den uppenbara anledningen att de sänkte variansen för deltagande gruvarbetare. De har blivit ganska avancerade nu. Det finns många protokoll för hur man kör gruvpooler och det har till och med föreslagits att dessa gruvpoolsprotokoll bör standardiseras som en del av själva Bitcoin. Precis som det finns ett Bitcoin-protokoll för att köra peer-to-peer-nätverket, gruvpoolsprotokoll tillhandahåller ett kommunikations-API för poolhanteraren för att skicka alla medlemmar detaljerna om blocket att arbeta på och för gruvarbetarna att skicka tillbaka aktierna till poolchefen som de hittar. getblocktemplate (GBT) är officiellt standardiserad som en Bitcoin-förbättring Förslag (BIP). Ett konkurrerande protokoll, Stratum, är för närvarande mer populärt i praktiken och är ett förslag BIP. Till skillnad från själva Bitcoin-protokollet är det bara en mindre olägenhet att ha flera inkompatibla gruvpoolsprotokoll. Varje pool kan helt enkelt välja vilket protokoll de vill och marknaden kan besluta.

En del gruvhårdvara stöder till och med dessa protokoll på hårdvarunivå, vilket i slutändan kommer att göra det begränsa deras utvecklingsflexibilitet något. Detta gör det dock väldigt enkelt att köpa en bit av gruvhårdvara och gå med i en pool. Du kopplar bara in den i väggen — både elen och ditt nätverk anslutning — välj en pool, och sedan börjar den omedelbart få instruktioner från poolen, bryta och omvandla din el till pengar.

51% gruv pooler. Såsom av tidig 2015, den stora majoriteten av alla gruvarbetare gruv via pooler med mycket några gruvarbetare som gruvdrift "solo" längre. I juni 2014 blev Ghash.io, den största gruvpoolen, så stor att den hade faktiskt över 50 % av hela kapaciteten över Bitcoin-nätverket. Ghash erbjöd i huvudsak sådana en hel del till deltagande gruvarbetare som majoriteten ville ansluta sig till.

Detta är något som folk hade fruktat länge och detta ledde till en motreaktion mot Ghash. Förbi augusti hade Ghashs marknadsandel minskat designmässigt eftersom de slutade ta emot nya deltagare. Ändå kontrollerade två gruvpooler ungefär hälften av strömmen i nätet.

Figur 5.14 (a) Hash-kraft genom gruvpool, via blockchain.info (juni 2014)

155

Sida 57

Figur 5.14 (b) Hash-kraft genom gruvpool, via blockchain.info (augusti 2014)

Figur 5,14 (c) Hash driver vid gruv pool, via blockchain.info (April 2015)

156

Sida 58

I april 2015 ser situationen väldigt annorlunda och mindre koncentrerad ut, åtminstone på ytan. De möjligheten att en pool skaffar 51% är fortfarande ett problem i samhället, men den negativa publiciteten GHash mottagen har lett pooler för att undvika att bli för stora sedan dess. Som nya gruvarbetare och pooler har kommit in på marknaden och standardiserade protokoll har gjort det lättare att byta mellan pooler för gruvarbetare har marknadsandelen för olika pooler förblivit ganska flytande. Det återstår att se hur saker kommer att utvecklas på lång sikt.

Det är dock värt att notera att gruvpooler kan gömma den faktiska koncentrationen av gruvkraft i händerna på några få stora gruvorganisationer som kan delta i flera gruvpooler samtidigt för att dölja deras verkliga storlek. Denna praxis kallas *tvätta hashar.* Det är fortfarande okänd hur koncentrerad fysisk kontroll av gruvhårdvara faktiskt är och gruvpooler gör detta ganska svårt att avgöra utifrån.

Är gruv pooler bra? Fördelarna med gruv pooler är att de gör gruv mycket mer förutsägbara för deltagarna och de gör det lättare för mindre gruvarbetare att engagera sig i spel. Utan gruvpooler skulle variansen göra gruvdrift omöjlig för många små gruvarbetare.

En annan fördel med gruvpooler är att eftersom det finns en central poolchef som sitter på nätverk och montering av block gör det enklare att uppgradera nätverket. Uppgradering av programvaran att mining pool manager körs som effektivt uppdaterar programvaran som hela poolen medlemmar kandiderar.

Den största nackdelen med gruvpooler är förstås att de är en form av centralisering. Det är en öppen fråga hur mycket makt operatörerna av en stor gruvpool faktiskt har. I teorin gruvarbetare

är fria att lämna en pool om den upplevs som för kraftfull, men det är oklart hur ofta gruvarbetare gör det i öva.

En annan nackdel med gruvpooler är att det minskar antalet personer som faktiskt driver en fullständigt validerar Bitcoin-noden. Tidigare var alla gruvarbetare, oavsett hur små, tvungna att driva sina egna helt
validerande nod. De var alla tvungna att lagra hela blockkedjan och validera varje transaktion. Nu, de flesta gruvarbetare överför den uppgiften till sin poolchef. Detta är huvudorsaken till att, som vi nämnde i I kapitel 3 kan antalet fullt validerade noder faktiskt minska i Bitcoin-nätverket.

Om du är orolig över nivån av centralisering som introduceras av gruvpooler, kan du fråga: kunde vi designar om gruvprocessen så att vi inte har några pooler och alla måste bryta för sig själva? Vi kommer att överväga denna fråga i kapitel 8.

5.5 Incitament och strategier för gruvdrift

Vi har ägnat större delen av det här kapitlet åt att beskriva hur den största utmaningen med att vara gruvarbetare är att bli bra
hårdvara, hitta billig el, komma igång så fort du kan och hoppas på lite

---

lycka till. Det finns också några intressanta strategiska överväganden som varje gruvarbetare måste göra innan de väljer vilka block de ska arbeta på.

1. *vilka transaktioner som ska ingå.* Gruvarbetare får välja vilka transaktioner de ingå i en blockera. Standardstrategin är att inkludera alla transaktioner som inkluderar en transaktionsavgift högre än något minimum.
2. *Vilket block till gruvan på.* Gruvarbetare får också besluta ovanpå som blockerar de vill gruvan. Standardbeteendet för detta beslut är att förlänga den längsta kända giltiga kedjan.
3. *Välja mellan block vid samma höjd.* Om två olika block bryts och meddelade ungefär samtidigt resulterar det i en 1-blocksgaffel, där båda blocken är tillåtna under längst giltiga kedjepolicy. Gruvarbetare måste sedan bestämma vilket block som ska förlängas. Standarden beteende är att bygga ovanpå blocket som de hörde talas om först.
4. *När att tillkännage nya block.* När de hittar ett block, gruvarbetare måste bestämma när meddela detta till Bitcoin-nätverket. Standardbeteendet är att meddela det omedelbart, men de kan välja att vänta ett tag innan de tillkännager det.

Således ställs gruvarbetare inför många beslut. För varje beslut finns det en standardstrategi som används av Bitcoin-referensklienten, som drivs av de allra flesta gruvarbetare när detta skrivs. Det kan dock vara möjligt att en icke-standardstrategi är mer lönsam. Att hitta sådana scenarier och strategier är ett aktivt forskningsområde. Låt oss titta på flera sådana potentiellt lönsamma avvikelser från standardbeteende. I följande diskussion antar vi att det finns en icke-standardgruvarbetare som kontrollerar någon bråkdel av gruvkraft som vi betecknar med α.

Forking attack. Den enklaste attack är en forking attack och det självklara sättet att vinst för att utföra en dubbla utgifter. Gruvarbetaren skickar lite pengar till ett offer, Bob, som betalning för någon vara eller tjänst. Bob väntar och ser att transaktionen som betalar honom verkligen har inkluderats i blockkedjan. Kanske följer han den vanliga heuristiken och väntar till och med på sex bekräftelser för att vara säker. Övertygad

att han har fått betalt, skickar Bob varan eller utför tjänsten.

Gruvarbetaren går nu vidare och börjar arbeta på ett tidigare block — före blocket som innehåller transaktionen till Bob. I den här delade kedjan infogar gruvarbetaren en alternativ transaktion - eller en dubbel spend — som skickar mynten som betalats till Bob på huvudkedjan tillbaka till en av gruvarbetarnas egna adresser.

Figur 5.15 Forking attack. En illvillig gruvarbetare skickar en transaktion till Bob och tar emot några varor eller tjänst i utbyte mot det. Gruvarbetaren klaffar sedan blockkedjan för att skapa en längre gren som innehåller en motstridig transaktion. Betalningen till Bob kommer att vara ogiltig i denna nya konsensuskedja.

För att attacken ska lyckas måste den gaffelformade kedjan gå om den nuvarande längsta kedjan. När detta inträffar,
transaktionen som betalar Bob finns inte längre i konsensusblockkedjan. Detta kommer säkert att hända slutligen om den attackerande gruvarbetaren har en majoritet av hashkraften — det vill säga om $\alpha > 0{,}5$. Det vill säga till och med
även om det finns mycket slumpmässig variation i när block hittas, kedjan som växer snabbare på genomsnittet blir så småningom längre. Dessutom, eftersom gruvarbetarens mynt redan har förbrukats (på den nya konsensuskedjan) kan transaktionen som betalar Bob inte längre ta sig in på blocket kedja.

Är 51% nödvändigt? Starta en forking attack är förvisso möjligt om $\alpha > 0{,}5$. I praktiken kan det vara så möjligt att utföra denna attack med lite mindre än så på grund av andra faktorer som nätverk
över huvudet. Standardgruvarbetare som arbetar på huvudkedjan kommer att generera några inaktuella block för det vanliga
anledning: det finns en latens för gruvarbetare att höra om varandras block. Men en centraliserad angripare kan kommunicera mycket snabbare och producera färre inaktuella block, vilket kan innebära besparingar på 1 % eller mer.

Ändå, vid nära 50 % kan attacken ta lång tid att lyckas på grund av en slumpmässig slump. Attacken blir mycket enklare och effektivare ju längre du kommer över 50%. Folk pratar ofta om 51 % angripare som om 51% är en magisk tröskel som plötsligt möjliggör en forking attack. I verkligheten är det mer av en gradient.

Praktiska motåtgärder. Det är inte klart om en forking attack faktiskt skulle lyckas i praktiken. Attacken kan upptäckas och det är möjligt att samhället skulle besluta att blockera attacken vägrar att acceptera den alternativa kedjan trots att den är längre.

Attacker och växelkursen. Ännu viktigare är det troligt att en sådan attack skulle helt krascha Bitcoin-växelkursen. Om en gruvarbetare utförde en sådan attack, skulle förtroendet för systemet göra det

sjunka och växelkursen skulle falla när människor försöker flytta ut sina förmögenheter ur systemet. Således, medan en angripare med 51 % av hashkraften på kort sikt kan tjäna på dubbla utgifter, de kan allvarligt undergräva sin långsiktiga intjäningspotential för att bara bryta ärligt och tjäna pengar sina gruvbelöningar.

Av dessa skäl är kanske en mer rimlig motivation för en forking attack att specifikt förstöra valutan genom en dramatisk förlust av förtroende. Detta har kallas G *oldfinger attack* efter Bondskurken som försökte bestråla allt guld i Fort Knox för att göra det värdelöst. En Goldfinger angriparens mål kan vara att förstöra valutan, möjligen för att tjäna på antingen genom att kortsluta Bitcoin eller genom att ha betydande innehav i någon konkurrerande valuta.

Forking attack via mutor. Köpa tillräckligt hårdvara för att styra de flesta hash makt verkar vara en dyr och svår uppgift. Men det är möjligt att det finns ett enklare sätt att lansera en gaffelangrepp. Medan det skulle vara riktigt dyrt att direkt köpa tillräckligt med gruvkapacitet för att ha mer än alla andra i världen kan det vara möjligt att muta de människor som kontrollerar allt förmågan att arbeta för din räkning.

Det finns några sätt att muta gruvarbetare. Ett sätt är att göra det här "utanför bandet" - kanske hitta några stora gruvarbetare och ge dem ett kuvert med kontanter för att arbeta på din gaffel. En smartare Tekniken är att skapa en ny gruvpool och driva den med förlust, vilket ger större incitament än andra pooler. Även om incitamenten kanske inte är hållbara, kan en angripare hålla dem igång tillräckligt länge för att framgångsrikt starta en forking attack och kanske vinst. En tredje teknik är att lämna stora "spetsar" i block på gaffelkedjan - tillräckligt stora för att få gruvarbetare att lämna den längsta kedjan och jobba på gaffelkedjan i hopp om att den ska bli den längsta kedjan och de kan samla ihop spetsarna.

Oavsett mekaniken bakom mutan är idén densamma: istället för att faktiskt skaffa alla gruvkapacitet direkt, angriparen betalar bara de som redan har den för att hjälpa sin gaffel att övervinna den längsta kedjan.

Kanske vill gruvarbetare inte hjälpa till för att göra det skulle skada den valuta de har investerat så mycket pengar och gruvutrustning. Å andra sidan, medan gruvarbetare som grupp kanske vill behålla valutan solvent, agerar de inte kollektivt. Enskilda gruvarbetare kan hoppa av och acceptera en muta om de trodde att de kunde tjäna mer pengar på kort sikt. Det här skulle vara en klassiker allmänningens tragedi ur ett ekonomiskt perspektiv.

Inget av detta har faktiskt hänt och det är en öppen fråga om en mutattack som denna faktiskt skulle kunna göra det
vara livskraftig.
160

Tillfälliga blockundanhålla attacker. Säg att du bara hittat ett block. Standardbeteendet är att meddela det omedelbart för nätverket, men om du genomför en tillfällig blockeringsinnehållning attack, du meddelar det inte direkt. Istället försöker du ta dig framåt genom att göra lite mer brytning toppen av detta block i hopp om att hitta två block i rad innan resten av nätverket hittar ens ett, hålla dina block hemliga hela tiden.

Om du ligger före den offentliga blockkedjan med två hemliga block, kommer all gruvdrift från resten av

nätverket kommer att gå till spillo. Andra gruvarbetare kommer att bryta ovanpå vad de tror är den längsta kedjan, men som
så fort de hittar ett giltigt block kan du meddela de två blocken som du undanhöll. Den där
skulle omedelbart bli den nya längsta giltiga kedjan och blocket som resten av nätverket fungerade så
svårt att hitta skulle omedelbart bli föräldralös och avskuren från den längsta kedjan. Detta har kallats
självisk gruvdrift. Genom att bringa resten av nätverket till avfalls hash makt att försöka hitta ett block kan du omedelbart orsaka att bli inaktuell, hoppas du att öka din effektiva andel av gruvdrift belöningar.

Figur 5.16: Illustration av självisk gruvdrift. Detta visar ett av flera möjliga sätt på vilka attacken kunde spela ut. (1) Blockera kedjan före attack. (2) Angriparen bryter ett block, håller kvar det, börjar bryta vidare
toppen av det. (3) Angriparen har tur, hittar ett andra block före resten av nätverket, fortsätter att undanhålla block. (4) Icke-angripare hittar ett block och sänder det. Som svar sänder angriparen båda hans block, att göra det röda blocket föräldralöst och slösa bort gruvkraften som gick till att hitta det.

Haken är att du måste ha tur för att hitta två block i rad. Chansen är stor att någon annan kommer in nätverket meddelar ett giltigt block när du bara ligger ett kvarter före. Om detta händer, vill du tillkännage omedelbart ditt hemliga block själv. Detta skapar en 1-blocks gaffel och varje gruvarbetare kommer att göra det
måste fatta ett beslut om vilket av dessa block som ska brytas på. Din förhoppning är att en stor bråkdel av andra gruvarbetare kommer att höra om ditt block först och besluta sig för att arbeta med det. Livskraften av denna attack
beror mycket på din förmåga att vinna dessa lopp, så nätverkspositionen är avgörande. Du kan försöka peer med varje nod så att ditt block når de flesta noder först.

Som det visar sig, om du antar att du bara har 50 procents chans att vinna dessa lopp, självisk gruvdrift är en förbättring jämfört med standardstrategin om $\alpha > .25$. Även om du förlorar varje lopp, självisk

---

gruvdrift är fortfarande mer lönsamt om $\alpha > .333$. Förekomsten av denna attack är ganska överraskande och det är det
i motsats till den ursprungliga utbredda uppfattningen att utan en majoritet av nätverket - det vill säga med $\alpha \leq .5$, det fanns ingen bättre gruvstrategi än standarden. Så det är inte säkert att anta att en gruvarbetare som kontrollerar inte 50 procent av nätverket har inget att vinna på att byta till en alternativ
strategi.

Vid det här laget är tillfällig blockering bara en teoretisk attack och har inte observerats i öva. Självisk gruvdrift skulle ganska lätt att upptäcka eftersom det skulle öka hastigheten på nästan samtidiga blockmeddelanden.

Svartlistning och bestraffande forking. Säg en gruvarbetare vill svartlista transaktioner från adress X . I andra ord, de vill frysa pengarna som innehas av den adressen, vilket gör dem oanvändbara. Kanske du har för avsikt att dra nytta av detta genom någon form av lösensumma eller utpressningssystem som kräver att personen
du svartlistar betala dig för att tas bort från din svarta lista. Svartlistning kan också vara det

något som du är tvungen att göra av juridiska skäl. Kanske är vissa adresser betecknade som
ondska av regeringen. Brottsbekämpande myndigheter kan kräva att alla gruvarbetare som verkar i deras jurisdiktion
försök att svartlista dessa adresser.

Konventionell visdom är att det inte finns något effektivt sätt att svartlista adresser i Bitcoin. Även om några gruvarbetare vägrar att inkludera vissa transaktioner i block, andra gruvarbetare kommer att göra det. Om du är en gruvarbetare som försöker
svartlista, men du kan prova något starkare, nämligen punitive forking. Du skulle kunna meddela
att du kommer att vägra arbeta på en kedja som innehåller en transaktion som kommer från den här adressen. Om du har
en majoritet av hashkraften, bör detta vara tillräckligt för att garantera svartlistade transaktioner
aldrig publiceras. Faktum är att andra gruvarbetare förmodligen skulle sluta försöka, eftersom det helt enkelt skulle orsaka
deras block som ska glida i gafflar.

Feather-forking. Bestraffande forking verkar inte arbete utan en majoritet av nätverks hash
kraft. Genom att meddela att du kommer att vägra att bryta på någon kedja som har vissa transaktioner, om en sådan
kedjan uppstår och accepteras av resten av nätverket som den längsta kedjan, du
kommer att ha avskurit dig själv från konsensuskedjan för alltid (genom att effektivt införa en hård gaffel) och allt av gruvdriften som du gör kommer att gå till spillo. Ännu värre, de svartlistade transaktionerna kommer fortfarande att göras
den i den längsta kedjan.

Med andra ord, ett hot mot att svartlista vissa transaktioner via punitive forking på ovanstående sätt är inte trovärdigt när det gäller de andra gruvarbetarna. Men det finns ett mycket smartare sätt att göra det på. Istället för att meddela att du kommer att betala för alltid så fort du ser en transaktion komma
från adress X , meddela dig att du försöker gaffel om du ser ett block som har en transaktion
från adress X , men du kommer att ge upp efter ett tag. Till exempel kan du meddelade att efter k block bekräfta transaktionen från adress X, kommer du gå tillbaka till den längsta kedjan.

Om du ger upp efter en bekräftelse, din chans att föräldralösa blocket med transaktionen från X är α_2
. Anledningen till detta är att du måste hitta två på varandra följande block för att bli av med blocket

---

transaktionen från adress X innan resten av nätet finner ett block och α_2 är chansen att
du kommer att ha tur två gånger.

En chans av α_2 kanske inte verkar bra. Om du kontrollerar 20 % av hashkraften finns det bara 4 %
chans att faktiskt bli av med den transaktionen som du inte vill se i blockkedjan. Men det är
bättre än det kan verka eftersom du kan motivera andra gruvarbetare att gå med dig. Så länge du har varit väldigt allmänheten om dina planer, andra gruvarbetare vet att om de innehåller en transaktion från adress X , de ha en α_2 chansen att blocket som de hittar kommer att sluta elimineras på grund av din

fjädergaffelangrepp. Om de inte har någon stark motivation att inkludera den transaktionen från adress X och det inte har en hög transaktionsavgift, det α^2 chans att förlora sin gruvbelöning kan vara ett mycket större incitament än att samla in transaktionsavgiften.

Det visar sig då att andra gruvarbetare rationellt kan besluta sig för att gå med dig för att upprätthålla den svarta listan, och du kan därför genomdriva en svartlista även om $\alpha < .5$. Framgången för denna attack kommer att bero helt och hållet om hur övertygande du är för de andra gruvarbetarna att du definitivt kommer att gå på gaffel.

Övergång till gruv belöningar som domineras av transaktionsavgifter. Från och med 2015, transaktionsavgifter inte spelar så stor roll eftersom blockbelöningar ger den stora majoriteten – över 99 % – av alla intäkter som gruvarbetare gör. Men vart fjärde år är blockbelöningen planerad att halveras, och så småningom blockbelöningen kommer att vara tillräckligt låg för att transaktionsavgifterna kommer att vara den huvudsakliga inkomstkällan för gruvarbetare. Det är en öppen fråga exakt hur gruvarbetare kommer att fungera när transaktionsavgifterna blir deras huvudsakliga inkomstkälla. Kommer gruvarbetare att vara mer aggressiva när det gäller att genomdriva minimitransaktioner avgifter. Kommer de att samarbeta för att genomdriva det?

Öppna problem. Sammanfattningsvis gruvarbetare är fria att genomföra en strategi som de vill även i praxis har vi sett väldigt lite beteende av något annat än standardstrategin. Det finns ingen komplett modell för gruvarbeteende som säger att standardstrategin är optimal. I det här kapitlet har vi sett specifika exempel på avvikelser som kan vara lönsamma för gruvarbetare med tillräcklig hashkraft. Gruvstrategi kan vara ett område där praktiken ligger före teorin. Empiriskt har vi sett att i en värld där de flesta gruvarbetare väljer standardstrategin verkar Bitcoin fungera bra. Men vi är inte säkra på om det fungerar i teorin än.

Vi kan inte heller vara säkra på att det alltid kommer att fungera bra i praktiken. Fakta på plats är kommer att ändras för Bitcoin. Gruvarbetare blir mer centraliserade och mer professionella, och nätverkskapaciteten ökar. Dessutom, i det långa loppet Bitcoin måste kämpa med övergången från fasta gruvbelöningar till transaktionsavgifter. Vi vet inte riktigt hur detta kommer att fungera och hur det kommer att användas spelteoretiska modeller för att försöka förutsäga det är ett mycket intressant aktuellt forskningsområde.

Vidare läsning

En utmärkt artikel om utvecklingen av gruvhårdvara:

Taylor, Michael Bedford. Bitcoin och ålder skräddarsydda Silicon . Förfarandet 2013 Internationell konferens om kompilatorer, arkitekturer och syntes för inbyggda system. IEEE

Press, 2013.

Ett dokument som diskuterar några aspekter av att driva ett Bitcoin gruvcenter inklusive kylningskostnader:

Kampl, Alex. Analys av storskalig Bitcoin Gruvor . Vitbok, Allied Control, 2014.

Uppsatsen "systematisering av kunskap" om Bitcoin och kryptovalutor, särskilt avsnitt III om Stabilitet:

Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll och Edward W. Felten. Forskningsperspektiv och utmaningar för Bitcoin och Cryptocurrencies . Förfarandet 2015 IEEE Security and Privacy Conference, 2015.

Ett omfattande dokument från 2011 som analyserar olika belöningssystem för pooled gruvdrift (några av de informationen är lite inaktuell, men totalt sett är det fortfarande en bra resurs):

Rosenfeld, Meni. Analys av Bitcoin poolade gruvbelöningssystem . arXiv förtryck arXiv:1112.4980 (2011).

Flera artiklar som analyserar gruvstrategi:

. Eyal, Ittay och Emin Gün Sirer Majoriteten räcker inte: Bitcoin mining är sårbar . Finansiell Kryptografi och datasäkerhet. Springer Berlin Heidelberg, 2014.

Kroll, Joshua A., Ian C. Davey, och Edward W. Felten. Ekonomin i Bitcoin gruvdrift, eller Bitcoin i närvaron av motståndare . WEIS förfarande. Vol. 2013.

Eyal, Ittay. The Miner dilemma . Proceedings of 2015 IEEE Security and Privacy Conference, 2015.

Kapitel 6: Bitcoin och anonymitet

"Bitcoin är en säker och anonym digital valuta"

— WikiLeaks donationssida

"Bitcoin kommer inte att dölja dig från NSA:s nyfikna ögon"

— Wired Storbritannien

En av de mest kontroversiella sakerna med Bitcoin är dess förmodade anonymitet. För det första är Bitcoin anonym? Som du kan se från de ömsesidigt motstridiga citaten ovan, finns det viss förvirring om detta. För det andra, vi *vill ha* en kryptovaluta som verkligen är anonym? Det finns för- och nackdelar med anonymitet, vilket leder till några grundläggande frågor: är det fördelaktigt att ha en anonym kryptovaluta för intressenterna? Är det bra för samhället? Finns det ett sätt att isolera de positiva aspekterna av anonymitet samtidigt som man tar bort de negativa delarna?

Dessa frågor är svåra eftersom de delvis beror på ens etiska värderingar. Vi kommer inte att svara på dem i detta kapitel, även om vi kommer att undersöka argument för och emot anonymitet. För det mesta kommer vi att hålla oss till

studera olika teknologier — några finns redan i Bitcoin och andra som har föreslagits att läggas till det — som syftar till att öka Bitcoins anonymitet. Vi ska också titta på förslag till alternativa kryptovalutor som har andra anonymitetsegenskaper än Bitcoin. Dessa teknologier väcker nya frågor: Hur bra fungerar de? Hur svåra skulle de vara att adoptera? Vilka avvägningar ska göras för att anta dem?

6.1 Grundläggande anonymitet

Definiera anonymitet. Innan vi kan väl diskutera om (eller i vilken omfattning) Bitcoin är anonym, vi måste definiera anonymitet. Vi måste förstå exakt vad vi menar med anonymitet, och förhållandet mellan anonymitet och liknande termer, såsom integritet.

På en bokstavlig nivå betyder anonym "utan namn." När vi försöker tillämpa denna definition på Bitcoin, det finns två möjliga tolkningar: interagera utan att använda ditt riktiga namn, eller interagera utan att använda något namn alls. Dessa två tolkningar leder till mycket olika slutsatser beträffande om Bitcoin är anonymt. Bitcoin-adresser är hash av offentliga nycklar. Du behöver inte använda ditt riktiga namn för att interagera med systemet, men du använder din publika nyckelhash som din identitet. Således, vid den första tolkningen, är Bitcoin anonym eftersom du inte använder ditt riktiga namn. Men enligt den andra tolkningen är det inte det; adressen du använder är en pseudoidentitet. I den språket i datavetenskap, denna mellanväg för att använda en identitet som inte är ditt riktiga namn kallas ***pseudonymity.***

Kom ihåg att du är fri att skapa så många Bitcoin-adresser du vill. Med detta i åtanke kanske du undrar om Bitcoin-adresser verkligen är pseudo-identiteter med tanke på att du kan skapa så många av dessa pseudonymer du vill. Som vi kommer att se gör detta fortfarande inte Bitcoin anonym.

I datavetenskap hänvisar anonymitet till pseudonymity tillsammans med ***unlinkability*** . Unlinkability är en egendom som definieras med hänsyn till förmågan hos en specifik motståndare. Intuitivt, okopplingsbarhet betyder att om en användare interagerar med systemet upprepade gånger bör dessa olika interaktioner inte vara det kunna bindas till varandra ur motståndarens synvinkel.

Sidofält. Skillnaden mellan anonymitet och ren pseudonymitet är något som du kanske kan vara bekant med från en mängd andra sammanhang. Ett bra exempel är onlineforum. På ett forum som Reddit väljer du en långsiktig pseudonym och interagerar under en tidsperiod med den pseudonym. Du kan skapa flera pseudonymer, eller till och med en ny för varje kommentar, men det skulle vara tråkigt och irriterande och de flesta gör det inte. Så att interagera på Reddit är vanligtvis pseudonym men inte helt anonym. 4Chan, däremot, är ett onlineforum där användare postar i allmänhet anonymt - utan tillskrivning alls.

Bitcoin är pseudonym, men pseudonymitet räcker inte om ditt mål är att uppnå integritet. Återkallelse att blockkedjan är offentlig och vem som helst kan slå upp alla Bitcoin-transaktioner som involverade en given adress. Om någon någonsin kan länka din Bitcoin-adress till din verkliga identitet, då alla dina transaktioner – tidigare, nutid och framtida – kommer att ha kopplats tillbaka till din identitet.

För att göra saken värre är det ofta enkelt att länka en Bitcoin-adress till en verklig identitet. Om du interagerar med ett Bitcoin-företag - vare sig det är en onlineplånbokstjänst, börs eller annan handlare - det är de kommer vanligtvis att vilja ha din verkliga identitet för att låta dig handla med dem. Till exempel en utbyte kan kräva dina kreditkortsuppgifter, medan en handlare kommer att behöva din leveransadress. Eller så kanske du går till ett kafé och betalar ditt kaffe med bitcoins. Eftersom du är fysiskt närvarande i butiken vet baristan mycket om din identitet även om de inte frågar efter ditt riktiga namn. Din Den fysiska identiteten blir alltså knuten till en av dina Bitcoin-transaktioner, vilket gör alla andra transaktioner som innebar den adressen som är länkbar till dig. Detta är uppenbarligen inte anonymt.

Sidokanalerna. Även om en direkt koppling inte sker kan din pseudonym profilen ***deanonymized*** grund av sidokanaler, eller indirekta läckage av information. Till exempel kan någon

titta på en profil av pseudonyma Bitcoin-transaktioner och notera vid vilka tider på dygnet den användaren är aktiva. De kan korrelera denna information med annan allmänt tillgänglig information. Det kanske de gör Observera att vissa Twitter-användare är aktiva under ungefär samma tidsintervall, vilket skapar en länk mellan pseudonym Bitcoin-profil och en verklig identitet (eller åtminstone en Twitter-identitet). Klart Pseudonymitet garanterar inte sekretess eller anonymitet. För att uppnå dessa kräver vi de starkare egendomen att koppla bort också.
166

Sida 68

Unlinkability. För att förstå unlinkability i Bitcoin sammanhang mer konkret, låt oss enumerate några nyckelegenskaper som krävs för att Bitcoin-aktivitet ska kunna kopplas bort:
1. Det borde vara svårt att länka samman olika adresser för samma användare.
2. Det borde vara svårt att länka samman olika transaktioner gjorda av samma användare.
3. Det borde vara svårt att koppla avsändaren av en betalning till dess mottagare.
De två första egenskaperna är intuitiva, men den tredje är lite knepig. Om du tolkar "en betalning" som en Bitcoin transaktion, då är den tredje egenskapen uppenbart falsk. Varje transaktion har ingångar och utgångar, och dessa input och output kommer oundvikligen att vara i blockkedjan och offentligt länkade tillsammans. Men vad vi menar med en betalning är inte en enda Bitcoin-transaktion, utan snarare allt som har effekten att överföra bitcoins från avsändaren till mottagaren. Det kan innebära en rondell serie av transaktioner. Det vi vill säkerställa är att det inte är möjligt att länka avsändaren och den slutliga mottagaren av betalningen genom att titta på blockkedjan.
Anonymitet set. Även under vår bredare definition av en betalning, verkar svårt att den tredje egenskapen uppnå. Säg att du betalar för en produkt som kostar ett visst antal bitcoins och du skickar den betalningen genom en kringgående transaktionsväg. Någon som tittar på blockkedjan kommer fortfarande att kunna härleda något av det faktum att ett visst antal bitcoins lämnade en adress och ungefär samma antal bitcoins (minus transaktionsavgifter, kanske) hamnade på någon annan adress. Dessutom, trots den omväxlande vägen kommer den första sändningen och den slutliga mottagningen att ske i ungefär samma tidsperiod eftersom handlaren kommer att vilja ta emot betalning utan alltför mycket fördröjning. På grund av denna svårighet försöker vi vanligtvis inte uppnå fullständig okopplingsbarhet mellan alla möjliga transaktioner eller adresser i systemet, utan snarare något mer begränsat. Med tanke på en viss fiende, den *anonymitet uppsättningen* är din transaktion uppsättningen transaktioner som motståndaren inte kan skilja från din transaktion. Även om motståndaren vet att du gjort en transaktion kan de bara berätta att det är en av transaktionerna i setet, men inte vilken det är. Vi försöker maximera storleken på anonymitetsuppsättningen — uppsättningen av andra adresser eller transaktioner som vi kan gömma oss bland. Att beräkna anonymitetsuppsättningen är knepigt. Eftersom anonymitetsuppsättningen definieras med avseende på en viss
motståndare eller uppsättning motståndare måste du först konkret definiera vad din motståndarmodell är. Du måste resonera noggrant om vad den motståndaren vet, vad de inte vet och vad är det vi försöker dölja från motståndaren - det vill säga vad motståndaren *inte kan* känna till transaktionen att betraktas som anonym. Det finns ingen generell formel för att göra detta. Det kräver noggrant analysera varje protokoll och system från fall till fall.
Lukt analys. I Bitcoin samhället, människor ofta utföra intuitiva analyser av anonymitet tjänster utan rigorösa definitioner. *Lukt analys* är särskilt populärt: det är ett sätt att beräkna hur "relaterade" två adresser är. Om bitcoins som skickas av en adress S alltid hamnar på en annan adress R, vare sig direkt eller efter att ha passerat några mellanliggande adresser, så kommer S och R att ha en hög fläcka poäng. Formeln tar hänsyn till transaktioner med flera ingångar och/eller utgångar och specificerar hur man fördelar fläck.
167

Tyvärr är fläckanalys inte ett bra mått på Bitcoin-anonymitet. Det förutsätter implicit att motståndaren använder samma mekaniska beräkning för att länka par av adresser. En lite smartare motståndare kan använda andra tekniker som att titta på tidpunkten för transaktioner eller till och med utnyttja egenheter i plånboksprogramvara som vi kommer att se senare i det här kapitlet. Så fläckanalys kan tyda på det du har en hög grad av anonymitet i en viss situation, men i själva verket kanske du inte.

Därför behöver vi anonymitet. Efter att ha sett vad anonymitet medel, låt oss svara på några meta frågor om anonymitet innan vi går vidare: Varför vill folk ha anonymitet? Vad är de etiska konsekvenser av att ha en anonym valuta?

I blockkedjebaserade valutor registreras alla transaktioner på reskontran, vilket innebär att de är offentligt och permanent spårbara till tillhörande adresser. Så integriteten för din Bitcoin transaktioner kan potentiellt vara mycket värre än med traditionell bankverksamhet. Om din verkliga identitet någonsin blir länkad till en Bitcoin-adress, då har du helt förlorat integriteten för alla transaktioner - tidigare, nu, och framtid — förknippad med den adressen. Eftersom blockkedjan är allmänt tillgänglig, bokstavligen vem som helst kanske kan utföra den här typen av deanonymisering utan att du ens inser att du har gjort det identifieras.

Med detta i åtanke kan vi identifiera två olika motiv för att ha anonyma kryptovalutor.

Den första är helt enkelt att uppnå den nivå av integritet som vi redan är vana vid från traditionell bankverksamhet, och minska risken för deanonymisering som den offentliga blockkedjan medför. Det andra är att gå över och utöver integritetsnivån för traditionell bankverksamhet och utveckla valutor som gör det tekniskt omöjligt för någon att spåra deltagarna.

Etik av anonymitet. Det finns många viktiga (men ofta förbises) skäl för anonymitet som vi tar för givet med traditionella valutor. De flesta människor är obekväma att dela med sig av sina löner med sina vänner och arbetskamrater. Om individens adresser i blockkedjan är lätta att identifiera men och de får sin lön i Bitcoin, skulle det vara ganska lätt att sluta sig till deras lön genom att titta för en stor, regelbunden månadsbetalning. Organisationer har också viktiga ekonomiska integritetsproblem. Till exempel, om en tillverkare av videospelskonsoler skulle observeras i blockkedjan som betalade en underleverantör som tillverkar virtual reality-glasögon, kan detta tipsa allmänheten (och deras konkurrenter) om en ny produkt som de förbereder att lansera.

Det finns dock en berättigad oro för att verkligt anonyma kryptovalutor kan användas för pengar tvättning eller annan olaglig verksamhet. Den goda nyheten är att medan kryptovaluta transaktioner själva kan vara pseudonyma eller anonyma, gränssnittet mellan digitala kontanter och fiat valutor är det inte. Faktum är att dessa flöden är mycket reglerade, som vi kommer att se i nästa kapitel. Så kryptovalutor är inget universalmedel mot penningtvätt eller andra ekonomiska brott.

Ändå kan man fråga sig: kan vi inte designa tekniken på ett sådant sätt som bara det goda använder är anonymitet tillåtet och de dåliga användningarna är på något sätt förbjudna? Detta är faktiskt en återkommande vädjan till datasäkerhets- och integritetsforskare. Tyvärr visar det sig aldrig vara möjligt. De

---

Anledningen är att användningsfall som vi klassificerar som bra eller dåliga ur moralisk synvinkel visar sig vara tekniskt identiska. I Bitcoin är det inte klart hur vi skulle kunna ge gruvarbetare i uppdrag att göra moral beslut om vilka transaktioner som ska inkluderas.

Vår uppfattning är att den potentiella nyttan som möjliggörs genom att ha anonyma kryptovalutor motiverar deras existens, och att vi bör skilja systemets tekniska anonymitetsegenskaper från de juridiska principer vi tillämpar när det gäller att använda valutan. Det är inte helt tillfredsställande lösning, men det är kanske det bästa sättet att uppnå en fördelaktig avvägning.

Sidebar. Tor det moraliska dilemmat om hur man handskas med en teknik som har både goda och dåliga användningsområden
är inte på något sätt unikt för Bitcoin. Ett annat system vars anonymitet är kontroversiell är Tor, an anonymt kommunikationsnätverk.
Å ena sidan används Tor av normala människor som vill skydda sig från att bli spårade uppkopplad. Det används av journalister, aktivister och oliktänkande för att tala fritt online utan rädsla för återverkningar från förtryckande regimer. Det används också av brottsbekämpande agenter som vill övervaka misstänkta online utan att avslöja deras IP-adress (trots allt intervall eller block av IP adresser som tilldelats olika organisationer, inklusive brottsbekämpande myndigheter, tenderar att vara bra känd). Det är uppenbart att Tor har många applikationer som vi moraliskt sett kan godkänna. Det har det dock också
helt klart dåliga användningsområden: det används av operatörer av botnät för att utfärda kommandon till de infekterade maskinerna
under deras kontroll och det används för att distribuera bilder på sexuella övergrepp mot barn.
Att skilja mellan dessa användningar på teknisk nivå är i princip omöjligt. Tor-utvecklarna och Tor-gemenskapen har brottats mycket med denna gåta. Samhället i stort har brottats med det till viss del också. Vi verkar ha kommit fram till att det överlag är bättre för världen att tekniken finns. Faktum är att en av de viktigaste finansieringskällorna för Tor-projektet är USA:s utrikesdepartement. De är intresserade av Tor eftersom det möjliggör fritt tal online för oliktänkande i förtryckande regimer. Samtidigt verkar brottsbekämpande myndigheter ha motvilligt accepterat Tors existens och har utvecklat sätt att kringgå den. FBI har regelbundet
lyckades spränga hemsidor på det "mörka nätet" som till och med distribuerade bilder av sexuella övergrepp mot barn
även om dessa webbplatser gömde sig bakom Tor. Ofta beror det på att operatörerna snubblade. Vi måste kom ihåg att teknik bara är ett verktyg och att förövare av brott lever i den verkliga världen, där de kan lämna fysiska bevis eller begå alltför mänskliga fel när de interagerar med teknologi.
Anonymisering vs decentralisering. Vi får se ett återkommande tema under hela det här kapitlet att designkriterier för anonymisering och decentralisering är ofta i konflikt med varandra. Om du minns Chaums ecash från förordet, uppnådde den perfekt anonymitet på sätt och vis, men genom en interaktivt blindsignaturprotokoll med en central myndighet, en bank. Som ni kan föreställa er, sådana protokoll är mycket svåra att decentralisera. För det andra, om vi decentraliserar, måste vi behålla något slags

---

mekanism för att spåra transaktioner och förhindra dubbla utgifter. Denna offentliga spårbarhet av transaktioner är ett hot mot anonymiteten.
Senare i det här kapitlet kommer vi att se Zerocoin och Zerocash, anonyma decentraliserade kryptovalutor som har vissa likheter med Chaums pengar, men de måste ta sig an svåra kryptografiska utmaningar på grund av dessa två begränsningar.

6.2 Hur man avanonymiserar Bitcoin

Vi har sagt flera gånger att Bitcoin bara är pseudonym, så alla dina transaktioner eller adresser skulle kunna kopplas samman. Låt oss ta en närmare titt på hur det faktiskt kan hända.
Figur 6.1 visar ett utdrag av Wikileaks donationssida (inklusive citatet i början av kapitel). Lägg märke till uppdateringsknappen bredvid donationsadressen. Som du kanske förväntar dig, klicka på knappen kommer att ersätta donationsadressen med en helt ny, nyskapad adress. På samma sätt, om om du uppdaterar sidan eller stänger den och besöker den senare, kommer den att ha en annan adress, aldrig tidigare sett.
Det beror på att Wikileaks vill se till att varje donation de får går till en ny publik

nyckel som de skapar just för det ändamålet. Wikileaks drar maximal nytta av förmågan att skapa nya pseudonymer. Detta är faktiskt den bästa praxis för anonymitet som används av Bitcoin-plånböcker.
Figur 6.1. Fragment från Wikileaks donation sida Meddelande uppdaterings ikonen bredvid Bitcoin adress. Wikileaks följer Bitcoins bästa praxis att skapa en ny mottagningsadress för varje donation.
Till en början kanske du tror att dessa olika adresser måste kunna kopplas bort. Wikileaks tar emot var och en donation separat, och förmodligen kan de också spendera var och en av dessa donationer separat. Men saker går snabbt sönder.
Länkning. Antag Alice vill köpa en tekanna som kostar 8 Bitcoins (mer troligt 8 centi-Bitcoins på 2015 växlingskurs). Antag vidare att hennes bitcoins finns i tre separata outnyttjade utgångar vid olika adresser vars belopp är 3, 5 respektive 6 bitcoins. Alice har faktiskt ingen adress med 8 bitcoins i den, så hon måste kombinera två av sina utgångar som indata till en enda transaktion att hon betalar till butiken.

Sida 72

Men detta avslöjar något. Transaktionen registreras permanent i blockkedjan, och vem som helst vem som ser det kan dra slutsatsen att de två ingångarna till transaktionen sannolikt är under kontroll av samma användare. Med andra ord **delas utgifterna bevis för gemensam kontroll** av de olika ingångs adresser. Det kan naturligtvis finnas undantag. Kanske är Alice och Bob rumskamrater och går med på det köper gemensamt tekannan genom att var och en tillhandahålla en transaktionsingång. Men i stort sett gemensamma insatser innebär gemensam kontroll.
Figur 6,2: Till lön för tekannan, har Alice för att skapa en enda transaktion har ingångar som är i två annan adress. Genom att göra det avslöjar Alice att dessa två adresser kontrolleras av en enda enhet.
Men det stannar inte där. Motståndaren kan upprepa denna process och **transitivt** länka en hel klunga transaktioner som tillhör en enda enhet. Om en annan adress är länkad till **antingen** en av Alices adresser på detta sätt, då vet vi att alla tre adresser tillhör samma enhet, och vi kan använda denna observation för att gruppera adresser. I allmänhet, om en utdata på en ny adress förbrukas tillsammans med en från någon av adresserna i klustret, så kan även denna nya adress läggas till till klustret.
Senare i det här kapitlet kommer vi att studera en anonymitetsteknik som heter CoinJoin som fungerar genom att bryta mot detta antagande. Men för nu, om du antar att folk använder vanlig Bitcoin-plånbok utan någon speciell anonymitetsteknik, då tenderar denna klustring att vara ganska robust. Vi har ännu inte sett hur man länkar dessa kluster till verkliga identiteter, men vi kommer till det snart.
Sidebar. Ändra adress randomisering En tidig version av Bitcoin-qt bibliotek hade en bugg som lägg alltid ändringsadressen som den första utgången i en transaktion med två utgångar. Detta betydde det det var trivialt att identifiera ändringsadressen i många transaktioner. Denna bugg fixades 2012 men lyfter fram en viktig punkt: plånboksmjukvara har en viktig roll att spela för att skydda anonymitet. Om du utvecklar plånboksmjukvara finns det många fallgropar du bör vara medveten om; i särskilt bör du alltid välja positionen för ändringsadressen slumpmässigt för att undvika att ge för mycket bort till motståndaren!

Sida 73

Om vi går tillbaka till vårt exempel, anta att priset på tekannan har gått upp från 8 bitcoins till 8,5 bitcoins. Alice kan inte längre hitta en uppsättning outnyttjade utdata som hon kan kombinera för att producera exakt

byte som behövs för tekannan. Istället utnyttjar Alice det faktum att transaktioner kan ha flera utgångar, som visas i figur 6.3. En av utgångarna är butikens betalningsadress och den andra är en "ändra" adress som ägs av henne själv.
Betrakta nu denna transaktion från en motståndares synvinkel. De kan dra slutsatsen att de två inmatningsadresser tillhör samma användare. De kan ytterligare misstänka att en av utdataadresserna tillhör också samma användare, men har inget sätt att veta säkert vilken det är. Det faktum att 0,5 utgång är mindre betyder inte att det är ändringsadressen. Alice kanske har 10 000 bitcoins sitter i en transaktion, och hon kanske spenderar 8,5 bitcoins på tekannan och skickar resten 9 991,5 bitcoins tillbaka till sig själv. I det scenariot är den större effekten i själva verket ändringsadressen.
Figur 6.3. Ändra adress För att betala för tekannan, har Alice att skapa en transaktion med en utgång som går till handlaren och en annan utgång som skickar växel tillbaka till henne själv.
En något bättre gissning är att om tekannan bara hade kostat 0,5 bitcoin, så hade Alice inte haft att skapa en transaktion med två olika ingångar, eftersom antingen 3 bitcoin-ingången eller 6 bitcoin-ingången skulle ha varit tillräckligt i sig. Men effektiviteten av denna typ av heuristik beror helt på implementeringsdetaljerna för vanliga plånboksprogram. Det finns inget som hindrar plånböcker (eller användare) från att kombinera transaktioner även när det inte är absolut nödvändigt.
Idiom användnings. Genomförande detaljer av detta slag kallas ”idiom användning”. Under 2013, en grupp av forskare hittade ett uttryck för användning som var sant för de flesta plånboksprogram, och ledde till en kraftfull heuristisk för att identifiera ändringsadresser. Specifikt fann de att plånböcker vanligtvis genererar en ny adress närhelst en ändring av adress krävs. Byt adresser på grund av denna användningsform är i allmänhet adresser som aldrig tidigare förekommit i blockkedjan. Ej ändrade utgångar, på

å andra sidan, är ofta inte nya adresser och kan ha dykt upp tidigare i blockkedjan. En motståndare kan använda denna kunskap för att särskilja ändringsadresser och koppla dem till indata adresser.
Att utnyttja idiom för användning kan vara felbenäget. Det faktum att byta adresser är färska adresser bara råkar vara en funktion i plånboksprogramvara. Det var sant 2013 när forskarna testade det. Kanske det är fortfarande sant, men det kanske inte är det. Användare kan välja att åsidosätta detta standardbeteende. Mest viktigare är att en motståndare som är medveten om denna teknik lätt kan undvika den. Även 2013, den forskare fann att det producerade många falska positiva resultat, där de hamnade i kluster adresser som faktiskt inte tillhörde samma enhet. De rapporterade att de behövde betydande manuell tillsyn och ingripande för att beskära de falska positiva.
Figur 6,4:. Kluster av adresser I 2013 papper *För en handfull Bitcoins: kännetecknande Betalningar Bland Män med inga namn,* forskare kombinerade delad spendera heuristiska och fresh-change-address heuristik för att klustera Bitcoin-adresser. Storleken på dessa cirklar representerar mängd pengar som flödar in i dessa kluster, och varje kant representerar en transaktion.
Fästa verkliga identiteter till kluster. I figur 6.4. vi ser hur Meiklejohn et al. klustrade Bitcoin-adresser använder grundläggande idiom för användning som heuristik. Men grafen är inte märkt - det har vi inte

---

ändå kopplade identiteter till klustren.
Vi kanske kan göra några kvalificerade gissningar baserat på vad vi vet om Bitcoin ekonomi. Tillbaka 2013 var Mt. Gox den största Bitcoin-börsen, så vi kan gissa att den är den stora lila cirkel representerar adresser som kontrolleras av dem. Vi kanske också märker att den bruna klungan på vänstern har en liten volym i Bitcoins trots att de har det största antalet transaktioner. Detta passar till

mönster av speltjänsten Satoshi Dice, som är ett populärt spel som du skickar en liten summa till bitcoins som satsning. Sammantaget är detta dock inte ett bra sätt att identifiera kluster. Det kräver kunskap och gissningar och kommer bara att fungera för de mest framstående tjänsterna.

Taggning av transagerande. Hur bara besöker webbplatsen för varje utbyte eller handlaren och leta upp adressen de annonserar för att ta emot bitcoins? Det fungerar dock inte riktigt, eftersom de flesta tjänster kommer att annonsera en ny adress för varje transaktion och adressen som visas till du är ännu inte i blockkedjan. Det är ingen idé att vänta heller, för den adressen kommer aldrig att bli det visat för någon annan.

Det enda sättet att på ett tillförlitligt sätt härleda adresser är att faktiskt göra transaktioner med den tjänsteleverantören - insättning

bitcoins, köpa en vara och så vidare. När du skickar bitcoins till eller tar emot bitcoins från tjänsteleverantör kommer du nu att känna till en av deras adresser, som snart hamnar i blockkedjan (och i ett av klustren). Du kan sedan tagga hela det klustret med tjänsteleverantörens identitet.

Detta är exakt vad *Fistful of Bitcoins* forskare (och andra sedan) har gjort. De köpte en olika saker, gick med i gruvpooler, använde Bitcoin-börser, plånbokstjänster och spelsajter, och interagerade på en mängd andra sätt med tjänsteleverantörer, vilket kompromitterade totalt 344 transaktioner. I figur 6.5 visar vi återigen klustren i figur 6.4, men denna gång med etiketterna fästa. Medan våra gissningar om Mount Gox och Satoshi Dice var korrekta, kunde forskarna identifiera många andra tjänsteleverantörer som skulle ha varit svåra att identifiera utan att ha gjort transaktioner med dem.

174

Sida 76

Figur 6,5 . ***Märkt kluster.*** Genom transaktioner med olika Bitcoin tjänsteleverantörer, Meiklejohn et al. kunde fästa verkliga identiteter till sina kluster.

. *Identifiera individer* Nästa fråga är: kan vi göra samma sak för individer? Det vill säga kan vi koppla ihop små kluster som motsvarar individer till deras verkliga identiteter?

. *Direkt transaktions* Den som gör transaktioner med en individuell - en online eller offline köpman, en utbyte, eller en vän som delar en middagsräkning med Bitcoin — vet minst en adress som tillhör dem.

Via tjänsteleverantörer. Under använda Bitcoin under några månader eller år, kommer de flesta användare att sluta interagerar med en börs eller annan centraliserad tjänsteleverantör. Dessa tjänsteleverantörer vanligtvis fråga användarna om deras identiteter – ofta är de juridiskt skyldiga att göra det, som vi kommer att se i nästa kapitel. Om lag

verkställighet vill identifiera en användare kan de vända sig till dessa tjänsteleverantörer.

Slarv. Människor lägger ofta sina Bitcoin adresser i offentliga forum. En vanlig anledning är att begära donationer. När någon gör detta skapar det en länk mellan deras identitet och en av deras adresser. Om de inte använder anonymitetstjänsterna som vi kommer att titta på i följande avsnitt, de riskerar att få alla sina transaktioner avanonymiserade.

175

Sida 77

Saker och ting blir värre med tiden. Historien visar att deanonymization algoritmer förbättras vanligtvis över tid då data är allmänt tillgänglig när fler forskare studerar problemet och identifierar nya attacktekniker. Dessutom blir mer hjälpinformation tillgänglig som angripare kan använda sig av fästa identiteter till kluster. Detta är något att oroa sig för om du bryr dig om integritet.

Deanonymiseringsteknikerna vi har undersökt hittills är alla baserade på att analysera graferna för transaktioner i blockkedjan. De är kollektivt kallas ***transaktionsgrafanalys.***

Nätverkslager deanonymization. Det finns ett helt annat sätt där användarna kan få

deanonymiserad som inte förlitar sig på transaktionsdiagrammet. Kom ihåg att för att bokföra en transaktion till blockkedjan sänder man vanligtvis den till Bitcoins peer-to-peer-nätverk där meddelanden finns skickas runt som inte nödvändigtvis registreras permanent i blockkedjan.
I nätverk terminologi är blocket kedjan kallas ***applikationslagret*** och peer-to-peer nätverket är ***nätverkslagret*** . Deanonymisering av nätverkslager påpekades först av Dan Kaminsky vid 2011 års Black Hat-konferens. Han märkte att när en nod skapar en transaktion ansluter den till många noder samtidigt och sänder transaktionen. Om tillräckligt många noder på nätverket samarbetar med varandra (eller drivs av samma motståndare) kan de komma på den första noden som ska sändas någon transaktion. Förmodligen skulle det vara en nod som körs av användaren som skapade transaktion. Motståndaren kunde sedan länka transaktionen till nodens IP-adress. En IP-adress är nära en verklig identitet; det finns många sätt att försöka avslöja personen bakom en IP-adress. Således är avanonymisering av nätverkslager ett allvarligt problem för integriteten.
Figur 6.6 . ***Network nivå deanonymization.*** Som Dan Kaminsky påpekade i sitt 2011 Black Hat prata, "den första noden som informerar dig om en transaktion är förmodligen källan till den." Denna heuristik förstärks när flera noder samarbetar och identifierar samma källa.
176

Sida 78

Lyckligtvis är detta ett problem med kommunikationsanonymitet, som redan har varit föremål för betydande forskning. Som vi såg tidigare i avsnitt 6.1, finns det ett brett utrullat system som heter Tor som du kan använda för att kommunicera anonymt.
Det finns ett par varningar för att använda Tor som en anonymitetslösning för nätverkslager för Bitcoin. Först, det kan finnas subtila interaktioner mellan Tor-protokollet och vilket protokoll som helst som läggs ovanpå det, vilket resulterar i nya sätt att bryta anonymiteten. Ja, forskare har hittat potentiell säkerhet problem med att använda Bitcoin-over-Tor, så detta måste göras med extrem försiktighet. För det andra, där kan vara annan anonym kommunikationsteknik som är bättre lämpad att använda i Bitcoin. Tor är avsedd för aktiviteter med "låg latens" som att surfa på webben där du inte vill sitta runt väntar för länge. Det gör vissa kompromisser för att uppnå anonymitet med låg latens. Bitcoin, av jämförelse, är ett system med hög latens eftersom det tar ett tag för transaktioner att bli bekräftade i blockkedja. I teorin kanske du åtminstone vill använda en alternativ anonymitetsteknik som en **mix net** , men för tillfället har Tor fördelen av att vara ett faktiskt system som har en stor användare bas och vars säkerhet har studerats intensivt.
Hittills har vi sett att olika adresser kan länkas samman genom analys av transaktionsdiagram och att de också kan kopplas till en verklig identitet. Vi har också sett att en transaktion eller adress kan kopplas till en IP-adress baserad på peer-to-peer-nätverket. Det senare problemet är relativt lätt att lösa, även om det inte kan anses helt löst än. Det förra problemet är mycket svårare, och vi kommer att ägna resten av det här kapitlet till att prata om sätt att lösa det.

6.3 Blandning

Det finns flera mekanismer som kan göra analys av transaktionsdiagram mindre effektiv. En sådan Tekniken är *att blanda* och intuition bakom det är mycket enkelt: om du vill vara anonym, använda en mellanhand. Denna princip är inte specifik för Bitcoin och är användbar i många situationer där anonymitet är ett mål. Blandning illustreras i figur 6.7.
177

Sida 79

Figur 6.7. Blandning Användare skickar mynt till en mellanhand och få tillbaka mynt som deponerats av andra användare. Detta gör det svårare att spåra en användares mynt på blockkedjan.
Online plånböcker som mixar. Om ni minns vår diskussion online plånböcker, kan de verkar vara lämpliga som förmedlarna. Onlineplånböcker är tjänster där du kan lagra dina bitcoins online och ta ut pengar

dem vid något senare tillfälle. Vanligtvis kommer mynten du tar ut inte att vara desamma som mynten du deponeras. Ger onlineplånböcker effektiv blandning då?
Online-plånböcker ger ett mått på olänkning som kan förhindra försök till transaktionsdiagram analys — i ett fall var framstående forskare tvungna att dra tillbaka ett påstående som hade fått mycket av publicitet eftersom länken de trodde att de hade hittat var en falsk länk orsakad av en onlineplånbok.
Å andra sidan finns det flera viktiga gränser för att använda onlineplånböcker för att blanda. Först, de flesta onlineplånböcker lovar faktiskt inte att blanda användarnas pengar; istället gör de det för att det förenklar teknik. Du har ingen garanti för att de inte kommer att ändra sitt beteende. För det andra, även om de gör det blanda medel, kommer de nästan säkert att hålla register internt som gör att de kan länka din insättning till ditt uttag. Detta är ett klokt val för plånbokstjänster av både säkerhetsskäl och laglig efterlevnad. Så om din hotmodell inkluderar möjligheten för tjänsteleverantören själv genom att spåra dig, bli hackad, eller tvingas lämna över sina register, är du tillbaka till rutten ett. För det tredje, förutom att föra loggar internt, kommer ansedda och reglerade tjänster också att kräva och registrera din identitet (vi kommer att diskutera reglering mer i detalj i nästa kapitel). Det kommer du inte att vara
kan helt enkelt skapa ett konto med ett användarnamn och lösenord. Så i en mening gör det dig värre av än att inte använda plånbokstjänsten. Det var därför vi ropade på spänningen mellan centralisering och anonymitet i föregående avsnitt.
Anonymiteten som tillhandahålls av onlineplånböcker liknar den som tillhandahålls av traditionell bank systemet. Det finns centraliserade mellanhänder som vet mycket om våra transaktioner, men från en främlings synvinkel utan privilegierad information har vi en rimlig grad av integritet. Men som vi diskuterade betyder blockkedjans offentliga karaktär att om något går fel (säg, a plånbok eller växlingstjänst blir hackad och register exponeras), är integritetsrisken värre än med det traditionella systemet. Dessutom tenderar de flesta som vänder sig till Bitcoin för anonymitet att göra det pga

---

de är missnöjda med det traditionella systemets anonymitetsegenskaper och vill ha en bättre (eller en annan typ av) anonymitetsgaranti. Dessa är motiven bakom dedikerade blandningstjänster.
Dedikerad blandnings tjänster. Till skillnad från online-plånböcker, särskilda blandningar lovar att inte föra register,
De kräver inte heller din identitet. Du behöver inte ens ett användarnamn eller annan pseudonym för att interagera med blandningen. Du skickar dina bitcoins till en adress som tillhandahålls av mixen, och du berättar för mixen a destinationsadress att skicka bitcoins till. Förhoppningsvis kommer mixen snart att skicka dig (andra) bitcoins kl adress du angett. Det är i princip ett byte.
Även om det är bra att dedikerade mixar lovar att inte hålla rekord, måste du fortfarande lita på att de håller det löftet. Och du måste lita på att de skickar tillbaka dina mynt överhuvudtaget. Eftersom blandningar inte är en plats där du lagrar dina bitcoins, till skillnad från plånböcker, vill du ha tillbaka dina mynt relativt snabbt, vilket innebär att poolen av andra mynt som din insättning kommer att blandas med är mycket mindre — de som deponerades ungefär samtidigt.
Sidebar. Terminologi I denna bok kommer vi att använda termen ***mix*** för att hänvisa till en särskild blandning tjänst. En
likvärdig term som en del människor föredrar är ***mixer*** .
Du kan också stöta på termen ***tvätten*** . Vi gillar inte den här termen, för den fäster i onödan en moralisk bedömning av något som är ett rent tekniskt koncept. Som vi har sett finns det mycket goda skäl till varför du kanske vill skydda din integritet i Bitcoin och använda mixar till vardags Integritet. Självklart måste vi också erkänna de dåliga användningarna, men att använda termen tvätt främjar den negativa klangen, eftersom det antyder att dina mynt är "smutsiga" och att du måste rengöra dem.
Det finns också termen ***tumlaren*** . Det är inte klart om detta syftar på blandningsverkan av trumlande trummor eller

deras rengöringseffekt (på ädelstenar och sådant). Oavsett vilket kommer vi att hålla oss till termen "mix".
En grupp forskare, inklusive fyra av de fem författarna till denna lärobok, studerade blandningar och föreslog en uppsättning principer för att förbättra hur blandningar fungerar, både när det gäller att öka anonymiteten och när det gäller säkerheten att anförtro dina mynt till mixen. Vi kommer att gå igenom var och en av dessa riktlinjer.
Använd en serie av blandningar. Den första principen är att använda en serie av blandningar, den ena efter den andra, istället för att bara
en enda blandning. Detta är en välkänd och väletablerad princip — till exempel Tor, som vi kommer att se i en bit, använder en serie av 3 routrar för anonym kommunikation. Detta minskar ditt beroende av trovärdigheten för varje enskild blandning. Så länge någon av mixarna i serien håller vad de lovar och raderar dess register, har du anledning att förvänta dig att ingen kommer att kunna koppla din första ingång till ultimata resultat som du får.

Figur 6.8. Serie av blandningar . Vi börjar med en användare som har ett mynt eller inmatningsadress som vi antar
motståndare har lyckats länka till dem. Användaren skickar myntet genom olika mixar, varje gång tillhandahålla en nygenererad utdataadress till mixen. Förutsatt att minst en av dessa blandar förstör sina register över ingång till utgångsadressmapping, och det finns inga sidokanalläckor av information, kommer en motståndare inte att kunna länka användarens ursprungliga mynt till deras sista.
Enhetliga transaktioner. If mix transaktioner olika användare hade olika mängder av Bitcoins, sedan blandning skulle inte vara särskilt effektivt. Eftersom värdet går in i mixen och kommer ut ur en mix skulle måste bevaras, kommer det att göra det möjligt att länka en användares mynt när de flödar genom mixen, eller åtminstone
avsevärt minska storleken på anonymitetsuppsättningen.
Istället vill vi att mixtransaktioner ska vara enhetliga i värde så att länkbarheten minimeras. Alla blandningar bör komma överens om en standard ***bit storlek*** , ett fast värde att inkommande mix transaktioner måste ha. Detta skulle öka anonymitet set som alla transaktioner går igenom *varje* mix skulle ser likadana ut och inte skulle kunna särskiljas utifrån deras värde. Dessutom har en enhetlig storlek över alla blandningar skulle göra det enkelt att använda en serie mixar utan att behöva dela eller slå samman transaktioner.
I praktiken kan det vara svårt att komma överens om en enda bitstorlek som fungerar för alla användare. Om vi väljer det
vara för stor, kommer användare som vill blanda en liten summa pengar inte att kunna göra det. Men om vi väljer att det ska vara det
för liten, användare som vill blanda en stor summa pengar måste dela upp den i ett stort antal bitar som kan vara ineffektiva och kostsamma. Flera standardstorlekar skulle impregna prestanda, men också dela upp anonymitetsuppsättningarna efter bitstorlek. Kanske en serie på två eller tre ökade bitstorlekar kommer att ge en rimlig avvägning mellan effektivitet och integritet.
Klientsidan ska automatiseras. Förutom att försöka länk mynt baserat på transaktionsvärden, en smart motståndare kan försöka olika andra sätt att avanonymisera, till exempel genom att observera tidpunkt för transaktioner. Dessa attacker kan undvikas, men de nödvändiga försiktighetsåtgärderna är för komplexa
och besvärligt för mänskliga användare. Istället klientsidans funktionalitet för att interagera med mixar bör vara automatiserad och inbyggd i integritetsvänlig plånboksmjukvara.

Avgifterna bör vara allt-eller-inget. Mixes är företag och räknar med att få betalt. Ett sätt för en mix att

Debitera avgifter är att ta ett snitt av varje transaktion som användare skickar in. Men detta är problematiskt för anonymitet, eftersom mixtransaktioner inte längre kan vara i standardstorlekar. (Om användare försöker dela och slå ihop sina lite mindre bitar tillbaka till den ursprungliga bitstorleken introducerar det allvarliga och svåranalyserade anonymitetsrisker på grund av de nya kopplingarna mellan mynt som introduceras.)
Blanda inte ihop blandningsavgifter med transaktionsavgifter, som samlas in av gruvarbetare. Blandningsavgifter är
separat från och utöver sådana avgifter.
För att undvika detta problem bör blandningsavgifter vara allt-eller-inget och tillämpas sannolikt. I andra ord, mixen bör svälja hela biten med en liten sannolikhet eller returnera den i sin helhet. För
Om mixen till exempel vill ta ut en blandningsavgift på 0,1 %, bör en ut var 1 000:e gång mixen
svälja hela biten, medan 999 gånger av 1 000 blandningen borde returnera hela biten
utan att ta någon blandningsavgift.
Detta är svårt att åstadkomma. Blandningen måste fatta ett probabilistiskt beslut och övertyga användaren om att det
fuskade inte: att den inte påverkade sin slumptalsgenerator så att den har (säg) en 1 % sannolikhet för behåller en bit som avgift, istället för 0,1%. Kryptografi ger ett sätt att göra detta, och vi kommer att hänvisa dig till *Mixcoin* papper i Ytterligare läsning sektionen för detaljer. Tidningen talar också om olika
sätt på vilka blandningar kan förbättra deras tillförlitlighet.
Blandning i praktiken. Från och med 2015, det finns inte en fungerande mix ekosystem. Det finns många mixtjänster
där ute, men de har låga volymer och därför små anonymitetsuppsättningar. Ännu värre har många blandningar rapporterats stjäla bitcoins. Kanske är svårigheten att "bootstrapping" ett sådant ekosystem en anledningen till att det aldrig har kommit igång. Med tanke på blandningarnas tvivelaktiga rykte är det inte många som kommer att göra det
vill använda dem, vilket resulterar i låga transaktionsvolymer och därmed dålig anonymitet. Det finns en gammal säger att *anonymitet älskar företaget* . Det vill säga, ju fler som använder en anonymitetstjänst, desto bättre anonymitet det kan ge. Dessutom, i avsaknad av mycket pengar att tjäna på att tillhandahålla annonserade tjänster, kan mixoperatörer frestas att stjäla pengar istället, vilket vidmakthåller cykeln av opålitliga blandningar.
Dagens mixar följer inte någon av de principer vi har lagt ut. Varje mix fungerar oberoende och tillhandahåller vanligtvis ett webbgränssnitt, med vilket användaren interagerar manuellt för att specificera mottagandet
adress och andra nödvändiga parametrar. Användaren får välja det belopp som de skulle vilja
gillar att blanda. Mixen kommer att ta ett snitt av varje transaktion som en blandningsavgift och skicka resten till Destinations adress.
Vi tror att det är nödvändigt för blandningar (och plånboksmjukvara) att gå över till modellen vi presenterade i ordning
för att uppnå stark anonymitet, motstå smarta attacker, tillhandahålla ett användbart gränssnitt och attrahera high volymer. Men det återstår att se om ett robust mixekosystem någonsin kommer att utvecklas.
181

6.4 Decentraliserad blandning

Decentraliserad blandning är idén att bli av med blandningstjänster och ersätta dem med en peer-to-peer-protokoll genom vilket en grupp användare kan blanda sina mynt. Som du kan föreställa dig, detta tillvägagångssätt är bättre filosofiskt anpassat till Bitcoin.
Decentralisering har också mer praktiska fördelar. För det första har den inte bootstrapping-problemet: användare behöver inte vänta på att välrenommerade centraliserade mixar kommer till stånd. För det andra är stöld omöjligt vid decentraliserad blandning; protokollet säkerställer att när du lägger in bitcoins som ska blandas,

du får tillbaka bitcoins av samma värde. På grund av detta, även om vissa centrala samordning vänder
ut för att vara till hjälp vid decentraliserad blandning är det lättare för någon att sätta upp en sådan tjänst eftersom de
behöver inte övertyga användarna om att de är pålitliga. Slutligen, på vissa sätt kan decentraliserad blandning
ge bättre anonymitet.

Coinjoin. Den huvudsakliga förslag till decentraliserad blandning kallas Coinjoin. I detta protokoll, olika användare
skapa tillsammans en enda Bitcoin-transaktion som kombinerar alla deras indata. Den tekniska nyckelprincipen
som gör det möjligt för Coinjoin att fungera är detta: när en transaktion har flera ingångar som kommer från olika
adresser, signaturerna som motsvarar varje ingång är separata från och oberoende av var och en
Övrig. Så dessa olika adresser kunde kontrolleras av olika personer. Du behöver inte ett parti
för att samla in alla privata nycklar.

Figur 6.9. En Coinjoin-transaktion.

Detta gör att en grupp användare kan blanda sina mynt med en enda transaktion. Varje användare anger en ingång
och utgående adress och tillsammans bildar de en transaktion med dessa adresser. Ordningen på inmatningen
och utgående adresser är randomiserade så att en utomstående inte kommer att kunna bestämma mappningen mellan
ingångar och utgångar. Deltagarna kontrollerar att deras utgående adress ingår i transaktionen och att
den får samma mängd Bitcoin som de matar in (minus eventuella transaktionsavgifter). När de
har bekräftat detta, undertecknar de transaktionen.

182

Sida 84

Någon som tittar på den här transaktionen i blockkedjan - även om de vet att det är en Coinjoin
transaktion — kommer inte att kunna bestämma mappningen mellan ingångarna och utgångarna. Från en
outsiders perspektiv mynten har blandats, vilket är kärnan i Coinjoin.

Det vi har beskrivit hittills är bara en omgång av mixning. Men principerna som vi diskuterade tidigare
gäller fortfarande. Du skulle vilja upprepa denna process med (förmodligen) olika grupper av användare. Det skulle du också
vill se till att chunkstorlekarna är standardiserade så att du inte introducerar någon sida
kanaler.

Låt oss nu fördjupa oss i detaljerna i Coinjoin, som kan delas upp i 5 steg:

1.
Hitta kamrater som vill blanda
2.
Byt in-/utgångsadresser
3.
Konstruera transaktion
4.
Skicka runt transaktionen. Varje kamrat skriver under efter att ha verifierat att deras utdata finns.
5.
Sänd transaktionen

Att hitta kamrater. Först en grupp kamrater som alla vill blanda behov av att hitta varandra. Det här kan vara
underlättas av servrar som fungerar som "vattenhål", så att användare kan ansluta och gruppera sig.
Till skillnad från centraliserade mixar är dessa servrar inte i en position att stjäla användares pengar eller kompromissa
anonymitet.

Utbyta adresser. När en referensgrupp har bildats, måste kamrater utbyta ingång och
mata ut adresser med varandra. Det är viktigt för deltagarna att byta dessa adresser i sådana
ett sätt att även de andra medlemmarna i kamratgruppen inte känner till kartläggningen mellan input och

utgående adresser. Annars, även om du utför en coinjoin-transaktion med en förmodat slumpmässig uppsättning av jämnåriga kan en motståndare kanske ta sig in i gruppen och notera kartläggningen av ingångar till utgångar. För att byta adresser på ett okopplat sätt behöver vi en anonym kommunikation protokoll. Vi kunde använda Tor-nätverket, som vi tittade på tidigare, eller ett anonymt speciellt ändamål routingprotokoll som kallas ett dekrypteringsmix-nät.

Samla signaturer och denial of service. När in- och utgångar har kommunicerats, en av dessa användare — det spelar ingen roll vem — kommer då att konstruera transaktionen som motsvarar dessa in- och utgångar. Den osignerade transaktionen kommer sedan att skickas runt; varje kamrat kommer att verifiera att dess ingångs- och utgångsadress är korrekt inkluderade och underteckna.

Om alla kamrater följer protokollet fungerar detta system bra. Vilken kamrat som helst kan sätta ihop transaktionen och vilken som helst peer kan sända transaktionen till nätverket. Två av dem kunde till och med sända den oberoende av; det kommer att publiceras endast en gång till blockkedjan, naturligtvis. Men om en eller flera av de kamrater vill vara störande är det lätt för dem att starta en överbelastningsattack, vilket förhindrar protokoll från att slutföras.

183

Sida 85

I synnerhet kan en peer delta i den första fasen av protokollet och tillhandahålla dess input och output adresser, men vägrar sedan att skriva under i den andra fasen. Alternativt, efter att ha undertecknat transaktionen, a störande kamrater kan försöka ta den input som den gav till sina kamrater och spendera den i någon annan transaktion istället. Om den alternativa transaktionen vinner loppet på nätverket, kommer den att bekräftas först och Coinjoin-transaktionen kommer att avvisas som en dubbel utgift.

Det har funnits flera förslag för att förhindra denial of service i Coinjoin. En är att lägga en kostnad på delta i protokollet, antingen via ett bevis på arbete (analogt med gruvdrift), eller genom ett bevis på bränning, en teknik för att bevisligen förstöra en liten mängd bitcoins som du äger, vilket vi studerade i kapitel 3. Alternativt finns det kryptografiska sätt att identifiera en icke-kompatibel deltagare och sparka dem ut ur gruppen. För detaljer, se avsnittet Ytterligare läsning.

Högnivåflöden. Vi nämnde sidokanalerna tidigare. Vi ska nu titta närmare på hur knepig sida kanaler kan vara. Låt oss säga att Alice får en mycket specifik mängd bitcoins, säg 43,12312 BTC, vid en särskild adress på veckobasis, kanske som hennes lön. Antag vidare att hon har en vana att automatiskt och omedelbart överföra 5% av det beloppet till hennes pensionskonto, dvs en annan Bitcoin-adress. Vi kallar detta överföringsmönster för ett flöde på hög nivå. Ingen blandningsstrategi kan effektivt dölja det faktum att det finns ett samband mellan de två adresserna i det här scenariot. Tror om mönstren som kommer att synas i blockkedjan: de specifika mängderna och tidpunkten är utomordentligt osannolikt att inträffa av en slump.

Figur 6,10: Merge undvikande . Alice vill köpa en tekanna för 8 BTC. Butiken ger henne två adresser och hon betalar 5 till den ena och 3 till den andra, vilket motsvarar hennes tillgängliga insatsmedel. Detta förhindrar avslöjar att dessa två adresser båda tillhörde Alice.

En teknik som kan bidra till att återfå unlinkability i närvaro av hög nivå flöden kallas **merge undvikande** , föreslagit av Bitcoin-utvecklaren Mike Hearn. I allmänhet skapar en användare för att göra en betalning

184

Sida 86

en enda transaktion som kombinerar så många mynt som behövs för att betala hela beloppet till en enda adress. Tänk om de kunde undvika behovet av att slå samman och följaktligen länka ihop alla sina input?

Protokollet för att undvika sammanslagning möjliggör detta genom att tillåta mottagaren av en betalning att tillhandahålla flera
utgående adresser — så många som behövs. Avsändaren och mottagaren kommer överens om en uppsättning av valörer för att dela upp betalningen i och utföra den med flera transaktioner, som visas i
Figur 6.10.
Förutsatt att butiken så småningom kombinerar dessa två betalningar med många andra insatser från andra betalningar den har mottagit kommer det inte längre att vara uppenbart att dessa två adresser var förknippade med varandra. Butiken bör undvika att kombinera dessa två indata så snart den tar emot dem eller så
det kommer fortfarande att stå klart att de gjordes av samma enhet. Dessutom kanske Alice vill undvika att skicka två betalningar på exakt samma tidpunkt, vilket på liknande sätt kan avslöja denna information.
I allmänhet kan dock undvikande av sammanslagningar hjälpa till att mildra problemet med flöden på hög nivå: en motståndare
kanske inte kan urskilja ett flöde om det är uppdelat i många mindre flöden som inte är kopplade till varje Övrig. Det besegrar också adressklustringstekniker som bygger på att mynt spenderas tillsammans i en singel transaktion.

6.5 Zerocoin och Zerocash

Inga anonymitetslösningar för kryptovaluta har orsakat så mycket spänning som Zerocoin och dess efterträdare Zerocash. Det är både på grund av den geniala kryptografi som de använder och på grund av av den kraftfulla anonymitet som de lovar. Medan alla anonymitetsförbättrande teknologier som vi har sett hittills lägga anonymitet ovanpå kärn protokollet ***Zerocoin*** och ***Zerocash*** införliva anonymitet på protokollnivå. Vi kommer att presentera en översikt över protokollet här och nödvändigtvis förenkla vissa detaljer, men du kan hitta referenser till originaldokumenten i den ytterligare Läsavdelning.

Kompatibilitet. Som vi ser, den starka anonymitet garantier för Zerocoin och Zerocash har sitt kostnad: till skillnad från centraliserad blandning och Coinjoin är dessa protokoll inte kompatibla med Bitcoin som det
står idag. Det är tekniskt möjligt att distribuera Zerocoin med en mjuk gaffel till Bitcoin, men det är praktiskt svårigheterna är tillräckligt allvarliga för att göra detta omöjligt. Med Zerocash är en gaffel inte ens möjlig, och en altcoin är det enda alternativet.

Kryptografiska garantier. Zerocoin och Zerocash inkorporerar protokollnivå blandning, och anonymitetsegenskaper kommer med kryptografiska garantier. Dessa garantier är kvalitativt bättre än de för de andra blandningsteknikerna som vi har diskuterat. Du behöver inte lita på någon
– blandningar, peers eller mellanhänder av något slag, eller till och med gruvarbetare och konsensusprotokollet – för att
säkerställa din integritet. Löftet om anonymitet förlitar sig endast på motståndarens beräkningsgränser, som med de flesta kryptografiska garantier.

Zerocoin. För att förklara Zerocoin, vi först införa begreppet Basecoin. Basecoin är ett Bitcoin-liknande altcoin, och Zerocoin är en förlängning av detta altcoin. Nyckelfunktionen som ger anonymitet är det du kan konvertera basmynt till nollmynt och tillbaka igen, och när du gör det bryter det länken mellan det ursprungliga basmyntet och det nya basmyntet. I detta system är Basecoin den valuta som du gör transaktioner, och Zerocoin tillhandahåller bara en mekanism för att byta in dina basmynt mot nya som är det kan inte kopplas till de gamla.

Du kan se varje nollmynt du äger som en token som du kan använda för att bevisa att du ägde ett basmynt och gjorde det oanvändbart. Beviset avslöjar inte vilket basmynt du ägde, bara det du ägde ett basmynt. Du kan senare lösa in detta bevis mot ett nytt basmynt genom att presentera detta bevis till gruvarbetarna. En analogi är att gå in på ett kasino och byta ut dina pengar mot pokermarker. Dessa tjänar

som bevis på att du satt in lite kontanter, som du senare kan byta mot olika kontanter av samma värde på att lämna kasinot. Naturligtvis, till skillnad från pokermarker, kan du faktiskt inte göra något med en nollmynt förutom att hålla fast vid det och senare lösa in det mot ett basmynt.
För att få detta att fungera i en kryptovaluta implementerar vi dessa bevis kryptografiskt. Vi måste se till att varje bevis endast kan användas en gång för att lösa in ett basmynt. Annars skulle du kunna tjäna basmynt gratis genom att förvandla ett basmynt till ett nollmynt och sedan lösa in det mer än en gång.
Zero-kunskap bevis. Nyckeln kryptografiska verktyg vi använder är en nollkunskapsbevis, vilket är en sätt för någon att bevisa ett (matematiskt) påstående utan att avslöja någon annan information som leder till att det påståendet är sant. Anta till exempel att du har gjort mycket arbete för att lösa en hash pussel, och du vill övertyga någon om detta. Du vill med andra ord bevisa påståendet
Jag känner x så att H(x‖ 〈andra kända ingångar〉) < 〈mål〉 .
Du kan naturligtvis göra detta genom att avslöja x. Men ett noll-kunskapsbevis gör att du kan göra detta i en sådan sätt att den andra personen inte är klokare på värdet av x efter att ha sett beviset än vad de var innan.
Du kan också bevisa ett påstående som "Jag vet x så att H(x) tillhör följande uppsättning: {...}".
Beviset skulle inte avslöja något om x, inte heller om vilket element i mängden som är lika med H(x). Zerocoin förlitar sig avgörande på noll-kunskapsbevis och i själva verket är påståendena som bevisats i Zerocoin mycket liknande det senare exemplet. I den här boken kommer vi att behandla nollkunskapsbevis som svarta lådor. Väl presentera de egenskaper som uppnåtts med nollkunskapsbevis och visa var de är nödvändiga i protokoll, men vi kommer inte att fördjupa oss i de tekniska detaljerna om hur dessa bevis implementeras. Nollkunskapsbevis är en hörnsten i modern kryptografi och utgör grunden för många protokoll. Återigen hänvisar vi den motiverade läsaren till avsnittet Ytterligare läsning för mer detaljerad information
behandling.
Prägla Zerocoins. Zerocoins tillkommit genom prägling, och vem som helst kan mint en zerocoin. De kommer i standardvalörer. För enkelhetens skull antar vi att det bara finns en valör värt 1,0 nollmynt, och att varje nollmynt är värt ett basmynt. Även om vem som helst kan prägla ett Zerocoin,

---

att bara prägla en ger det inte automatiskt något värde – du kan inte få gratis pengar. Det får värde bara när du lägger den på blockkedjan, och om du gör det kommer du att behöva ge upp ett basmynt.
Att prägla en Zerocoin använder du en kryptografisk ***engagemang*** . Minns från kapitel 1 att ett åtagande Schema är den kryptografiska analogen för att försegla ett värde i ett kuvert och lägga det på ett bord i allas syn.
Figur 6.11. Sikte ett serienummer Den verkliga världen analog av en kryptografisk engagemang är försegla ett värde inuti ett kuvert.
Att prägla ett nollmynt görs i tre steg:
1. Generera serienummer *S* och en slumpmässig hemlig *r*
2. Beräkna *Commit (S, R)* , åtagandet att serienumret
3. Publicera åtagandet i blockkedjan som visas i figur 6.12. Detta bränner ett basmynt, vilket gör det oanvändbart och skapar ett Zerocoin. Håll S och r hemliga tills vidare.
Figur 6,12: Att sätta en zerocoin på blocket kedjan . För att sätta ett nollmynt på blockkedjan skapar du ett speciell "mint"-transaktion vars utgående "adress" är nollmyntens kryptografiska åtagande serienummer. Ingången till mynttransaktionen är ett basmynt, som nu har använts för att skapa nollmyntet. Transaktionen ger *inte* avslöja serienumret.
För att spendera ett nollmynt och lösa in ett nytt basmynt måste du bevisa att du tidigare har präglat ett nollmynt. Du kan göra detta genom att öppna tidigare engagemang, det vill säga, avslöjar *S* och *r* . Men det här gör kopplingen mellan ditt gamla basmynt och ditt nya basmynt uppenbar. Hur kan vi bryta länk?

Sida 89

Det är här nollkunskapsbeviset kommer in. När som helst kommer det att finnas många åtaganden på blocket kedjan - låt oss kalla dem c_1, c_2, ..., c_n.

Här är stegen för att spendera ett nollmynt med serienummer S för att lösa in ett nytt basmynt:

● Skapa en speciell "spend"-transaktion som innehåller S, tillsammans med ett noll-kunskapsbevis på påstående:

”Jag vet *r* sådan att *Commit (S, R)* är i mängden { c_1, c_2, ..., c_n }”.

● gruvarbetare kommer verifiera din nollkunskapsbevis som fastställer din *förmåga* att öppna en av de nollmyntåtaganden på blockkedjan, utan att faktiskt öppna den.

● Gruvarbetare kommer också att kontrollera att serienumret S aldrig har använts i några tidigare utgifter transaktion (eftersom det skulle vara en dubbelutgift).

● Resultatet av din utgiftstransaktion kommer nu att fungera som ett nytt basmynt. För utgående adress, du bör använda en adress som du äger.

Figur 6,13: spendera en zerocoin . Utgiftstransaktionen avslöjar serienumret S begått av den tidigare mynta transaktionen, tillsammans med en nollkunskapsbevis att S motsvarar *några* tidigare myntaffär. Till skillnad från en mynttransaktion (eller en normal Bitcoin/basecoin-transaktion), spenderas transaktionen har inga ingångar och därför ingen signatur. Istället tjänar noll-kunskapsbeviset till fastställa dess giltighet.

När du har spenderat ett nollmynt blir serienumret offentligt och du kommer aldrig att kunna lösa in det detta serienummer igen. Och eftersom det bara finns ett serienummer för varje nollmynt betyder det det varje nollmynt kan bara spenderas en gång, precis som vi krävde för säkerheten.

. *Anonymitet* Observera att *r* hålls hemlig hela, varken mynt- eller utgiftstransaktionen avslöjar det. Det betyder att ingen vet vilket serienummer som motsvarar vilket nollmynt. Det här är nyckelbegreppet bakom Zerocoins anonymitet. Det finns ingen länk på blockkedjan mellan myntverket transaktion som begick ett serienummer S och utgiftstransaktionen som senare avslöjade S för lösa in ett basmynt. Detta är en magiskt klingande egenskap som är möjlig genom kryptografi men vi skulle inte hamna i ett fysiskt kuvertbaserat system. Det är som om det ligger ett gäng förseglade kuvert på en tabell med olika serienummer, och du kan bevisa att ett visst serienummer är ett av dem, utan att behöva avslöja vilken och utan att behöva öppna några kuvert.

Sida 90

. *Effektivitet* Minns uttalande som är visat i en spendera transaktion:

”Jag vet *r* sådan att *Commit (S, R)* är i mängden { c_1, c_2, ..., c_n }”.

Detta låter som att det skulle vara fruktansvärt ineffektivt att implementera, eftersom storleken på noll-kunskapen bevis skulle växa linjärt när *n* ökar, vilket är antalet zerocoins som har *någonsin* varit präglat. Anmärkningsvärt, Zerocoin lyckas göra storleken på dessa bevis endast logaritmisk i *n* . Notera

att även om *uttalandet* som skall bevisas har en linjär längd, behöver det inte ingå tillsammans med beviset. Uttalandet är implicit; det kan slutas av gruvarbetarna eftersom de kan uppsättningen av alla nollmynt i blockkedjan. Själva beviset kan vara mycket kortare. Ändå jämfört med Bitcoin, Zerocoin lägger fortfarande till en ganska stor overhead, med bevis på cirka 50 kB i storlek.

Betrodd installationen . Ett av de kryptografiska verktygen som används för att bygga Zerocoin (RSA-ackumulatorer) som kräva en engångs *betrodd installationen* . Speciellt behöver en pålitlig part välja två stora primtal p och q och publicera $N = p \cdot q$ som är en parameter som alla kommer att använda för livslängden hos systemet. Tror av N som en publik nyckel, med undantag för alla Zerocoin i motsats till en viss enhet. Så länge som betrodd part förstör eventuella uppgifter om p och q är systemet tros vara säker. I synnerhet detta vilar på det allmänt trodda antagandet att det är omöjligt att faktorisera ett tal som är en produkt av två stora primtal. Men om *någon* känner till hemliga faktorer p och q (kallad ”luckan”), då de skulle kunna skapa nya nollmynt åt sig själva utan att upptäckas. Så dessa hemliga ingångar måste användas en gång för att generera de offentliga parametrarna och sedan förstöras på ett säkert sätt.

Det finns ett intressant sociologiskt problem här. Det är inte klart hur ett företag kan välja N och övertyga alla om att de säkert har förstört faktorerna p och q som användes under uppstart. Det har kommit olika förslag på hur man ska uppnå detta, bland annat ”tröskel kryptografi”tekniker som gör en uppsättning av delegater till gemensamt compute N på ett sådant sätt att så länge eftersom någon av dem tar bort sina hemliga inmatningar kommer systemet att förbli säkert.

Det är också möjligt att använda en något annorlunda kryptografisk konstruktion för att undvika den pålitliga installationen.

Specifikt har det visats att helt enkelt generera ett mycket stort slumpvärde för N är säker med hög sannolikhet, eftersom siffran troligen inte helt kan räknas in. Tyvärr detta har en enorm effektivitetsträff och anses därför inte vara praktisk.

Zerocash. Zerocash är en annan anonym kryptovaluta som bygger på idén om Zerocoin men tar kryptografin till nästa nivå. Den använder en kryptografisk teknik som kallas nollkunskap SNARKs (zk-SNARKS) som är ett sätt att göra nollkunskapsbevis mycket mer kompakta och effektivt att verifiera. Resultatet är att effektiviteten i systemet totalt sett når en punkt där den blir möjligt att köra hela nätverket utan att behöva ett basmynt. Alla transaktioner kan göras på ett nollkunskapssätt. Som vi såg stöder Zerocoin vanliga transaktioner när du inte gör det behöver inte länkas, utökat med beräkningsmässigt dyra transaktioner som endast används för blandning. Mixtransaktionerna är av fasta valörer och uppdelning och sammanslagning av värden kan händer bara i Basecoin. I Zerocash är den distinktionen borta. Transaktionsbeloppen är nu inne åtagandena och inte längre synliga på blockkedjan. De kryptografiska bevisen säkerställer att splittring och sammanslagning sker korrekt och att användare inte kan skapa zerocash ur tomma luften.

Det enda som huvudboken registrerar offentligt är förekomsten av dessa transaktioner, tillsammans med bevis som tillåter gruvarbetarna att verifiera alla egenskaper som behövs för att systemet ska fungera korrekt. Varken adresser eller värden avslöjas i blockkedjan vid något tillfälle. De enda användare som behöver vet beloppet för en transaktion är avsändaren och mottagaren av den specifika transaktionen. De gruvarbetare behöver inte veta transaktionsbeloppen. Naturligtvis, om det finns en transaktionsavgift, gruvarbetarna behöver veta den avgiften, men det äventyrar inte din anonymitet.

Möjligheten att köra som ett helt ospårbart system av transaktioner sätter zerocash i sin egen kategori när det kommer till anonymitet och integritet. Zerocash är immun mot sidokanalangreppen mot blandning eftersom huvudboken inte längre innehåller transaktionsbelopp.

Ställa in Zerocash. I termer av dess tekniska egenskaper kan Zerocash låter för bra för att vara sant. Det finns verkligen en hake. Precis som Zerocoin kräver Zerocash "offentliga parametrar" för att ställa in

noll-kunskapssäkert system. Men till skillnad från Zerocoin, som kräver bara ett nummer *N* som ligger bara en några hundra byte kräver Zerocash en enorm uppsättning offentliga parametrar — över en gigabyte lång.
Än en gång, för att generera dessa offentliga parametrar kräver Zerocash *slump och hemliga ingångar* , och om *någon* känner till dessa hemliga ingångar det äventyrar säkerheten i systemet genom att möjliggöra oupptäckbara dubbla utgifter.
Vi kommer inte att fördjupa oss i utmaningen att sätta upp ett zk-SNARK-system här. Det förblir en aktivt forskningsområde, men från och med 2015 vet vi inte exakt hur vi ska lägga upp systemet i praktiken i en tillräckligt pålitligt sätt. Hittills har zk-SNARK inte använts i praktiken.
Att sätta ihop allt. Låt oss nu jämföra de lösningar som vi har sett, både i termer av anonymitetsegenskaper som de tillhandahåller och i termer av hur utplacerbara de är i praktiken.

Systemet
Typ
Anonymitetsattacker
Utplaceringsbarhet

Bitcoin
pseudonym-
analys av transaktionsdiagram
standard
Manuell
blandning
blanda
analys av transaktionsdiagram,
dåliga mixar/kamrater
användbar idag
Kedja av blandningar
eller coinjoins
blanda
sidokanaler, dåliga
mixar/kamrater
bitcoin-kompatibel
Nollmynt
kryptografiskt
blanda
sidokanaler (eventuellt)
altcoin, pålitlig installation
Zerocash
Ospårbar
ingen känd
altcoin, pålitlig installation

Tabell 6.14: En jämförelse av anonymitet tekniker som presenteras i detta kapitel

---

Vi börjar med själva Bitcoin, som redan är utplacerat och är "default"-systemet. Men det är bara pseudonym och vi har sett att kraftfull analys av transaktionsdiagram är möjlig. Vi tittade på sätt att gruppera stora grupper av adresser, och hur man ibland kan koppla verkliga identiteter till dessa kluster.

Nästa nivå av anonymitet är att använda en enda mix på ett manuellt sätt, eller göra en Coinjoin genom att hitta kamrater
manuellt. Detta skymmer kopplingen mellan input och output men lämnar för många potentiella ledtrådar i transaktionsdiagram. Dessutom kan blandningar och kamrater vara skadliga, hackade eller tvingade att avslöja deras register. Även om det är långt ifrån perfekt när det gäller anonymitet, finns det blandtjänster och så är det här alternativet
användbar idag.
Den tredje nivån vi tittade på är en kedja av blandningar eller Coinjoins. Anonymitetsförbättringen kommer från det faktum att det är mindre beroende av någon enskild blandning eller grupp av kamrater. Funktioner som standardiserad chunk
storlekar och automatisering på klientsidan kan minimera informationsläckor, men vissa sidokanaler är det fortfarande
närvarande. Det finns också risken för en motståndare som kontrollerar eller samarbetar med flera mixar eller kamrater. Plånböcker och tjänster som implementerar en kedja av mixar skulle kunna distribueras och införas idag, men såvitt vi vet är en säker mix-chain-lösning ännu inte lätt tillgänglig.
Därefter såg vi att Zerocoin bakar in kryptografi direkt i protokollet och tar med en matematisk
garanti för anonymitet. Vi tror att vissa sidokanaler fortfarande är möjliga, men det är verkligen överlägset de andra blandningsbaserade lösningarna. Zerocoin skulle dock behöva lanseras som ett altcoin.
Till sist tittade vi på Zerocash. På grund av sin förbättrade effektivitet kan Zerocash köras som en helt ospårbar – och inte bara anonym – kryptovaluta. Men precis som Zerocoin är Zerocoin det inte Bitcoin-kompatibel. Ännu värre, det kräver en komplex installationsprocess som samhället fortfarande räknar med ut hur man bäst gör.
Vi har täckt mycket teknik i det här kapitlet. Låt oss nu ta ett steg tillbaka. Bitcoins anonymitet (och potential för anonymitet) är kraftfull och får makt när den kombineras med andra teknologier, särskilt anonym kommunikation. Som vi kommer att se i nästa kapitel är detta det potenta kombinationen bakom Silk Road och andra anonyma onlinemarknadsplatser.
Trots sin kraft är anonymiteten bräcklig. Ett misstag kan skapa en oönskad, oåterkallelig länk. Men anonymitet är värt att skydda, eftersom den har många bra användningsområden utöver de uppenbart dåliga. Medan
dessa moraliska distinktioner är viktiga, vi finner oss själva oförmögna att uttrycka dem på en teknisk nivå. Anonymitetsteknologier verkar vara djupt och i sig moraliskt tvetydiga, och som ett samhälle
måste lära sig att leva med detta faktum.
Bitcoin anonymitet är ett aktivt område för teknisk innovation såväl som etisk debatt. Det gör vi fortfarande inte vet vilket anonymitetssystem för Bitcoin, om något, kommer att bli framträdande eller mainstream. Det är en fantastisk möjlighet för dig - oavsett om du är utvecklare, beslutsfattare eller användare - att engagera dig och

---

göra en insats. Förhoppningsvis har det du har lärt dig i det här kapitlet gett dig rätt
bakgrund för att göra det.

Vidare läsning

Till och med mer än de ämnen som diskuterats i tidigare kapitel, är anonymitetsteknologier ständigt utvecklar och är ett aktivt område för forskning om kryptovaluta. Det bästa sättet att hänga med i det senaste i detta fält är att börja med de tidningar som listas här, och att leta efter papper som citerar dem.
Uppsatsen "Fistful of bitcoins" om analys av transaktionsdiagram:
Meiklejohn, Sarah, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker och Stefan Savage. En handfull Bitcoins: karakteriserar betalningar bland män utan namn . In Proceedings of the 2013 conference on Internet measurement conference, 2013.
En studie av blandningsteknologier och källan till principerna för effektiv blandning som vi diskuterade:

Bonneau, Joseph, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll och Edward W. . Felten Mixcoin: anonymitet för Bitcoin med ansvariga mixar . Finansiella Kryptering 2014.
En studie av att blanda tjänster i praktiken, som visar att många inte är välrenommerade:
Malte Moser, Rainer Böhme och Dominic Breuker. En utredning penningtvätt Verktyg i Bitcoin Ecosystem . 2013 eCrime Researchers Summit.
Coinjoin presenterades på Bitcoin-forum av Bitcoin Core-utvecklaren Greg Maxwell:
. Maxwell, Gregory CoinJoin: Bitcoin sekretess för den verkliga världen . Bitcoin Forum, 2013.
Zerocoin utvecklades av kryptografer från Johns Hopkins University. Tänk på Zerocoin och Zerocash har den mest komplexa kryptografin av alla scheman som vi har diskuterat i den här boken.
Miers, Ian, Christina Garman, Matthew Green, och Aviel D. Rubin. Zerocoin: Anonym distribueras E-Cash från Bitcoin . Proceedings of 2013 IEEE Symposium on Security och sekretess 2013.
Zerocoin-författarna slog sig ihop med andra forskare som hade utvecklat SNARK-tekniken. Detta samarbete resulterade i Zerocash:
Ben Sasson, Eli, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer och . Madars Virza Zerocash: Decentraliserad anonyma betalningar från Bitcoin . Proceedings of 2013 IEEE Symposium om säkerhet och integritet, 2014.
192

Sida 94

En alternativ design till Zerocoin är CryptoNote, som använder olika kryptografi och erbjudanden olika anonymitetsegenskaper. Vi diskuterade inte i det här kapitlet på grund av utrymmesbrist, men det är en intressant designstrategi:
Nicolas van Saberhagen. CryptoNote v. 2.0 .
Denna klassiska bok om kryptografi innehåller ett kapitel om nollkunskapsbevis:
Goldreich, Oded. Foundations of Cryptography: Volym 1. Cambridge university press, 2007.
Tidningen som beskriver den tekniska designen av det anonyma kommunikationsnätverket Tor:
Dingledine, Roger, Nick Mathewson och Paul Syverson. Tor: Den andra generationens lök router . Naval Research Lab Washington DC, 2004.
Uppsatsen "systematisering av kunskap" om Bitcoin och kryptovalutor, särskilt avsnitt VII om Anonymitet och integritet:
Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll och Edward W. Felten. Forskningsperspektiv och utmaningar för Bitcoin och Cryptocurrencies . Förfarandet 2015 IEEE Security and Privacy Conference, 2015.
193

Sida 95

Kapitel 7: Gemenskap, politik och reglering

I det här kapitlet kommer vi att titta på alla sätt som Bitcoin-världen och kryptovalutatekniken har berör människors värld. Vi kommer att diskutera Bitcoin-gemenskapens interna politik såväl som sätt som Bitcoin interagerar med traditionell politik, nämligen frågor om brottsbekämpning och reglering.

7.1: Konsensus i Bitcoin

Låt oss först titta på konsensus i Bitcoin, det vill säga hur driften av Bitcoin förlitar sig på bildandet av konsensus bland människor. Det finns tre typer av konsensus som måste verka för Bitcoin för att bli framgångsrik.

1. Konsensus om regler. Med regler menar vi saker som vad som gör en transaktion eller ett block giltigt kärnprotokoll och dataformat som är involverade i att få Bitcoin att fungera.

Du måste ha en konsensus om dessa saker så att alla olika deltagare i systemet kan prata med varandra och komma överens om vad som händer.

2. Konsensus om historien. Det vill säga konsensus om vad som finns och inte finns i blockkedjan, och därför enighet om vilka transaktioner som har skett. När du väl har det är vad som följer en konsensus om vilka mynt - vilka outnyttjade utgångar - som finns och vem som äger dem.

Denna konsensus är resultatet av de processer vi har tittat på i kapitel 2 och andra tidigare kapitel från vilken blockkedjan är uppbyggd och genom vilka noder kommer till konsensus om innehållet i blockkedja. Detta är den mest välbekanta och mest tekniskt invecklade typen av konsensus i Bitcoin.

3. Konsensus om att mynt är värdefulla. Den tredje formen av konsensus är den allmänna överenskommelsen om att bitcoins är värdefulla och i synnerhet konsensus att om någon ger dig en bitcoin idag, då imorgon kommer du att kunna lösa in eller byta ut det mot något av värde.

Vilken valuta som helst, oavsett om det är en fiatvaluta som dollarn eller kryptovaluta som Bitcoin, är beroende av konsensus om att det har ett värde. Det vill säga, du behöver att folk generellt accepterar att det är utbytbart mot något annat av värde, nu och i framtiden.

I en fiat valuta, är detta den *enda* typ av konsensus. Reglerna uppstår inte genom konsensus --- vad är det och är inte en dollarsedel deklareras av fiat. Historien är inte framträdande, men staten är vem som äger vad. Staten är antingen bestäms av fysisk innehav, som med kontanter, eller delegeras till professionella registerförare, dvs banker. I kryptovalutor å andra sidan är regler och historia också föremål för konsensus.

I Bitcoin är denna form av konsensus, till skillnad från de andra, lite cirkulär. Med andra ord, min övertygelse om att bitcoins jag får idag är av värde beror på min förväntning om att andra människor kommer att göra det imorgon
194

---

tror samma sak. Så konsensus om värde bygger på att man tror att konsensus om värde kommer Fortsätta. Detta kallas ibland Tinkerbell-effekten i analogi med Peter Pan där det sägs det Tinkerbell finns för att du tror på henne.

Oavsett om det är cirkulärt eller inte verkar det finnas och det är viktigt för Bitcoin att fungera. Nu, vad är det viktigt med alla tre former av konsensus är att de är sammanflätade med varandra, som figur 7.1 visar.

Figur 7.1: Relationer mellan de tre formerna av konsensus i Bitcoin

Först och främst går konsensus om regler och konsensus om historia ihop. Utan att veta vilken block är giltiga du kan inte ha konsensus om blockkedjan. Och utan konsensus om vilket

block finns i blockkedjan, du kan inte veta om en transaktion är giltig eller om den försöker spendera en redan förbrukad produktion.

Konsensus om historien och konsensus om att mynt är värdefulla hänger också ihop. Konsensus om historia innebär att vi är överens om vem som äger vilka mynt, och det är en förutsättning för att tro att mynten har ett värde — utan konsensus om att jag äger ett visst mynt kan jag inte ha något förväntan att folk kommer att acceptera det myntet från mig som betalning i framtiden. Det är sant omvänt som ja — som vi såg i kapitel 2, är konsensus om värde det som motiverar gruvarbetare att upprätthålla säkerheten i blockkedjan, vilket ger oss konsensus om historien.

Genialiteten i Bitcoins ursprungliga design var att inse att det skulle vara mycket svårt att få tag i någon av dessa typer av konsensus i sig. Konsensus om reglerna i en världsomspännande decentraliserad miljö där det inte finns någon uppfattning om identitet är inte den typ av sak som sannolikt kommer att hända.

Konsensus om historia är på samma sätt ett mycket svårt problem med distribuerad datastruktur som inte är det sannolikt löses på egen hand. Och en konsensus om att någon form av kryptovaluta har värde är också mycket svårt att uppnå. Vad utformningen av Bitcoin och den fortsatta driften av Bitcoin visar är att även om du inte kan bygga någon av dessa former av konsensus av sig själv kan du på något sätt stå upp alla tre tillsammans och få dem att fungera på ett ömsesidigt beroende sätt. Så när vi pratar om hur saker fungerar i Bitcoin-gemenskapen måste vi komma ihåg att Bitcoin förlitar sig på överenskommelse mellan deltagarna och att samförstånd är en bräcklig och ömsesidigt beroende sak.
195

7.2: Bitcoin Core Software

Bitcoin Core är en mjukvara med öppen källkod som är en samlingspunkt för diskussion och debatt om Bitcoins regler.

Bitcoin Core är licensierad under MIT-licensen som är en mycket tillåtande öppen källkodslicens. Det tillåter programvaran som ska användas för nästan alla ändamål så länge som källan tillskrivs och MIT-licensen är inte avskalad. Bitcoin Core är den mest använda Bitcoin-mjukvaran och även de som inte gör det använda det tenderar att titta på det för att definiera vad reglerna är. Det vill säga människor som bygger alternativa Bitcoin
programvara försöker vanligtvis efterlikna de regeldefinierande delarna av Bitcoin Core-programvaran, de delar som
kontrollera giltigheten av transaktioner och block.

Bitcoin Core är de facto regelboken för Bitcoin. Om du vill veta vad som är giltigt i Bitcoin, Bitcoin Core-programvara - eller förklaringar av den - är var du ska leta.

Bitcoin Improvement Proposals. Vem som helst kan bidra med tekniska förbättringar via "pull requests" till Bitcoin Core, en välbekant process i världen av öppen källkod. För mer omfattande förändringar, speciellt protokoll modifieringar, det finns en process som kallas Bitcoin förbättringsförslag eller BIPs. Det här är formella förslag på förändringar av Bitcoin. Vanligtvis kommer en BIP att innehålla en teknisk specifikation för en föreslagen ändring samt en motivering för den. Så om du har en idé om hur förbättra Bitcoin genom att göra några tekniska förändringar, du uppmuntras att skriva en av dessa

dokument och att publicera det som en del av Bitcoin Improvement Proposal-serien, och det kommer då starta en diskussion i samhället om vad man ska göra. Även om den formella processen är öppen för alla, det finns en inlärningskurva för deltagande som alla projekt med öppen källkod.

BIP publiceras i en numrerad serie. Var och en har en mästare, det vill säga en författare som evangeliserar till förmån för det, samordnar diskussionen och försöker bygga en samsyn inom samhället till förmån för gå vidare med eller implementera ett visst förslag.

Det vi sa ovan gäller förslag om att förändra tekniken. Det finns också några BIP som är det rent informativt och existerar bara för att berätta saker för människor som de annars kanske inte visste standardisera någon del av protokollet som tidigare bara specificerats i källkoden, eller som är process orienterade, som pratar om hur saker och ting ska bestämmas i Bitcoin-gemenskapen.

Sammanfattningsvis har Bitcoin en regelbok såväl som en process för att föreslå, specificera och diskutera regel förändringar, nämligen BIP.

Bitcoin Core-utvecklare. För att förstå rollen för Bitcoin Core-mjukvaran måste vi också förstå rollen för Bitcoin Core-utvecklare. Den ursprungliga koden skrevs av Satoshi Nakamoto, som vi återkommer till senare i kapitlet. Nakamoto är inte längre aktiv, utan istället finns det en grupp

---

av utvecklare som underhåller Bitcoin Core. Från och med början av 2015 finns det fem med "commit"-åtkomst till
Kärnlager: Gavin Andresen, Jeff Garzik, Gregory Maxwell, Wladimir J. van der Laan och Pieter Wuille. Kärnutvecklarna leder arbetet med att fortsätta utvecklingen av mjukvaran och är inne ansvar för vilken kod som skjuts in i nya versioner av Bitcoin Core.

Hur mäktiga är dessa människor? I en mening är de väldigt kraftfulla, för det kan man hävda någon av regeländringarna i koden som de gör kommer att skickas i Bitcoin Core och kommer att bli det följt som standard. Det här är människorna som håller pennan som kan skriva in saker i de facto regelbok för Bitcoin. I en annan mening är de inte alls kraftfulla. Eftersom det är öppen källkod, vem som helst kan kopiera den och modifiera den, med andra ord, dela programvaran när som helst, och så om ledningen
utvecklare börjar bete sig på ett sätt som communityn inte gillar, communityn kan gå in i en annan riktning.

Ett sätt att tänka på detta är att säga att de ledande utvecklarna leder paraden. De är ute framför paraden marscherar och paraden kommer i allmänhet att följa dem när de svänger ett hörn, men om de försöker leda paraden till en aktion som är katastrofal, då marscherar paradmedlemmarna bakom dem kan besluta att gå i en annan riktning. De kan uppmana folk på, och så länge de verkar bete sig rimligt, gruppen kommer förmodligen att följa dem, men de har inte formellt makt att tvinga människor att följa dem om de tar systemet i en teknisk riktning som gemenskapen gillar inte.

Låt oss fundera på vad du som användare av ett system kan göra om du inte gillar hur reglerna ser ut eller hur den sköts och jämför den med en centraliserad valuta som en fiatvaluta. I en centraliserad valuta om du inte gillar vad som händer har du rätt att avsluta, det vill säga du kan sluta använda den. Det skulle du

måste försöka sälja vilken valuta du har, och du kanske måste flytta till någonstans med en annan fiat-valuta. Oavsett om det är enkelt eller inte, med en centraliserad valuta är det verkligen ditt enda alternativ.

Med Bitcoin har du förvisso rätt att avsluta, men eftersom det fungerar på ett sätt med öppen källkod, du dessutom har rätt att dela reglerna. Det betyder du och några av dina vänner och kollegor kan bestämma att du hellre vill leva under en annan regeluppsättning, och du kan dela reglerna och gå en annan riktning från de ledande utvecklarna. Rätten till gaffel är mer bemyndigande för användare än rätt att avsluta, och därför har samhället mer makt i ett system som Bitcoin som är öppet källa än vad det skulle göra i ett rent centraliserat system. Så även om de ledande utvecklarna kan se ut som en centraliserad enhet som kontrollerar saker, i själva verket har de inte makten som en rent centraliserad chef eller programvaruägare skulle ha.

Delar in reglerna. Ett sätt att dela mjukvaran och reglerna är att starta en ny blockkedja med en ny genesis block. Detta är ett populärt alternativ för att skapa altcoins, vilket vi kommer att diskutera i kapitel 10. Men för låt oss nu överväga en annan typ av gaffel i reglerna, en där de som gaffel bestämmer sig för att gaffel blockkedja också.

Om du minns skillnaden mellan en hård gaffel och en mjuk gaffel från kapitel 3, talar vi om en hård gaffel här. Vid den punkt då det råder oenighet om reglerna, kommer det att finnas en gaffel i

---

blockkedja, vilket resulterar i två grenar. En gren är giltig enligt regeluppsättning A men ogiltig enligt regeluppsättning B och vice versa. När gruvarbetarna som arbetar under de två regeluppsättningarna separeras kan de inte komma tillbaka tillsammans eftersom varje gren kommer att innehålla transaktioner eller block som är ogiltiga enligt annan regeluppsättning.

Figur 7.2: En gaffel i valutan. Om en gaffel i reglerna leder till en hård gaffel i blockkedjan, valutan själv gafflar och två nya valutor resulterar.

Vi kan tänka på valutan vi hade fram till gaffeln som Bitcoin - den stora glada Bitcoin som alla var överens om. Efter gaffeln är det som om det finns två nya valutor, A-mynt motsvarande regel set A och B-mynt motsvarande regeluppsättning B. I gaffelögonblicket är det som om alla som ägde en bitcoin får ett A-mynt och ett B-mynt. Från den tidpunkten kommer A-mynt och B-mynt att fungera separat som om de vore separata valutor, och de kan fungera oberoende. De två grupper kan fortsätta att utveckla sina regler på olika sätt.

Vi bör betona att det inte bara är programvaran, eller reglerna, eller programvaran som implementerar regler som gafflas — det är själva valutan som gafflas. Detta är en intressant sak som kan hända i en kryptovaluta som inte kunde hända i en traditionell valuta där alternativet att gaffela inte är det tillgängliga för användarna. Såvitt vi vet har varken Bitcoin eller någon altcoin någonsin splittrats på detta sätt, men det är en fascinerande möjlighet.

Hur kan folk reagera på en sådan här gaffel? Det beror på varför gaffeln hände. Det första fallet är där gaffeln inte var tänkt som en oenighet om reglerna, utan istället som ett sätt att börja en altcoin. Någon kan starta en altcoin genom att dela Bitcoins blockkedja om de vill börja med en regeluppsättning som är mycket nära Bitcoins. Detta utgör egentligen inte ett problem för samhället – den altcoin går sin väg, grenarna samexisterar i fred, och vissa människor föredrar att använda bitcoins medan andra föredrar altcoin. Men som vi sa tidigare, så vitt vi vet, är det ingen någonsin startade en altcoin genom att splittra Bitcoins eller en annan befintlig altcoins blockkedja. De har alltid börjat med ett nytt genesisblock.

Det intressanta fallet är om gaffeln återspeglade en kamp mellan två grupper om vad framtiden för Bitcoin borde vara - med andra ord, ett uppror inom Bitcoin-gemenskapen där en undergrupp bestämmer sig för att bryta och bestämmer sig för att de har en bättre uppfattning om hur systemet ska drivas. I det
fall är de två grenarna rivaler och kommer att slåss om marknadsandelar. A-mynt och B-mynt kommer var och en att försöka få
fler handlare att acceptera det och fler människor att köpa det. Var och en kommer att vilja uppfattas som den "riktiga
Bitcoin." Det kan finnas en pr-strid där var och en hävdar legitimitet och framställer den andra som en konstig splittergrupp.

Det troliga resultatet är att en gren så småningom kommer att vinna och den andra kommer att smälta bort. Dessa sorter
av tävlingar tenderar att tippa åt ena hållet. När en av de två blir sedd som mer legitim och får en större marknadsandel kommer nätverkseffekten att råda och den andra blir en nischvaluta och kommer så småningom att falla bort. Vinnarens regeluppsättning och styrningsstruktur kommer att bli de facto regeluppsättning och styrningsstruktur för Bitcoin.

7.3: Intressenter: Vem är ansvarig?

Vilka är intressenterna i Bitcoin, och vem är egentligen ansvarig? Vi har sett hur Bitcoin förlitar sig på konsensus och hur dess regelbok skrivs i praktiken. Vi har analyserat möjligheten till en gaffel eller en slåss om hur reglerna ska vara. Låt oss nu ta upp frågan om vem som har makten till avgöra vem som kan vinna en sådan kamp.

Med andra ord, om det finns en diskussion och förhandling i samhället om regelsättning, och det förhandlingar misslyckas vill vi veta vad som kommer att avgöra resultatet. Generellt sett, i alla förhandling har den part som har det bästa alternativet till ett förhandlat avtal fördelen i en förhandling. Så att ta reda på vem som kan vinna en kamp kommer att berätta för oss vem som har övertaget i samhället
diskussioner och förhandlingar om Bitcoins framtid.

Vi kan göra anspråk på många olika intressenters vägnar:
1. Kärnutvecklare har makten — de skriver regelboken och nästan alla använder deras koda.
2. Gruvarbetare har makten — de skriver historia och bestämmer vilka transaktioner som är giltiga. Om gruvarbetare

besluta att följa en viss uppsättning regler, utan tvekan måste alla andra följa den. Gaffeln med mer gruv kraft bakom det kommer att bygga en starkare, säkrare blockera kedja och så har några förmåga att driva reglerna i en viss riktning. Hur mycket makt de har beror på oavsett om det är en hård gaffel eller en mjuk gaffel, men hur som helst har de lite kraft.
3. Investerare har makten — de köper och håller bitcoins, så det är investerarna som bestämmer om Bitcoin har något värde. Du kan hävda att om utvecklarna kontrollerar konsensus om reglerna och gruvarbetarna kontrollerar konsensus om historien, det är investerarna som kontrollerar konsensus om att Bitcoin har värde. I fallet med en hård gaffel, om investerare oftast bestämmer sig för att sätta sina pengar i antingen A-mynt eller B-mynt, kommer den grenen att uppfattas som legitimt.
4. Handlare och deras kunder har makten — de genererar den primära efterfrågan på Bitcoin. Medan investerare tillhandahåller en del av efterfrågan som stöder priset på valutan,

---

Sida 101

Sida 1

den primära efterfrågan som driver priset på valutan, som vi såg i kapitel 4, härrör från en önskan att förmedla transaktioner med Bitcoin som betalningsteknik. Investerare, enligt till detta argument, gissar bara var den primära efterfrågan kommer att vara i framtiden.
5. Betaltjänster har makten — det är de som hanterar transaktioner. Massor handlare bryr sig inte om vilken valuta de följer och vill helt enkelt använda en betaltjänst som ger dem dollar i slutet av dagen, låter deras kunder betala med en kryptovaluta, och hantera alla risker. Så kanske betaltjänster driver primär efterfrågan och handlare, kunder och investerare kommer att följa dem.

Som du kanske har gissat finns det vissa fördelar med alla dessa argument, och alla dessa enheter har lite kraft. För att lyckas behöver ett mynt alla dessa former av konsensus - en stabil regelbok skriven av utvecklare, gruvkraft, investeringar, deltagande av handlare och kunder, och betaltjänster som stödjer dem. Så alla dessa partier har viss makt att kontrollera resultatet om en kamp om Bitcoins framtid, och det finns ingen som vi kan peka på som den klar vinnare. Det är en stor, ful, rörig konsensusbyggande övning.

Sidebar: styrning av öppna protokoll. Vi har beskrivit ett system där många intressenter med ofullständigt anpassade intressen samarbeta om öppna protokoll och mjukvara och försöka nå teknisk och social konsensus. Detta kan påminna dig om arkitekturen för själva Internet. Det finns verkligen många likheter mellan utvecklingsprocessen för Bitcoin Core och den för Internet. Till exempel påminner BIP-processen om RFC, eller Request For Comments, som är en typ av standarddokument för Internet.

Bitcoin-förespråkande grupper. En annan aktör som är relevant för styrningen av Bitcoin är Bitcoin Fundament. Det grundades 2012 som en ideell organisation. Den har spelat två huvudroller. Det första är finansiering
några av Core-utvecklarna ur stiftelsens tillgångar så att de kan arbeta heltid på fortsätter att utveckla programvaran. Den andra är att prata med regeringen, särskilt USA regeringen, som "Bitcoins röst".

Nu tror vissa medlemmar av Bitcoin-communityt att Bitcoin bör fungera utanför och förutom traditionella nationella regeringar. De tror att Bitcoin ska verka över gränserna och borde inte förklara eller motivera sig för regeringar eller förhandla med dem. Andra har en annan uppfattning.

De ser reglering som oundviklig, önskvärd eller både och. De skulle gilla Bitcoins intressen för att samhället ska vara representerat i regeringen och för att samhällets argument ska höras. De Stiftelsen uppstod delvis för att fylla detta behov, och det är rättvist att säga att dess kontakter med regeringen har gjort mycket för att jämna ut vägen för en förståelse och acceptans av Bitcoin.

Stiftelsen har haft en hel del kontroverser. Vissa styrelseledamöter har hamnat i kriminella eller ekonomiska problem, och det har funnits frågor om i vilken utsträckning några av dem representerar samhället. Stiftelsen har fått kämpa med ledamöter i styrelsen som blivit skulder och måste bytas ut med kort varsel. Det har anklagats för bristande insyn och för

---

faktiskt är i konkurs. I början av 2015 är det i bästa fall oklart om Bitcoin Foundation kommer att ha det mycket av en roll i Bitcoins framtid.

En annan ideell grupp, Coin Center, som lanserades i september 2014 med säte i Washington, DC, har tagit på sig en av rollerna som Bitcoin Foundation spelade, nämligen opinionsbildning och att prata med regeringen.
Coin Center fungerar som en "tankesmedja." Det har fungerat utan större kontroverser i början av 2015. Inte heller Bitcoin Foundation eller Coin Center är ansvarig för Bitcoin längre än någon av de andra intressenter. Framgången och den upplevda legitimiteten för en sådan representativ enhet kommer att styras av hur mycket stöd – och finansiering – den kan få från samhället över tid, som allt annat i denna typ av öppen källkod-baserat ekosystem.

För att sammanfatta, det finns ingen enhet eller grupp som definitivt har kontroll över Bitcoins utveckling. I en annan mening, alla är ansvarig eftersom det är existensen av konsensus om hur systemet kommer att fungera - de tre sammankopplade formerna av konsensus, om regler, om historia och om värde - det styr Bitcoin. Varje regeluppsättning, grupp eller ledningsstruktur som kan upprätthålla den konsensus över tiden kommer, i en mycket verklig mening, att vara ansvarig för Bitcoin.

7.4: Bitcoins rötter

Låt oss titta på Bitcoins rötter – hur det började, vilka dess föregångare var och vad vi vet om dess mystiska grundare.

Cypherpunk och digitala kontanter. Det finns två föregångare till Bitcoin värda att diskutera. En av dessa var *cypherpunk* , en rörelse som förde samman två synpunkter. Först var libertarianism och in särskilt tanken att samhället skulle ha det bättre med antingen ingen regering eller mycket minimal regering. Tillsammans med den starka libertarianska (eller kanske till och med anarkistiska) föreställningen hade vi idén om stark kryptografi och i synnerhet kryptografi med offentlig nyckel som startade i slutet av 1970-talet. Cypherpunkrörelsen var en grupp människor som trodde att med stark online integritet och stark kryptografi kan du omarbeta hur människor interagerar med varandra. I denna världen, trodde cypherpunks, människor kunde skydda sig själva och sina intressen mer effektivt och med mycket mindre aktivitet (eller, som de skulle säga, inblandning) från regeringen.

En av utmaningarna i cypherpunkrörelsen var hur man handskas med pengar i en framtid cypherpunk-värld där människor interagerar online via stark teknisk och kryptografisk

åtgärder. Detta inspirerade till mycket forskning, framför allt ledd av tidigt digitalt kontantarbete av David Chaum och
andra, som syftade till att skapa nya former av digitalt värde som fungerade som pengar, särskilt kontanter, i känslan av att vara anonym och lätt utbytbar. Det finns en hel intressant historia om hur
dessa tekniska idéer utvecklades och varför tidig digital kontanter *inte* sopa världen, men vi kommer inte gå in på det här. I vilket fall som helst kom tidigt arbete på det området tillsammans med cypherpunk-tro och in särskilt önskan att ha en stark valuta som skulle vara decentraliserad, online och relativt
privat för att så fröna som Bitcoin skulle födas ur. Det är också grunden för filosofin som
många av Bitcoins anhängare följer.
201

. Satoshi Nakamoto Bitcoin inleddes 2008 med lanseringen av en vitbok med titeln *Bitcoin: A Peer to Peer Electronic Cash System* som författats av Satoshi Nakamoto. Detta papper, som gjordes
fritt tillgänglig online, är den första beskrivningen av vad Bitcoin är, hur det fungerar och filosofin
bakom dess design. Det är fortfarande en bra resurs för att få en snabb uppfattning om hur Bitcoins tekniska design och
filosofi specificerades. Programvara med öppen källkod som implementerar den specifikationen släpptes snart efter av samma Satoshi Nakamoto, och det var där allt började. Till denna dag är Satoshi en
av Bitcoins centrala mysterier.

Vi vet att namnet Satoshi Nakamoto nästan säkert är en pseudonym som någon person eller
grupp människor adopterade för Bitcoin-relaterade ändamål. Det finns inga tidigare uppgifter om samma Satoshi Nakamoto existerande och Satoshi Nakamoto talade i princip bara om Bitcoin. Satoshis identitet är kopplade till vissa offentliga nycklar och vissa konton på vissa webbplatser. Digitala signaturer med
dessa nycklar erbjuder det enda övertygande beviset på att något har sagts eller gjorts av den verklige Satoshi. Så Satoshi är, samtidigt som den är en pseudonym, också en identitet som kan tala, och som talade speciellt omfattande i Bitcoins tidiga historia. Satoshi var ganska aktiv i arbetet med och skrev om
Bitcoin, och deltagande i onlineforum från 2008 till mitten av 2010, då Satoshi överlämnade
över kontrollen av Bitcoin Core-källkoden till andra utvecklare och har sedan dess inte sagt något. De flesta i samhället känner att Satoshi inte kommer att återvända.

Satoshi påstod sig vara en 37-årig man som bodde i Japan (från 2009). Det finns dock inga bevis
att Satoshi talade eller förstod japanska men vi vet att Satoshi skriver ganska flytande på
Engelska, men ibland med amerikansk stavning ibland med brittisk stavning. Det har varit
många försök att titta på Satoshis text, kod, posttider, maskinidentifierare och så vidare för att försöka svara på frågor som: vad är Satoshis modersmål? Var kommer Satoshi ifrån? Det har till och med
gjorts försök att använda *stylometry* (den algoritm analys av text för författare specifika mönster) till avslöja Satoshis identitet. Satoshis verkliga identitet är fortfarande okänd, trots enstaka självsäker uttalanden från enskilda personer och, åtminstone en gång, en nyhetsorganisation.

Vi vet också att Satoshi förvärvade ett stort antal bitcoins från tidig gruvdrift. I själva
början Satoshi var den enda gruvarbetaren och en av ett begränsat antal under mycket av Bitcoins tidiga historia. Tills Bitcoin mining tog fart och nätverkets hashhastighet började öka på grund av inflödet av andra gruvarbetare samlade Satoshi på sig en betydande del av blockbelöningarna, som då var 50 bitcoins var 10:e minut. När Bitcoins pris ökade, förvandlades detta till en liten förmögenhet, vid ett tillfälle värt flera miljarder dollar. Vi vet att dessa bitcoins inte har betalats ut. Det har de faktiskt aldrig gjort flyttats sedan den bröts. Alla kan se vilka Bitcoin-adresser förmodligen tillhör

Satoshi, och så om dessa mynt skulle säljas och intäkterna överförs till en viss bank
skulle det vara en mycket anmärkningsvärd händelse och en viktig ledtråd till Satoshis identitet. Så intressant nog,
även om Satoshi på pappret har gjort en avsevärd vinst på Bitcoin-brytning, kan Satoshi inte göra det
kontanter in den vinsten utan att identifiera sig själv, och det är något som, oavsett vad
skäl, Satoshi vill inte göra.

202

Sida 4

I en viktig mening spelar det ingen roll att vi inte känner till Satoshis identitet på grund av det anmärkningsvärda
särdrag hos Bitcoin att det är decentraliserat och med ingen enskild enhet ansvarig. Satoshi är inte ansvarig,
och till viss del spelar det egentligen ingen roll vad Satoshi tycker längre. Någon speciell påverkan som
Satoshi har är bara på grund av respekt som Satoshi skulle ha i Bitcoin-gemenskapen borde
Satoshi blir aktiv igen.

Tillväxt. Bitcoin har vuxit avsevärt sedan systemet togs i drift i januari 2009. Vi
kan se det i grafen för transaktionsvolymen (Figur 7.3) och i grafen för växelkursen (7.4),
även om topppriset genom tiderna, i april 2015, var tillbaka i slutet av 2013. Ibland har tillväxten
varit gradvis, men ibland har det förekommit hopp eller sprutor, ofta motsvarande nyhetsvärde
evenemang. Generellt sett har tillväxten accelererat över tid.

Figur 7.3: Marknadspriset för Bitcoin (7-dagars genomsnitt). Notera den logaritmiska skalan.
Källa: bitcoincharts.com.

Figur 7.4: Daglig transaktionsvolym (7-dagars genomsnitt). Källa: bitcoincharts.com.

203

Sida 5

7.5: Regeringar uppmärksammar Bitcoin

Resten av det här kapitlet handlar om regeringar – regeringsinteraktion med Bitcoin och försök att
reglera Bitcoin. Låt oss börja med det ögonblick då regeringar märkte Bitcoin, det vill säga när Bitcoin
blev ett tillräckligt stort fenomen för att regeringen började oroa sig för vilken inverkan det kan få
och hur man reagerar på det. I det här avsnittet och nästa kommer vi att diskutera varför regeringar kan oroa sig
för
Bitcoin specifikt. Sedan kommer vi i avsnitt 7.7 att vända oss till områden där Bitcoin-företag kan vara reglerade
av liknande skäl som andra typer av företag. Slutligen i avsnitt 7.8 ska vi titta på en fallstudie av en
föreslagen förordning som kombinerar inslag av regelbundet konsumentskydd med
Bitcoin-specifika aspekter.

Kapitalkontroll. En anledning till att regeringar skulle lägga märke till en digital valuta som Bitcoin är det
ospårbara digitala kontanter, om de finns, besegrar kapitalkontroller. Kapitalkontroll är regler eller lagar som a
land har på plats som är utformade för att begränsa flödet av kapital (pengar och andra tillgångar) in eller ut
landets. Genom att sätta kontroller på banker, investeringar och så vidare kan landet försöka reglera
dessa flöden.

Bitcoin är ett mycket enkelt sätt, under vissa omständigheter, att besegra kapitalkontroller. Någon kan helt enkelt köpa bitcoins med kapital inom landet, överföra dessa bitcoins utanför landet elektroniskt, och sedan byta ut dem mot kapital eller rikedom utanför landet. Det skulle låta dem flytta kapital eller rikedom från insidan till utsidan och på samma sätt kan de flytta kapital utifrån till insidan. Eftersom rikedom i denna elektroniska form kan röra sig så lätt över gränserna och kan inte riktigt kontrolleras, en regering som vill genomdriva kapitalkontroller i en värld med Bitcoin måste försöka koppla bort Bitcoin-världen från det lokala banksystemet för fiatvaluta. Det skulle göra det omöjligt för någon att omvandla stora mängder lokal valuta till Bitcoin, eller stora mängder Bitcoin till lokal valuta. Vi har verkligen sett länder som försöker skydda sina kapitalkontroller göra precis det, med Kina som en anmärkningsvärt exempel. Kina har engagerat sig i allt starkare åtgärder för att försöka koppla bort bitcoins från det kinesiska banksystemet för fiatvaluta genom att hindra företag från att byta bitcoins mot yuan.

Crime. En annan anledning till att regeringar kan oroa sig för ospårbara digitala kontanter är att de tjänar vissa typer av brott lättare — i synnerhet brott som kidnappning och utpressning som involverar betalning av lösen. De brotten blir lättare när betalning kan ske på distans och anonymt.

Brottsbekämpning mot kidnappare, till exempel, har ofta förlitat sig på att utnyttja överlämnandet av pengar från offret eller offrets familj till brottslingarna. När det kan göras på avstånd in på ett anonymt sätt blir det mycket svårare för brottsbekämpande myndigheter att följa pengarna. Annan Exempel: skadlig kod "CryptoLocker" krypterar offrens filer och kräver lösen i Bitcoin (eller annan typer av elektroniska pengar) för att dekryptera dem. Så brottet och betalningen genomförs båda vid en distans. Likaså blir skatteflykt lättare när det är lättare för människor att flytta runt pengar och att engagera sig i transaktioner som inte lätt är knutna till en viss individ eller identitet. Äntligen försäljningen

---

av illegala föremål blir potentiellt lättare när överföring av medel kan ske på distans och utan att behöva gå via en reglerad institution.

Silk Road. Ett bra exempel på det är Silk Road, en självutformad "anonym marknadsplats" som också har kallats "ebay för illegala droger". Figur 7.5 visar en skärmdump av Silk Roads hemsida när den verkade. Olagliga droger var de primära föremålen till salu, med en liten mängd andra kategorier som du kan se till vänster.

Silk Road tillät säljare att annonsera varor till försäljning och köpare att köpa dem. Varorna var levereras vanligtvis via post eller via frakttjänster och betalningen gjordes i bitcoins. Webbplatsen fungerade som en Tor dold tjänst, ett koncept som vi diskuterade i kapitel 6. Som du kan se i skärmbilden, sin adress var http://silkroadvb5piz3r.onion . Så här var serverns plats dold för brottsbekämpande myndigheter. På grund av användningen av bitcoins för betalning var det också svårt för lag verkställighet för att följa pengarna och ta reda på vilka personerna som deltog på marknaden var.

Figur 7.5: Skärmdump av Silk Roads webbplats (april 2012).

Sida 7

Silk Road höll bitcoins i deposition medan varorna skickades. Det fanns en innovativ deposition system som hjälpte till att skydda köpare och säljare mot fusk från andra parter. Bitcoins skulle släppas när köparen intygat att varan kommit fram. Det fanns också en eBay-liknande ryktesystem som gjorde det möjligt för köpare och säljare att få rykte om sig att följa deras affärer, och genom att använda det ryktesystemet kunde Silk Road ge marknadsaktörerna en incitament att följa reglerna. Så Silk Road var innovativ bland kriminella marknader när det gällde att hitta sätt att upprätthålla den kriminella marknadens regler på distans, vilket är något som kriminella marknadsför in det förflutna har haft svårt att göra.

Silk Road drevs av en person som kallade sig Dread Pirate Roberts - uppenbarligen en pseudonym, som ni kanske känner igen som en hänvisning till hjälte i romanen / film *The Princess Bride* . Den opererade från Februari 2011 till oktober 2013. Silk Road stängdes av efter gripandet av dess operatör Ross Ulbricht, som senare identifierades som Dread Pirate Roberts. Ulbricht hade försökt dölja sina spår driva pseudonyma konton och genom att använda Tor, anonyma remailers och så vidare. De Regeringen kunde ändå koppla ihop punkterna och koppla honom till Silk Road-aktiviteten — till servrar och bitcoins som han kontrollerade som operatör för Silk Road. Han dömdes för olika brott rör driften av Silk Road. Han åtalades även för mordförsök för uthyrning, dock lyckligtvis var han så inkompetent att ingen faktiskt blev dödad.

I samband med nedläggningen av Silk Road beslagtog FBI cirka 174 000 bitcoins, värda över 30 miljoner dollar just då. Som med intäkterna från alla brott enligt amerikansk lag, kan de beslagtas av regeringen. Senare auktionerade regeringen ut en del av de beslagtagna bitcoinsna.

Lektioner från Silk Road. Det finns flera lektioner från Silk Road och från mötet däremellan brottsbekämpning och Ulbricht. För det första är det ganska svårt att behålla den verkliga världen och den virtuella världen
separat. Ulbricht trodde att han kunde leva sitt riktiga liv i samhället och samtidigt ha en hemlighet identitet där han drev en betydande affärs- och teknologiinfrastruktur. Det är svårt att hålla dessa separata världar är helt isär, och skapar inte av misstag någon koppling mellan dem. Dess svårt att vara anonym under lång tid samtidigt som man är aktiv och engagerar sig i en samordnad kurs arbeta med andra människor över tid. Om det någonsin finns ett samband mellan de två identiteter - säg, om du glider upp och använder namnet på en medan du bär en annans mask - den länken kan aldrig förstöras och med tiden de olika anonyma identiteter eller mask som någon är försöker använda tenderar att bli ansluten. Det var precis vad som hände med Ulbricht - han gjorde några misstag tidigt när han använde samma datorer för att komma åt sina personliga konton och Dread Pirate Roberts konton och detta räckte så småningom för att utredarna skulle upptäcka hans offline-identitet.

En annan lärdom är att brottsbekämpande myndigheter kan följa pengarna. Redan före Ulbrichts arrestering, regeringen visste att vissa Bitcoin-adresser kontrollerades av operatören av Silk Road, och de tittade på adresserna. Resultatet är att Ulbricht, medan rik enligt blocket kedjan, kunde faktiskt inte dra nytta av den rikedomen eftersom alla försök att överföra dessa tillgångar till dollar skulle ha resulterat i en spårbar händelse, och förmodligen skulle ha resulterat i snabb

gripa. Så även om Ulbricht var ägare till ungefär 174 000 bitcoins, i den verkliga världen han levde inte som en kung. Han bodde i en lägenhet med ett sovrum i San Francisco medan han tydligen oförmögen att komma till den rikedom som han kontrollerade.

Kort sagt, om du tänker driva ett underjordiskt kriminellt företag - och det skulle vi uppenbarligen inte göra rekommenderar den här vägen — då är det mycket svårare att göra än du kanske tror. Teknik som Bitcoin och Tor är inte skottsäkra och brottsbekämpande myndigheter har fortfarande betydande verktyg till sitt förfogande. Fastän
Det har varit en del panik i brottsbekämpningsvärlden över uppkomsten av Bitcoin, det börjar de göra inse att de fortfarande kan följa pengarna upp till en viss punkt och att de fortfarande har en betydande förmåga att göra det
utreda brott och försvåra livet för människor som vill ägna sig åt samordnad brottslighet handling.

Samtidigt menar vi inte att antyda att brottsbekämpningen har stängt genom att ta ner Silk Road ned Bitcoin-baserade dolda marknader för illegala droger för gott. Faktum är att efter Silk Roads bortgång det har skett en ökning av sådana marknader. Några av de mer framträdande är får Marketplace, Silk Road 2, Black Market Reloaded, Evolution och Agora. De flesta av dessa är nu upphört, antingen på grund av brottsbekämpande åtgärder eller på grund av stöld, ofta av insiders. Dock forskning har funnit att den totala försäljningsvolymen bara har ökat, med brottsbekämpande åtgärder mot enskilda webbplatser inte nämnvärt bromsa tillväxten av denna underjordiska marknad. För att ta itu med säkerhetsrisk för att webbplatsoperatören försvinner med köparnas spärrade medel, de nyare marknadsplatserna använd multisignature-deponering (som vi såg i kapitel 3) snarare än Silk Roads modell för deponering medlen hos marknadsaktören.

7.6: Anti penningtvätt

I det här avsnittet kommer vi att titta på penningtvätt och Anti Money Laundering (AML) regler som regeringar har infört, särskilt i USA, som påverkar vissa Bitcoin-relaterade företag.

Målet med anti-penningtvättspolitiken är att förhindra att stora flöden av pengar passerar gränser eller röra sig mellan den underjordiska och legitima ekonomin utan att bli upptäckt. Vi tittade tidigare vid kapitalkontroller som finns för att hindra pengar från att passera gränser. I vissa fall är länder rättvisa bra med pengar som passerar gränserna, men de vill veta vem som överför vad till vem och var de pengarna kom ifrån.

Anti-penningtvätt syftar till att försöka försvåra vissa typer av brott, särskilt organiserad brottslighet. Organiserade brottsgrupper finner ofta att de får in mycket pengar ett ställe och vill flytta det någon annanstans, men vill inte förklara var de pengarna kom från — därav önskan att få pengar över gränserna. Eller så kanske de kommer på sig själva att göra en mycket pengar i en underjordisk ekonomi och vill få in de pengarna i den legitima ekonomin så att de kan spendera det på sportbilar och stora hus eller vad det nu är som ledarna för gruppen vill göra. Anti-penningtvätt har alltså som mål att göra det svårare att flytta runt pengar detta sätt och gör det lättare att fånga folk som försöker göra det.
207

Lär känna din kund. En av de väsentliga motåtgärderna mot penningtvätt är något kallas Know Your Customer-lagar, ibland kallade KYC. Detaljerna kan vara lite komplicerade och kommer beror på din plats, men grundidén är denna: Känn din kundregler kräver vissa typer av företag som hanterar pengar gör tre saker:

1. *Identifiera och autentisera klienter* - få någon form av autentisering som kunderna verkligen som de hävdar att de är det och att de påstådda identiteterna motsvarar någon form av verklig värld identitet. Så en person kan inte bara gå in och de är John Smith från 123 Main Street in AnyTown, USA — de måste tillhandahålla tillförlitliga identifieringsdokument.
2. *Utvärdera risken för kunden* - bestämma risken för en viss kund delta i underjordiska aktiviteter. Detta kommer att baseras på hur kunden beter sig - hur länge deras verksamhet har varit relationen är med företaget, hur välkända de är i samhället och olika andra faktorer. KYC-regler kräver i allmänhet att företag som omfattas av behandling behandlar kunder vars verksamhet verkar mer riskfylld med mer uppmärksamhet.
3. *Se upp för avvikande beteende* - det vill säga, ett beteende som verkar vara ett tecken på pengar tvättning eller kriminell verksamhet. KYC kräver ofta att ett företag avbryter affärer med en klient som ser tvivelaktig ut eller som inte kan autentisera sig själv eller sina aktiviteter tillräckligt för regeln.

Obligatorisk rapportering. Det finns obligatoriska rapporteringskrav i USA värt att prata om. Företag inom ett brett spektrum av sektorer måste rapportera valutatransaktioner som är över $10 000. De måste lämna in vad som kallas en valutatransaktionsrapport för att säga vad transaktionen är och vem den andra parten i transaktionen är. Det finns också vissa krav på autentisera vem den parten är. När den väl har rapporterats går informationen in i statliga databaser och kan sedan analyseras för att leta efter beteendemönster som tyder på penningtvätt.

Företag måste också se efter kunder som kan "strukturera" transaktioner för att undvika rapportering, som att delta i en serie transaktioner på 9 999 USD för att komma runt rapporteringsregeln på 10 000 USD.

Företag som ser bevis på strukturering måste rapportera det genom att lämna in en misstänkt aktivitetsrapport. Om igen, informationen går in i en statlig databas och kan leda till utredning av klienten.

Kraven här skiljer sig avsevärt från land till land. Vi försöker inte på något sätt ge dig laglig råd om huruvida du behöver detta eller vad du måste göra. Denna diskussion är tänkt att ge dig en uppfattning om vilken typ av krav som ställs av regler mot penningtvätt. Som sagt, ta Observera att regeringar – i USA och andra länder – tenderar att ta regler mot penningtvätt mycket, mycket allvarligt med stora straffrättsliga straff för kränkningar. Det här är inte den typen av regler som du kan bara blåsa av och ta tag i om man får ett klagomål från regeringen senare.

Bitcoin-företag har stängts ner - ibland tillfälligt, ibland permanent. Företag människor har arresterats och människor har hamnat i fängelse för att de inte följt dessa regler. Det här är ett område där regeringen kommer att tillämpa lagen kraftfullt, oavsett om fiat-valuta eller Bitcoin är det Begagnade. Regeringen har upprätthållit dessa lagar mot Bitcoin-baserade företag ända sedan de märkte det att Bitcoin var tillräckligt stort för att utgöra en risk för penningtvätt. Om du är intresserad av att starta någon

typ av verksamhet som kommer att hantera stora volymer valuta, måste du prata med en advokat som förstår dessa regler.

7.7: Reglering

Låt oss nu direkt ta upp ordet "R" - förordning. Reglering får ofta ett dåligt rykte, särskilt bland den typ av människor som tenderar att gilla Bitcoin. Som argumentet går, är reglering en del en byråkrat som inte kan min verksamhet eller vad jag försöker göra, kommer in och stökar till saker. Det är en börda. Det är dumt och meningslöst. Detta argument är vanligt och väl förstått, och medan det är det ofta åtminstone delvis korrekt, vi kommer inte att upprepa det här.

Istället kommer vi i det här avsnittet att titta lite i detalj på orsaker till varför regler ibland kan vara det motiverat, eftersom det argumentet inte är lika väl uppfattat. För att vara tydlig, det faktum att vi spenderar det mesta av det här avsnittet som talar om varför reglering kan vara bra bör inte läsas som ett stöd omfattande reglering. Det är helt enkelt så att vi vill få lite mer balans i diskussionen i ett samhälle där reglering ofta anses vara i sig dålig.

Det viktigaste argumentet för reglering är detta: när marknader misslyckas och ger resultat som är dåliga - och överens om att vara dåliga av i stort sett alla på marknaden - då kan reglering träda in och försöka åtgärda misslyckandet. Så argumentet för reglering, när det finns ett argument, börjar med tanken att marknader inte alltid ger dig det resultat du vill ha.

Låt oss göra detta lite mer exakt, med hjälp av termer från ekonomi. Det som oroar oss är ett marknadsmisslyckande,
och med det menar vi inte bara att något dåligt händer eller att någon känner att de är det bli lurad eller orättvist behandlad. Vi menar att det finns en alternativ allokering av varor till marknadsaktörer som skulle leda till *att alla* är bättre, eller åtminstone inte sämre. Vilken alternativ allokering kallas en *Pareto förbättring* .

Citronmarknad. Låt oss diskutera ett sätt på vilket marknaden kan misslyckas, ett klassiskt exempel som kallas citronmarknaden. Namnet har sitt ursprung i samband med att sälja bilar. Låt oss säga att alla bilar är någon av låg kvalitet eller hög kvalitet (med inget emellan). En bil av hög kvalitet kostar lite mer tillverkning än en bil av låg kvalitet, men det är mycket, mycket bättre för konsumenten som köper den.

Om marknaden fungerar väl (om det är *effektivt* som ekonomerna kallar det) det kommer att leverera mestadels hög kvalitet
bil till konsumenterna. Det beror på att även om den högkvalitativa bilen är lite dyrare, är de flesta konsumenter föredrar det och är villiga att betala mer för det. Så under vissa antaganden kommer en marknad att tillhandahålla detta
lyckligt resultat.

Å andra sidan, låt oss anta att kunderna inte kan skilja bilar av låg kvalitet från bilar av hög kvalitet. A låg kvalitet bil (en *citron*) sitter på mycket kan se ganska bra, men du kan inte riktigt säga om det kommer att gå sönder imorgon eller om det ska köra på länge. Återförsäljaren vet förmodligen om det är en citron, men du som kund kan inte se skillnad.

Låt oss tänka på de incitament som driver människor på den här typen av citronmarknad. Som konsument är du det
inte villig att betala extra för en högkvalitativ bil, eftersom du helt enkelt inte kan se skillnad. Även om begagnad bilhandlare säger att en bil är perfekt och kostar bara en extra hundra dollar, du har inte en bra anledning att lita på återförsäljaren.

Till följd av detta kan producenterna inte tjäna några extra pengar på att sälja en högkvalitativ bil. Faktum är att de förlora pengar på att sälja en bil av hög kvalitet eftersom den kostar lite mer att tillverka och de inte får någon prispåslag. Så marknaden fastnar i en jämvikt där endast bilar av låg kvalitet tillverkas,
och konsumenterna är relativt missnöjda med dem.

Detta resultat är värre för alla än en väl fungerande marknad skulle vara. Det är värre för
köpare eftersom de måste nöja sig med bilar av låg kvalitet. På en mer effektiv marknad kunde de
har köpt en bil som var mycket, mycket bättre för ett lite högre pris. Det är också värre för producenterna
— eftersom bilarna som finns på marknaden alla är usla köper konsumenterna inte så många bilar som de kanske annars, så det finns mindre pengar att tjäna på att sälja bilar än vad det skulle vara på en sund marknad.

Det är ett marknadsmisslyckande. Detta specifika exempel är inte i sig beroende av bilar. Något bra (r "widget") till salu som lider av "asymmetrisk information" där antingen säljare eller köpare är
har mycket bättre information om kvaliteten på varan kan leda till ett marknadsmisslyckande. Den här typen av marknadsmisslyckanden kallas en citronmarknad, även om den ekonomiska litteraturen ger många fler exempel.

Att fixa en citronmarknad. Det finns några marknadsbaserade metoder som försöker fixa en citronmarknad. Den första bygger på säljarens rykte. Tanken är att om en säljare konsekvent berättar sanningen för konsumenterna om vilka widgetar som är hög respektive låg kvalitet, då kan säljaren få ett rykte om att berätta
sanning. När de väl har det ryktet kan de kanske sälja widgets av hög kvalitet till ett högre pris
eftersom konsumenterna kommer att tro på dem och därför kan marknaden fungera mer effektivt.

Detta fungerar ibland och ibland inte beroende på de exakta antaganden du gör om
marknaden. Naturligtvis kommer det aldrig att fungera så bra som en marknad där konsumenterna faktiskt kan berätta
skillnad i kvalitet. Dels tar det ett tag för en producent att bygga upp ett gott rykte. Den där
innebär att de måste sälja widgets av hög kvalitet till låga priser ett tag tills konsumenterna lär sig det
de talar sanning. Det gör det svårare för en ärlig säljare att komma in på marknaden.

Det andra potentiella problemet är att en säljare, även om de har varit ärliga fram till nu, inte längre har incitament att vara ärlig om de vill ta sig ur marknaden (säg om deras försäljning minskar). I det
deras incitament är att massivt lura människor på en gång och sedan lämna marknaden. Rykte alltså
fungerar inte bra varken i början eller slutet av en säljares närvaro på marknaden.

Ett ryktebaserat tillvägagångssätt tenderar inte heller att fungera i företag där konsumenterna inte upprepar sig verksamhet med samma enhet, eller där produktkategorin är mycket ny och därför inte har gjort det

---

funnits tillräckligt med tid för säljare att bygga upp ett rykte. En högteknologisk marknad som Bitcoin-börser

lider av just dessa problem.

Den andra marknadsbaserade metoden är garantier. Tanken är att en säljare kan ge en garanti till en köpare som säger att om widgeten visar sig vara av låg kvalitet, kommer säljaren att tillhandahålla ett utbyte eller en återbetalning. Det kan fungera bra upp till en punkt, men det finns också ett problem: en garanti är bara en annan typ av produkt som även kan komma i högkvalitativa eller lågkvalitativa versioner! En lågkvalitetsgaranti är en där säljaren inte riktigt kommer igenom när du kommer tillbaka med den trasiga produkten. De avstå från sitt löfte eller så får de dig att hoppa igenom alla typer av ringar.

Regulatoriska korrigeringar. Så om en citronmarknad har utvecklats och om dessa marknadsbaserade tillvägagångssätt inte gör det arbeta för den specifika marknaden, då kanske reglering kan hjälpa. Specifikt finns det tre sätt på vilka reglering skulle kunna lösa problemet.

För det första kan reglering kräva offentliggörande. Det kan till exempel kräva att alla widgetar märks som höga kvalitet eller låg kvalitet, kombinerat med påföljder för företagen för att ljuga. Det ger konsumenterna information om att de saknades. En andra strategi för reglering är att ha kvalitetsstandarder så att ingen widget kan säljas om den inte uppfyller någon standard för kvalitetstestning, med den standarden inställd så att endast widgets av hög kvalitet klarar testet. Det skulle resultera i en marknad som återigen bara har en typ av widget, men det är åtminstone högkvalitativa widgets, förutsatt att regleringen fungerar som avsett.
Det tredje tillvägagångssättet är att kräva att alla säljare utfärdar garantier och sedan genomdriva driften av dessa garantier så att säljarna hålls till de löften de ger.

Någon av dessa former av reglering kan uppenbarligen misslyckas - det kanske inte fungerar som det är tänkt, kanske felskriven eller felanvänd, eller kan vara betungande för säljare. Men det finns åtminstone en möjlighet reglering av denna typ kan hjälpa till att åtgärda marknadsmisslyckandet på grund av en citronmarknad. Människor som argumentera för reglering av Bitcoin-utbyten, till exempel, peka ibland på dem som ett exempel på en citronmarknaden.

Samverkan och antitrustlagar. Ytterligare ett exempel på marknader som inte fungerar som vi skulle vilja ha dem till är prissättning. Prissättning är när olika säljare samarbetar med varandra och kommer överens om att höja priserna eller att inte sänka dem. En relaterad situation är där företag som annars skulle gå in konkurrens med varandra är överens om att inte tävla. Till exempel om det fanns två bagerier i stan de kanske kommer överens om att en av dem bara kommer att sälja muffins och den andra bara kommer att sälja bagels, och det så det är mindre konkurrens mellan dem än om de både sålde muffins och bagels.
Som ett resultat av den minskade konkurrensen går antagligen priserna upp, och handlarna kan hämma marknadens funktion.

När allt kommer omkring är anledningen till att marknaden skyddar konsumenterna väl i sin normala drift genom konkurrensfordon. Säljare måste konkurrera för att kunna erbjuda de bästa varorna till det bästa priset konsumenter, och om de inte konkurrerar på det sättet kommer de inte att få affärer. Ett avtal att fixa

priser eller att inte konkurrera kringgår den konkurrensen. När människor vidtar åtgärder som förhindrar konkurrens, det är en annan typ av marknadsmisslyckande.
211

Den här typen av avtal – för att höja priserna eller att inte konkurrera – är olagliga i de flesta jurisdiktioner. Detta är en del av antitrustlagstiftningen eller konkurrenslagstiftningen. Målet med denna lagsamling är att förhindra avsiktliga handlingar
som förhindrar eller skadar konkurrensen. Mer generellt begränsar det andra åtgärder än att bara erbjuda gott produkter till bra priser, såsom försök att minska konkurrensen genom sammanslagningar. Antitrustlag är mycket komplicerat och vi har bara gett dig en skiss av det, men det är ett annat exempel på hur marknaden kan misslyckas och hur lagen kan och kommer att träda in för att förhindra det.

7.8: New Yorks BitLicense-förslag

Hittills har vi diskuterat reglering generellt: olika former av reglering, varför reglering kan vara det motiverat i vissa fall och kan vara ekonomiskt vettigt. Låt oss nu övergå till en specifik insats av en specifik stat för att införa specifik reglering av Bitcoin, nämligen New York State BitLicense-förslag. Informationen här är aktuell från början av 2015, men landskapet för Bitcoin-reglering förändras snabbt. Det spelar inte så stor roll för våra syften, för vårt mål är inte så mycket att hjälpa dig förstå en specifik del av faktisk eller föreslagen reglering. Snarare vill vi hjälpa dig att förstå vilken typ av saker tillsynsmyndigheter gör och ger dig en känsla av hur de tänker kring problemet.

BitLicense-förslaget utfärdades i juli 2014 och har sedan dess reviderats som svar på kommentarer från Bitcoin-gemenskapen, industrin, allmänheten och andra intressenter. Den utfärdades av New York State Department of Financial Services, den del av delstaten New York som reglerar finansiell industri. Naturligtvis har delstaten New York världens största finansiella centrum, och så är det en del av delstatsregeringen som är van vid att hantera relativt stora institutioner.

Vem omfattas. BitLicense är en föreslagen uppsättning koder, regler och förordningar som har att göra med virtuella
valutor. I grund och botten står det att du skulle behöva skaffa något som kallas en BitLicense från den nya York Department of Financial Services om du vill göra något av det som anges i rutan nedan:

Affärsaktivitet med virtuell valuta avser utförandet av någon av följande typer av aktiviteter som involverar New York eller en New York-invånare:
1. ta emot virtuell valuta för överföring eller sända virtuell valuta, utom där transaktionen genomförs i icke-finansiella syften och innebär inte överföring på mer än ett nominellt belopp av virtuell valuta;
2. lagring, innehav eller upprätthållande av förvar eller kontroll av virtuell valuta för andras räkning;
3. köpa och sälja virtuell valuta som kundföretag;
4. utföra Exchange Services som kundföretag; eller
5. kontrollera, administrera eller utfärda en virtuell valuta.
Utveckling och spridning av programvara utgör i sig inte virtuellt
Valuta Affärsverksamhet.

Texten hänvisar till "aktiviteter som involverar New York eller en New York-invånare", vilket återspeglar regelverket
NYDFS myndighet. Ändå sträcker sig effekterna av bestämmelser som dessa långt bortom gränserna stat, av två skäl. För det första, för stater med betydande befolkningar som New York eller Kalifornien, ställs inför valet mellan att följa statliga lagar och att inte göra affärer med konsumenter i dessa stater kommer de flesta företag att välja att följa. För det andra uppfattas vissa stater allmänt som ledare i att reglera vissa ekonomiska sektorer — finans i fallet med New York, teknologi i fallet med Kalifornien. Det betyder att andra amerikanska stater ofta följer den riktning som de anger.

Lägg märke till undantaget för icke-finansiella användningar i den första kategorin – detta lades till i den andra revidering, och den är bra. Det är ett urval för just den typen av Bitcoin-som-en-plattform-applikationer som vi kommer att titta på med början i kapitel 9. Den andra kategorin kan täcka saker som plånbokstjänster. Som för den tredje kategorin verkar det som att du kan köpa och sälja bitcoins för dig själv, men gör det som en kundföretag kräver en BitLicense. Den fjärde kategorin är självförklarande. Den sista kanske tillämpas mer på altcoins, av vilka många är något centraliserade, än på Bitcoin. Vi ska titta på altcoins i kapitel 10.

Undantaget för mjukvaruutveckling i slutet är återigen ett viktigt sådant. Språket fanns inte med originalversionen, och det kom ett ramaskri från samhället. NYDFS-intendent Benjamin Lawsky klargjorde strax efter att avsikten inte var att reglera utvecklare, gruvarbetare eller individer som använder Bitcoin. Den andra versionen innehåller det explicita språket ovan.

Krav. Om förordningen träder i kraft och du är en av de täckta enheterna måste du ansöka om licens. För att ansöka om en licens finns det ett detaljerat språk i förslaget som du kan läsa, men grovt sett måste du ge information om ägandet av ditt företag, om ditt ekonomi och försäkringar på din affärsplan - i allmänhet för att NYDFS ska kunna veta vem du är, hur välbackad du är, var dina pengar kommer ifrån och vad du planerar att göra. Och du måste betala en ansökningsavgift.

Om du får en licens måste du lämna uppdaterad information till NYDFS om de saker vi listade: ägande, ekonomi, försäkringar och så vidare. Du måste tillhandahålla periodiska bokslut så att de kan hålla reda på hur du har det ekonomiskt. Du skulle behöva upprätthålla en ekonomisk reserv, vars belopp kommer att fastställas av NYDFS baserat på olika faktorer om ditt företag.

Det finns detaljerade regler om saker som hur du skulle behålla vårdnaden om konsumenttillgångar. Det finns regler mot penningtvätt som kan eller inte går utöver vad som redan krävs enligt befintliga lagar. Det finns regler om att ha en säkerhetsplan och penetrationstestning och så vidare. Det finns regler om katastrofåterställning — du måste ha en katastrofåterställningsplan som uppfyller olika kriterier. Det finns regler om registerföring - du måste föra register och göra dem tillgängliga för NYDFS under vissa omständigheter. Du måste ha skrivna policyer om efterlevnad och det har du att utse en efterlevnadsansvarig – någon inom din organisation som är ansvarig för efterlevnaden och har det ansvar och befogenhet som krävs. Det finns ett krav som du avslöjar risk för konsumenter, så att konsumenterna förstår riskerna med att göra affärer med dig.

Som du kan se är kraven betydande och de är analoga med den typen av krav

för en värdepappersfond eller en börsnoterad aktie. NYDFS måste fortfarande besluta om vad de ska göra med förslaget
— om den ska dras tillbaka, utfärda den i dess nuvarande form eller göra ytterligare ändringar. Tillsammans med det
beslut de kommer att utfärda något slags dokument som ger skälen till vad de beslutat att göra.

Om något som BitLicense träder i kraft, skulle det vara ett stort steg i Bitcoins historia. Du skulle ha en situation där inte bara NYDFS, utan kanske andra jurisdiktioner skulle börja träda in och reglera, och du skulle börja se Bitcoin-företag börja komma närmare den traditionella modellen av reglerade finansinstitut.

Detta skulle vara ett steg som på något sätt strider mot cypherpunk eller cypher-libertarian idéer om vad Bitcoin borde vara, men å andra sidan finns det en viss oundviklighet att så snart som Bitcoin blev riktigt värdefulla, Bitcoin-företag blev stora företag och regeringen blev intresserad, reglering skulle följa. Bitcoin-företag berör verkliga människor och fiat-valutaekonomin. Om Bitcoin är tillräckligt stor för att spela roll, då är den tillräckligt stor för att bli reglerad. Det representerar en reträtt från vad som
ursprungliga förespråkare för Bitcoin hade i åtanke, men på ett annat sätt representerar det Bitcoins ekosystem växa upp och integreras i den vanliga ekonomin som är mycket mer reglerad. Oavsett din hållning till det, reglering börjar ske, och om du är intresserad av att starta en Bitcoin företag måste du vara uppmärksam på denna trend.

Kommer detta att bli en framgång? Det finns olika sätt att se på det, men här är ett sätt att utvärdera effektiviteten av reglering som BitLicense med avseende på det allmänna politiska målet att förbättra kvalitet på Bitcoin-företag: om något som BitLicense träder i kraft och om företag startar annonsera till kunder utanför New York att de kan lita på eftersom de har en BitLicense, och om det argumentet är övertygande för konsumenterna när de väljer ett företag att göra affärer med, då kommer regleringen att fungera på det sätt som dess förespråkare ville ha det. Om det kommer att hända och hur det kommer att påverka framtiden för Bitcoin är något som vi får vänta och se.

Vidare läsning

Två papper som innehåller många intressanta detaljer om hur Silk Road och dess efterföljare har fungerat:

. Christin, Nicolas resa Sidenvägen: En mätning analys av ett stort anonym på nätet marketplace . Proceedings of the 22nd International Conference on World Wide Web, 2013.

Soska, Kyle och Christin, Nicolas. Mäta Longitudinal Utvecklingen av Anonym Online Marknads Ecosystem . Proceedings of the 24th USENIX Security Symposium, 2015.

En guide till de regulatoriska frågor som Bitcoin väcker:

. Brito, Jerry och Andrea Castillo Bitcoin: en primer för beslutsfattarna . Mercatus Center i George Mason University, 2013.

En bok som tar upp historien om modern kryptografi och cypherpunkrörelsen, som ger lite intuition för Bitcoins tidiga politiska rötter:

. Levy, Steven *Crypto: hur koden Rebels Vispa regeringen spar Privacy i den digitala eran* . Penguin, 2001.

En populär utställning av tidigt arbete med digitala kontanter, kombinerat med en vision för en värld med digitala Integritet:

Chaum, David. Security utan identifikation: transaktionssystem för att göra storebror föråldrade . Kommunikation från ACM, 1985.

En undersökning av informationssäkerhetens ekonomi som diskuterar flera orsaker till marknadsmisslyckanden:

Anderson, Ross, och Tyler Moore. Ekonomin i informationssäkerhet . Science 314, nr. 5799, 2006.

En diskussion om Bitcoin-specifika ekonomiska frågor och regleringsalternativ:

Böhme, Rainer, Nicolas Christin, Benjamin Edelman och Tyler Moore. Bitcoin: Economics, Teknik och styrning . The Journal of Economic Perspectives 29, nr. 2, 2015.

Texten i BitLicense-förslaget:

New York State Department of Financial Services föreskrifterna för föreståndare för Financial Tjänster. Del 200: virtuella valutor (reviderad), 2015.

Kapitel 8: Alternativa gruvpussel

Gruvpussel är själva kärnan i Bitcoin eftersom deras svårighet begränsar förmågan hos en part att kontrollera konsensusprocessen. Eftersom Bitcoin-gruvarbetare tjänar belöningar för pussel som de löser, vi förväntar oss att de kommer att lägga ner stor ansträngning på att försöka hitta tillgängliga genvägar för att lösa dessa pusslar snabbare eller mer effektivt, i hopp om att öka sin vinst. Å andra sidan, om det finns arbete som hjälper nätverket men som inte direkt hjälper dem att lösa pussel snabbare, kan gruvarbetare vara det incitament att hoppa över det för att minimera sina kostnader. Så designen av pusslet spelar en viktig roll i styra och vägleda deltagande i nätverket.

I det här kapitlet kommer vi att diskutera en mängd möjliga alternativa pusseldesigner, förutsatt att vi kan modifiera Bitcoins pussel eller till och med designa ett nytt pussel från grunden. En klassisk designutmaning har varit att göra ett pussel som är ASIC-resistent, utjämna spelplanen mellan användare med vanliga datorutrustning och användare med optimerad anpassad hårdvara. Vad mer kan vi designa pussel att uppnå? Vilka andra typer av beteenden skulle vi vilja uppmuntra eller motverka? Väl prata om några exempel med olika intressanta egenskaper, från minskande energiförbrukning

att ha några socialt användbara bieffekter för att motverka bildandet av gruvpooler. Några av dessa används redan av altcoins, medan andra är forskningsidéer som kan visa sig användas i framtiden.

8.1 Viktiga pusselkrav

Vi börjar med att titta på några väsentliga säkerhetskrav för gruvpussel. Det gör oss ingenting bra att introducera snygga nya funktioner om pusslet fortfarande inte uppfyller de grundläggande kraven som det måste hålla Bitcoin säker.

Det finns många möjliga krav, av vilka några har vi pratat om i kapitel 2 och 5. Gruvpussel måste vara snabba att verifiera eftersom varje nod i nätverket validerar varje pussel lösning — även noder som inte är direkt involverade i gruvdrift, inklusive SPV-klienter. Det skulle vi också gillar att ha justerbar svårighetsgrad så att pusslets svårighetsgrad kan ändras över tid som ny användare kommer in i nätverket med ökande mängder hashkraft tillfört. Detta möjliggör pusslet att vara svårt nog att attacker mot blockkedjan är kostsamma, men pussellösningar finns fortfarande på en ganska jämn takt (ungefär en gång var tionde minut i Bitcoin).

Vad exakt är Bitcoins gruvpussel? Hittills har vi bara kallat det "Bitcoins pussel." Mer exakt kan vi kalla det en *partiell hash-preimage pussel* , eftersom målet är att hitta preimages för en delvis specificerad hash-utgång — nämligen en utgång under ett visst målvärde. Några andra sällsynta egenskap kan också fungera, som att hitta ett block vars hash har minst k bitar inställda på noll, men att jämföra resultatet med ett mål är förmodligen det enklaste.

Det är lätt att se hur Bitcoins SHA-256 hashbaserade gruvpussel redan uppfyller dessa två krav. Den kan göras godtyckligt svårare genom tweaking en enda parameter (det *målet*). Att kontrollera lösningar är trivialt, kräver bara en enda SHA-256-beräkning och en jämförelse, oavsett hur svårt pusslet var att lösa.

. *Progress-freeness* En annan central krav är mer subtil: chansen att vinna ett pussel lösning i valfri tidsenhet bör vara ungefär proportionell mot hashkraften som används. Detta innebär att riktigt stora gruvarbetare med mycket kraftfull hårdvara borde bara ha proportionell fördel i att vara den nästa gruvarbetare för att hitta en pussellösning. Även små gruvarbetare bör ha en viss proportionell chans att lyckas och få ersättning.

För att illustrera denna punkt, låt oss tänka på ett dåligt pussel som inte uppfyller detta krav. Överväg a gruv pussel som tar exakt n steg för att hitta en lösning. Till exempel istället för att hitta ett block vars SHA-256-hash är under ett visst mål, kunde vi kräver beräkning n konsekutiva SHA-256 hash. Detta skulle inte vara effektivt att kontrollera, men strunt i det för nu. Det större problemet här är att eftersom det tar exakt n steg för att hitta en lösning, då den snabbaste gruvarbetare i nätverket kommer alltid vara den som vinner nästa belöning. Det skulle snart bli klart vilken gruvarbetare som löste varje pussel, och andra gruvarbetare skulle inte ha något incitament att delta alls.

Återigen, ett bra pussel ger varje gruvarbetare chansen att vinna nästa pussellösning i proportion till

mängden hashkraft de bidrar med. Föreställ dig att kasta en pil på en bräda slumpmässigt, med olika storleksmål som motsvarar gruvkraften som innehas av olika gruvarbetare. Om du tänker om det betyder detta krav att oddsen för att lösa pusslet måste vara oberoende av hur mycket arbete du redan har lagt ner på att försöka lösa det (eftersom stora gruvarbetare alltid har lagt ner mer arbete). Det är därför en bra brytning pussel kallas *framsteg fritt* .

Ur ett matematiskt perspektiv, innebär detta att en god gruv pussel måste vara en *minnesfri process* - allt annat skulle oundvikligen belöna gruvarbetare för tidigare framsteg på något sätt. Därför någon genomförbara pussel kommer i sig att involvera någon form av trial-and-error. Det är dags att hitta en lösning bilda därför oundvikligen en exponentiell fördelning som vi såg i kapitel 2.

Justerbar svårighet, snabb verifiering och framstegsfrihet är tre avgörande egenskaper hos Bitcoin gruvpussel. SHA-256-baserad partiell pre-image-fynd uppfyller verkligen alla tre. Vissa människor argumentera för att andra egenskaper som Bitcoins gruvpussel uppfyller också är viktiga, men vi kommer att diskutera
andra potentiella krav när de dyker upp medan vi utforskar andra potentiella funktioner.

8.2 ASIC-resistenta pussel

Vi börjar med utmaningen att utforma en *ASIC-resistent* pussel, som har varit den i särklass brett diskuterad och eftertraktad typ av alternativ gruvpussel. Som vi diskuterade i kapitel 5, Bitcoin-brytning skedde till en början främst med vanliga datorer, så småningom utvidgades till GPU:er och anpassade FPGA-enheter, och görs nu nästan uteslutande av mycket kraftfulla optimerade ASIC-chips.

Dessa ASIC:er är så mycket effektivare än allmän datorutrustning som gruvdrift med en vanlig dator (eller till och med några tidig generation ASIC) är inte längre värt elpriset, även om hårdvaran är gratis.

Denna övergång har inneburit att de flesta individer som deltar i Bitcoins ekosystem (till exempel kunder eller handlare som gör transaktioner med Bitcoin) har inte längre någon roll i gruvprocessen. Vissa människor tycker att detta är en farlig utveckling, med en mindre grupp professionella gruvarbetare kontrollera gruvprocessen. I Satoshi Nakamotos ursprungliga papper om Bitcoin, frasen "one-CPU-one-vote" användes, vilket ibland har uppfattats som att Bitcoin borde vara en demokratiska system som ägs av alla dess användare.

Andra anser att uppkomsten av ASIC är oundviklig och inte till nackdel för Bitcoin, och att önskan om ASIC-resistens är helt enkelt människor som vill gå tillbaka till "den gamla goda tiden." Utan att ta parti om ASIC-resistens är önskvärt kan vi dyka in i de tekniska utmaningarna och några av de föreslagna metoder för att uppnå detta mål.

Vad gör ASIC-motstånd medelvärde? Generellt sett vill vi disincentivize användningen av specialbyggd hårdvara för gruvdrift. Att tolka detta strikt skulle innebära att designa ett pussel för vilket befintliga datorer för allmänt bruk är redan de billigaste och mest effektiva enheterna. Men det här skulle vara omöjligt. När allt kommer omkring har datorer för allmänna ändamål redan speciella ändamål optimeringar. Alla produkter har inte samma optimeringar och de förändras med tiden. Till exempel, under det senaste decenniet har både Intel och AMD lagt till stöd för speciella instruktioner (ofta kallade "lägga till hårdvarustöd") för att beräkna AES-blockchifferet mer effektivt. Så några datorer

kommer alltid att vara mindre effektiva än andra vid gruvdrift. Dessutom är det svårt att föreställa sig att designa en gruvdrift
pussel som skulle förlita sig på funktioner som högtalare och skärm som de flesta är personliga datorer innehåller. Så specialmaskiner som tagits bort från dessa funktioner skulle förmodligen fortfarande vara det
billigare och effektivare.

Så i verkligheten är vårt mål ett mer blygsamt mål: att komma på ett pussel som minskar klyftan mellan den mest kostnadseffektiva anpassade hårdvaran och vad de flesta allmänna datorer kan göra. ASIC:er kommer oundvikligen att vara något mer effektiva, men om vi kunde begränsa detta till en storleksordning eller
mindre kan det fortfarande vara ekonomiskt för enskilda användare att bryta med de datorer de redan har.

Minneshårda pussel. De mest använda pussel som är utformade för att vara ASIC säkra kallade *minneshårda* pussel - pussel som kräver en stor mängd minne för att beräkna, i stället för, eller utöver mycket CPU-tid. Ett liknande men annat koncept är *minnesbundna* pussel i vilka tiden för åtkomst till minne dominerar den totala beräkningstiden. Ett pussel kan bara vara minnesvärt utan att vara minnesbunden, eller minnesbunden utan att vara minneshård, eller båda. Det är en subtil men viktig distinktion som härrör från det faktum att även om CPU-hastighet är flaskhalsen för *beräkningstiden* är *kostnaden* för att lösa ett stort antal sådana pussel parallellt kan fortfarande vara domineras av kostnaden för minne, eller vice versa. Typiskt för ett beräkningspussel vi vill ha något som är minneshårda *och* minne bundna, vilket säkerställer att en stor mängd av minnet är krävs och detta är den begränsande faktorn.

Varför kan minneshårda och minnesbundna pussel hjälpa ASIC-motstånd? De logiska operationerna som krävs för att beräkna moderna hash-funktioner är bara en liten del av vad som händer i en CPU, alltså att för Bitcoins pussel får ASIC: er mycket körsträcka genom att inte implementera något av det onödiga funktionalitet. En relaterad faktor är att variationen i minnesprestanda (och kostnad per enhet av prestanda) är mycket lägre än variationen i beräkningshastigheter mellan olika typer av processorer. Så om vi kunde designa ett pussel som var minneshårt, som kräver relativt enkelt beräkning men mycket minne att beräkna, betyder detta att kostnaden för att lösa ett pussel skulle göra det förbättras i långsammare takt med förbättringar av minneskostnaden.

SHA-256 är definitivt inte minneshård, som vi har sett, och kräver bara ett litet 256-bitars tillstånd som lätt passar in i CPU-register. Men det är inte så svårt att designa ett minneshårt arbetsprovspussel.

Scrypt . Det mest populära minneshårda pusslet kallas scrypt. Detta pussel används redan flitigt i Litecoin, den näst mest populära kryptovalutan, och en mängd andra Bitcoin-alternativ.

Scrypt är en minneshård hashfunktion, ursprungligen designad för att hasha lösenord på ett sätt som är svårt att brute-force, så gruvpusslet är samma Bitcoins partiella hash-preimage pussel förutom med scrypt som ersätter SHA-256.

Det faktum att scrypt existerade före Bitcoin och har använts för lösenordshashning ger en del förtroende för dess säkerhet. Lösenordshashing har ett liknande mål med ASIC-resistens, på grund av säkerheten vi vill att en angripare med anpassad hårdvara inte ska kunna beräkna lösenords-hashar mycket

snabbare än den legitima användaren eller servern, som förmodligen bara har datorer för allmänt bruk.

Scrypt fungerar i princip i två steg. Det första steget innebär att fylla en stor buffert av direktåtkomst minne (RAM) med slumpmässiga data. Det andra steget innebär att läsa från (och uppdatera) detta minne i en pseudoslumpmässig ordning, vilket kräver att hela bufferten lagras i RAM.

Figur 8.1: Kryptera pseudokod

```
En def Scrypt (N, frö):
2    V = [0] * N   // initialize minnesbuffert av längden N

   // Fyll upp minnesbufferten med pseudoslumpdata
3 V [0] = frö
4  för i = 1 till N:
5    V [i] = SHA-256 (V [i-1])

   // Tillgång minnesbuffert i en pseudoslumpmässig ordning
6 X = SHA-256 (V [N-1])
7  för i = 1 till N:
8    j = X% N   // Välj en slumpmässig index baserat på X
9    X = SHA-256 (X ^ V [j])  // Update X baserat på detta index

10  retur X
219
```

Figur 8.1 visar Scrypt-pseudokod. Det visar kärnprinciperna men vi har utelämnat några detaljer: i verkligheten fungerar scrypt på lite större datablock och algoritmen för att fylla i bufferten är något mer komplex.

För att se varför scrypt är minnessvårt, låt oss föreställa oss att försöka beräkna samma värde utan att använda buffert V. Detta skulle säkert vara möjligt — men i rad 9 skulle vi behöva räkna om värde V[j] i farten, vilket skulle kräva beräkningsjiterationer av SHA-256. Eftersom värdet av j under varje iteration av slingan kommer att väljas pseudoslumpmässigt mellan 0 och N-1, detta kommer att kräva om N/2 SHA-256-beräkningar. Detta innebär att beräkning av hela funktionen nu tar N * N/2 = N^2 /2 SHA-256-beräkningar, istället för bara 2N om en buffert används! Således konverterar användningen av minne Scrypt från en O (N) för att en O (N^2). Det ska vara enkelt att välja N stor nog att O (N^2) är långsam nog att använda minne är snabbare.

Tidsminnes kompromisser . Även om det skulle vara mycket långsammare att beräkna scrypt utan hjälp av en stor minnesbuffert är det fortfarande möjligt att använda mindre minne till priset av lite mer beräkning. Antag att vi använder en buffert av storlek N/2 (istället för storlek N). Nu kunde vi bara lagra värdena V[j] if j är jämnt, vilket förkastar värdena för vilka j är udda. I den andra slingan, ungefär hälften av tiden en udda värdet på j kommer att väljas, men detta är nu ganska lätt att beräkna i farten - vi beräknar helt enkelt SHA-256(V[j-1]) eftersom V[j-1] kommer att finnas i vår buffert. Eftersom detta händer ungefär halva tiden, lägger det till N/2 extra SHA-256-beräkningar.

Att halvera vårt minneskrav ökar således antalet SHA-256-beräkningar med endast en kvartal (från 2N till 5N/2). I allmänhet skulle vi bara kunna lagra varje k:te rad i bufferten V, med hjälp av N/k minne och beräkning (k+3)N/2 iterationer av SHA-256. I gränsen, om vi sätter k = N, är vi tillbaka till vår tidigare beräkning där körtiden blir O (N 2). Dessa siffror gäller inte exakt för scrypt själv, men de asymptotiska uppskattningarna gör det.

Det finns alternativa mönster som minskar förmågan att byta minne med tiden. Till exempel om bufferten uppdateras kontinuerligt i den andra slingan, det gör tidsminnesavvägningen mindre effektiv eftersom uppdateringarna måste lagras.

Verifiering kostnad . En annan begränsning av scrypt är att det tar lika mycket minne att verifiera som det gör beräkna. För att göra minneshårdheten meningsfull måste N vara ganska stor. Detta betyder att en enda beräkning av scrypt är storleksordningar dyrare än en singel iteration av SHA-256, vilket är allt som behövs för att kontrollera Bitcoins enklare gruvpussel.

Detta har några negativa konsekvenser, eftersom varje klient i nätverket måste upprepa denna beräkning i för att kontrollera att ett nytt block som gjorts anspråk på är giltigt. Detta kan bromsa spridningen och acceptansen av nya block och öka risken för gafflar. Det betyder också varje kund (även lätta SPV-klienter) måste ha tillräckligt med minne för att beräkna funktionen effektivt. Som ett resultat, mängden minne N som kan användas för kryptering i en kryptovaluta är något begränsad av praktiska problem.

Fram till nyligen var det inte känt om det var möjligt att designa ett gruvpussel som var minnesvärt beräkna men snabb (och minneslätt) att verifiera. Den här egenskapen är inte användbar för lösenordshashing, vilket hade varit det primära användningsfallet för minneshårda funktioner innan de användes i kryptovalutor.

2014 föreslog John Tromp ett nytt pussel kallat Cuckoo Cycle. Cuckoo Cycle är baserad på svårighet att hitta cykler i en graf som genereras från en cuckoo hash-tabell, en datastruktur som själv föreslogs först 2001. Det finns inget känt sätt att beräkna det utan att bygga upp en stor hash tabell, men det kan kontrolleras helt enkelt genom att kontrollera att en (relativt liten) cykel har hittats.

Detta kan göra minneshårda eller minnesbundna bevis på arbete mycket mer praktiska att använda i Bitcoin konsensus. Tyvärr finns det inga matematiska bevis för att denna funktion inte kan vara det beräknas effektivt utan att använda minne. Ofta verkar nya kryptografiska algoritmer säkra, men samhället är inte övertygat förrän de har funnits i många år utan att en attack förekommit hittades. Av denna anledning, och på grund av dess senaste upptäckt, har Cuckoo Cycle inte använts av någon kryptovaluta från och med 2015.

Scrypt i praktiken . Scrypt har använts i många kryptovalutor, inklusive flera populära som Litecoin. Resultaten har varit något blandade. Scrypt ASICs är redan tillgängliga för parametrar valda av Litecoin (och kopierade av många andra altcoins). Överraskande nog prestandan förbättringen av dessa ASIC jämfört med datorer för allmänna ändamål har varit lika med eller större än det för SHA-256! Således var scrypt definitivt inte ASIC-resistent i slutändan, åtminstone som det användes av

Litecoin. Utvecklarna av Litecoin hävdade initialt att ASIC-motstånd var en viktig fördel gentemot Bitcoin, men har sedan dess erkänt att detta inte längre är fallet.

Detta kan vara ett resultat av det relativt låga värdet på N (minnesanvändningsparametern) som används av Litecoin,
kräver bara 128 kB för att beräkna (eller mindre om en tidsminnesavvägning används, vilket var vanligt görs på GPU:er för att få hela bufferten att passa in i en snabbare cache). Detta har gjort det relativt enkelt att designa lättvikts-ASIC:er för gruvdrift utan en komplicerad minnesåtkomstbuss som behövs för åtkomst gigabyte RAM, som datorer för allmänna ändamål har. Litecoin-utvecklare valde inte ett värde som var mycket högre (vilket skulle göra ASIC svårare att designa) eftersom de ansåg att verifieringskostnad opraktisk.

Andra metoder för ASIC-motstånd. Minns att vårt ursprungliga mål var helt enkelt att göra det svårt att bygga ASIC:er med dramatiska prestandahöjningar. Minneshårdhet är bara en metod för detta mål, och det finns andra.

De andra tillvägagångssätten är tyvärr inte särskilt vetenskapliga och har inte utformats lika noggrant eller attackeras som minneshårda funktioner. Den mest kända heter X11, som helt enkelt är en kombination av elva olika hashfunktioner introducerade av ett altcoin kallat Darkcoin (senare omdöpt till DASH) och sedan dess använts av flera andra. Målet med X11 är att göra det betydligt mer komplicerat att designa en effektiv ASIC eftersom alla 11 funktioner måste implementeras i hårdvara. Men det här är inget annat än en olägenhet för hårdvarudesigners. Om en ASIC byggdes för X11 skulle den göra det gör säkerligen CPU och GPU mining föråldrad.
221

Sidebar: var kom X11s hashfunktioner ifrån? Från 2007 till 2012, US National Institute of Standards genomförde en tävling för att välja en ny hashfunktionsfamilj till SHA-3-standarden. Detta producerat ett stort antal hashfunktioner som lämnades in som kandidater, komplett med designdokument och källkod. Medan många av dessa kandidater visade sig inte vara det kryptografiskt säkra under tävlingen, 24 överlevde utan någon känd kryptografisk attacker. X11 valde elva av dessa, inklusive Keccak, den ultimata tävlingsvinnaren.

Ett annat tillvägagångssätt som har föreslagits, men som faktiskt inte har implementerats, är att ha ett gruvpussel det är ett rörligt mål. Det vill säga, själva gruvpusslet skulle förändras, precis som svårigheten periodvis förändringar i Bitcoin. Helst skulle pusslet förändras på ett sådant sätt att optimerad gruvhårdvara för det tidigare pusslet skulle inte längre vara användbart för det nya pusslet. Det är oklart exakt hur vi skulle göra byter faktiskt pusslet en gång då och då för att få de säkerhetskrav vi behöver.
Om beslutet skulle fattas av utvecklarna av en altcoin kan det vara en oacceptabel källa till centralisering. Till exempel kan utvecklarna välja ett nytt pussel som de redan har utvecklad hårdvara (eller bara en optimerad FPGA-implementering), vilket ger dem en tidig fördel.

Kanske kan sekvensen av pussel genereras automatiskt, men detta verkar också svårt.
En idé kan vara att ta en stor uppsättning hashfunktioner (säg de 24 SHA-3-kandidaterna som inte var trasig) och använd var och en i sex månader till ett år, för kort tid för att hårdvara ska kunna utvecklas. Av naturligtvis, om schemat var känt i förväg, så kunde hårdvaran helt enkelt designas just in tid att skicka för den tid varje funktion användes.

ASIC smekmånad. Avsaknaden av ASICs för X11 hittills, trots att de är klart möjligt att bygga, visar ett potentiellt användbart mönster. Eftersom inga altcoins som använder X11 har en särskilt hög marknadsandelar, det har helt enkelt inte funnits en tillräckligt stor marknad för att någon ska kunna bygga ASIC:er för X11 ännu.
Generellt sett har design av ASIC mycket höga initiala kostnader (i både tid och pengar) och relativt låga marginalkostnader per producerad hårdvara. För nya och oprövade kryptovalutor är det alltså inte det värt att göra en investering för att bygga hårdvara om valutan kan misslyckas innan den nya hårdvaran är tillgänglig för gruvdrift. Även när det finns en tydlig marknad finns det en tidsfördröjning innan hårdvaruenheter kommer vara redo. Det tog över ett år för de första Bitcoin ASIC:erna att skickas från när de var först designad, och detta ansågs vara blixtsnabbt för hårdvaruindustrin.

Således är det troligt att uppleva en eventuell ny altcoin med en ny gruv pussel *ASIC smekmånad* under vilken tid GPU och FGPA mining (och potentiellt CPU mining) kommer att vara lönsam. Det kanske inte är det möjligt att hejda strömmen av ASICs för alltid, men det finns kanske något värde i att göra det tilltalande individer att delta i gruvdrift (och tjäna några enheter av den nya valutan) medan den är bootstrapping.

Argument mot ASIC-motstånd. Vi har sett att det kan vara omöjligt att uppnå ASIC-motstånd i det långa loppet. Det finns också argument för att det är riskabelt att gå ifrån det relativt beprövade SHA-256 gruvpussel mot ett nytt pussel som kan vara svagare kryptografiskt. Vidare, SHA-256 mining ASICs designas redan nära moderna gränser för hårdvaruеffektivitet,
[222]

---

vilket innebär att den exponentiella tillväxtperioden troligen är över och SHA-256 gruvdrift kommer därför att erbjuda
mest stabilitet till nätverket.

Slutligen finns det ett argument att även på kort sikt är ASIC-resistens en dålig egenskap att ha. Återkallelse från kapitel 3 att även om det finns en gruvarbetare på 51 %, är många typer av attacker inte rationella för dem att försök eftersom det kan krascha växelkursen och decimera värdet av gruvarbetarens investering i hårdvara eftersom bitcoins de tjänar från gruvdrift kommer att vara värda mycket mindre.

Med ett mycket ASIC-resistent pussel kan detta säkerhetsargument falla isär. Till exempel en angripare kanske kan hyra en enorm mängd generisk datorkraft tillfälligt (från en tjänst som t.ex Amazons EC2), använder den för att attackera och drabbas sedan av inga ekonomiska konsekvenser eftersom de inte längre behöver
hyra kapaciteten efter attacken. Däremot, med ett "ASIC-vänligt" pussel, skulle en sådan angripare göra det inneboende behov av att kontrollera ett stort antal ASICs som endast är användbara för gruvdrift kryptovaluta. En sådan angripare skulle investeras maximalt i valutans framtida framgång.
Efter detta argument till sin logiska slutsats, för att maximera säkerheten, kanske gruvpussel borde möjliggör inte bara att effektiva gruv-ASIC:er kan byggas, utan utformas så att dessa ASIC:er är det helt värdelös utanför kryptovalutan!

8.3 Bevis-på-användbart-arbete

I kapitel 5 diskuterade vi hur energin som förbrukas (vissa skulle säga slösas bort) av Bitcoin-brytning, kallas *negativa externa* av ekonomer, är en potentiell oro. Vi uppskattade att Bitcoin gruvdrift förbrukar flera hundra megawatt ström. Den uppenbara frågan är om det finns

något pussel för vilket arbetet som görs för att lösa det ger någon annan nytta för samhället. Detta skulle uppgå till en form av återvinning och kan bidra till att öka det politiska stödet för kryptovalutor. Av naturligtvis skulle detta pussel fortfarande behöva uppfylla flera grundläggande krav för att göra det lämpligt för användning i
ett konsensusprotokoll.

Föregående distribuerad databehandling projekt. Idén att använda tomgång datorer (eller ”reserv cykler”) för bra är mycket äldre än Bitcoin. Tabell 8.3 listar några av de mest populära volontärdatorerna projekt. Alla dessa projekt har en egenskap som kan göra dem lämpliga för användning som beräkningsteknik pussel: specifikt involverar de något slags "nål i en höstack"-problem där det finns ett stort utrymme för potentiella lösningar och små delar av sökutrymmet kan kontrolleras relativt snabbt och parallellt. Till exempel i SETI@home får volontärer små portioner av observerad radio signaler för att söka efter potentiella mönster, medan i distributed.net ges volontärer ett litet utbud av potentiella hemliga nycklar att testa.

Frivilliga datorprojekt har lyckats genom att tilldela små delar av lösningsutrymmet individer för kontroll. Faktum är att detta paradigm är så vanligt att ett specifikt bibliotek kallas BOINC (Berkeley Open Infrastructure for Network Computing) utvecklades för att göra det enkelt att paketera små bitar av arbete för enskilda att avsluta.

I dessa ansökningar motiverades volontärerna främst av intresse för det underliggande problemet dessa projekt använder också ofta topplistor för volontärer för att visa upp hur mycket beräkning de har har bidragit. Detta har lett till några försök att spela topplistorna genom att rapportera arbete som var faktiskt inte färdig, vilket krävde att vissa projekt tog till att skicka en liten mängd överflödiga arbeta för att upptäcka fusk. För användning i en kryptovaluta är motivationen förstås främst monetär och vi kan förvänta oss att deltagarna försöker fuska så mycket som tekniskt möjligt.

Projekt
Grundat mål
Påverkan
Bra internet
Mersenne Prime Search
1996
Hitta stora
Mersenne primtal
Hittade den nya "största prime
nummer" tolv raka gånger,
inklusive $2^{57.885.161} - 1$
distributed.net
1997
Kryptografisk
brute-force demos
Första framgångsrika offentliga brute-force
av en 64-bitars kryptografisk nyckel

SETI@home
1999
Identifiera tecken på
utomjordiskt liv
Största projektet hittills med över 5
miljoner deltagare
Folding@home
2000
Atomnivå
simuleringar av
proteinveckning
Största beräkningskapacitet på
något frivilligt datorprojekt.
Mer än 118 vetenskapliga artiklar.

Tabell 8.3: Populära ”Volunteer computing”-projekt

Utmaningar att anpassa användbar-proof-of-arbete. Med tanke på framgången för dessa projekt kan vi försöka att helt enkelt använda dessa problem direkt. Till exempel när det gäller SETI@Home, där volontärer finns givet segment av radioobservationer som de testar för statistiska anomalier, kan vi bestämma det statistiska anomalier som är sällsynta än någon tröskel anses vara "vinnande" lösningar på pussel och låt alla gruvarbetare som hittar en skapa ett block.

Det finns några problem med denna idé. Först, notera att potentiella lösningar inte alla är lika troliga vara en vinnande lösning. Deltagarna kanske inser att vissa segment är mer benägna att producera anomalier än andra. Med ett centraliserat projekt tilldelas deltagarna arbete så att alla segment kan analyseras så småningom (kanske med mer lovande segment prioriterade). För gruvdrift, men alla gruvarbetare kan prova vilket segment som helst, vilket innebär att gruvarbetare kan flockas för att prova de mest troliga segmenten först.
Detta kan innebära att pusslet inte är helt framstegsfritt, om snabbare gruvarbetare vet att de kan testa det mesta lovande segment först. Jämför detta med Bitcoins pussel, där varje nonce är lika sannolikt för någon andra för att skapa ett giltigt block, så alla gruvarbetare uppmuntras att välja slumpmässiga nonces att prova. De problemet här visar en nyckelegenskap i Bitcoins pussel som vi tidigare tog för givet, den hos en *equiprobable lösning utrymme.*

Tänk sedan på problemet med att SETI@home har en fast mängd data att analysera baserat på observationer tagna med radioteleskop. Det är möjligt att det skulle bli det när gruvkraften ökade inga fler rådata att analysera. Jämför detta igen med Bitcoin, där ett effektivt oändligt antal SHA-256-pussel kan skapas. Detta avslöjar en annan viktigt krav: en *outtömlig pussel utrymme* behövs.

Tänk slutligen på att SETI@home använder en pålitlig, centraliserad uppsättning administratörer för att kurera det nya
radiodata och bestämma vad deltagarna ska leta efter. Återigen, eftersom vi använder vår pussel för att bygga en konsensusalgoritm kan vi inte anta att en centraliserad part ska hantera pusslet. Således,

Vi behöver ett pussel som kan *algoritm genereras* .

Vilken volontär computing projekt skulle kunna vara lämpliga som pussel ?. återgå till figur 8.3 kan vi se att SETI@home och Folding@home helt klart inte kommer att fungera för ett decentraliserat konsensusprotokoll.
Båda saknar förmodligen alla tre fastigheter vi nu har lagt till i vår lista. Den kryptografiska råkraften problem som tas upp av distributed.net kan fungera, även om de vanligtvis väljs som svar på specifika dekrypteringsutmaningar som har satts av företag som vill utvärdera säkerheten för vissa algoritmer. Dessa kan inte genereras algoritmiskt. Vi kan generera algoritmiskt dekryptering utmaningar att brytas av brute force, men på sätt och vis är detta exakt vad SHA-256 partiellt att hitta före bild gör det redan och det fyller ingen fördelaktig funktion.

Detta lämnar Great Internet Mersenne Prime Search, som visar sig vara nära att fungera. De utmaningar kan genereras algoritmiskt (hitta ett primtal som är större än det föregående) och pusslet utrymmet är outtömligt. Faktum är att det är oändligt, eftersom det har bevisats att det finns ett oändligt antal av primtal (och ett oändligt antal Mersenne-primtal i synnerhet).

Den enda verkliga nackdelen är att stora Mersenne Primes tar lång tid att hitta och är mycket sällsynta. I Faktum är att Great Internet Mersenne Prime Search har hittat endast 14 Mersenne-primtal på över 18 år! Det skulle helt klart inte fungera att lägga till mindre än ett block per år till en blockkedja. Detta specifika problem
verkar sakna den justerbara svårighetsegenskapen som vi angav var väsentlig i avsnitt 8.1. Det vänder men att ett liknande problem med att hitta primtal verkar fungera som ett beräkningspussel.

Primecoin. När detta skrivs, den enda proof-of-användbar-arbetssystem utnyttjas i praktiken kallas Primecoin. Utmaningen i Primecoin är att hitta en *Cunningham kedja* av primtal. A Cunningham kedjan är en sekvens av k primtalen p_1, S_2, ... p_k sådan att $p_i = 2p_{i-1} + 1$ för varje nummer i kedjan. Det vill säga att du tar ett primtal, dubblar det och lägger till ett för att få ytterligare ett primtal nummer och fortsätt tills du får ett sammansatt nummer. Sekvensen 2, 5, 11, 23, 47 är en Cunningham-kedja med längd 5. Det potentiella sjätte talet i kedjan, 95, är inte primtal (95 = 5·19).
Den längsta kända Cunningham-kedjan är av längd 19 (som börjar på 7991019772166787018701610 1). Det är gissade och allmänt trott, men inte visat, att det finns Cunningham kedjor av längd k för varje k .

Nu, för att göra detta till en beräknings pussel behöver vi tre parametrar m , n och k som vi kommer att förklara ett ögonblick. För en given utmaning x (omkastningen av det föregående blocket), vi tar de första m bitarna
av x och överväga någon kedja med längden k eller större, i vilken den första primär i kedjan är en n -bitars prime och har samma m ledande bitar som x för att vara en giltig lösning. Observera att vi kan justera n och k för att göra

pusslet svårare. Ökande k (den erforderliga kedjelängden) gör problemet exponentiellt hårdare, samtidigt öka n (storleken av utgångs prime) gör det linjärt hårdare. Detta ger finjustering av svårigheten. Värdet på m behöver bara vara stora nog att försöka pre-beräkna lösningar innan du ser värdet av föregående block är omöjligt.

Alla andra egenskaper vi har diskuterat verkar tillhandahållas: lösningar är relativt snabba att verifiera, problemet är framstegsfritt, problemutrymmet är oändligt (förutsatt att några väl studerade matematiska gissningar om fördelningen av primtal är sanna), och pussel kan vara det algoritmiskt genererade. Detta pussel har faktiskt använts för Primecoin i nästan två år och detta har producerat den största kända primtal i Cunningham kedjor för många värden på k . Primecoin har sedan dess utökats till att inkludera ytterligare, liknande typer av prime-kedjor i sitt proof of work, inklusive ”Andra slaget” Cunningham kedjor i vilka $p_i = 2p_{i-1} - 1$.

Detta ger starka bevis för att det är möjligt att göra bevis-på-användbart-arbete praktiskt i vissa begränsade omständigheter. Naturligtvis är det diskutabelt i vilken utsträckning att hitta stora Cunningham-kedjor är användbar. Det är möjligt att de kan ha något tillämpat syfte i framtiden och de står säkert kvar som ett litet bidrag till vår samlade matematiska kunskap. För närvarande har de dock ingen kända praktiska tillämpningar.

Permacoin och proof-of-lagring. Ett annat sätt att proof-of-nyttigt arbete är *proof-of-storage* (även ibland kallad *proof-of-återtagbarhet*). Istället för att kräva ett enbart beräkningspussel, tänk om vi kunde designa ett pussel som krävde lagring av en stor mängd data för att beräkna? Om denna data var användbara, skulle gruvarbetarnas investering i gruvhårdvara effektivt bidra till en brett distribuerat och replikerat arkivlagringssystem.

Vi tar en titt på *Permacoin* , det första förslaget till proof-of-lagring för användning i konsensus. Vi börjar med en stor fil som vi kallar F . För nu, låt oss anta att alla är överens om värdet av F och filen kommer inte att ändras. Till exempel F kan väljas av en betrodd återförsäljare när en kryptovaluta är lanseras, ungefär som varje ny valuta måste komma överens om ett genesisblock för att komma igång. Detta skulle helst vara en fil av allmänt värde. Till exempel experimentella data som samlats in från Large Hadron Collider består redan av flera hundra petabyte (PB). Att tillhandahålla en gratis säkerhetskopia av dessa data skulle vara ganska användbar.

Naturligtvis, eftersom F är en stor fil flesta deltagarna inte kommer att kunna lagra hela filen. Men vi redan vet hur man använder kryptografiska hashfunktioner för att säkerställa att alla är överens om F utan att veta det hela. Det enklaste tillvägagångssättet skulle vara för alla att komma överens om $H(F)$, men en bättre metod är att representera F med hjälp av en stor Merkle träd och har alla deltagare överens om rotens värde. Nu, alla kan komma överens om värdet av F och det är effektivt att bevisa att någon del av F är korrekt.

I Permacoin, varje gruvarbetare M lagrar en slumpmässig delmängd $F_M \subseteq F$. För att uppnå detta, när gruvarbetaren genererar en publik nyckel K_M

som de kommer att använda för att ta emot pengar, hash de sin publika nyckel för att generera en pseudoslumpmässig uppsättning block F_M som de måste lagra kunna gruvan. Denna delmängd kommer att vara av några fast antal block k_1. Vi måste här anta att det finns något sätt för dem att hämta dessa block när de börjar bryta - kanske ladda ner dem från en kanonisk källa.

När gruvarbetaren har lagrat F_M lokalt är pusslet ganska likt konventionell SHA-256 gruvdrift. Given ett tidigare block hash x, gruvarbetaren väljer ett slumpmässigt nonce värde n och hashar detta för att generera en pseudoslump delmängd $F_{M,n} \subseteq F_M$ bestående av $k_2 < k_1$ block. Observera att denna delmängd beror på båda nonce de har valt och deras publika nyckel. Slutligen beräknar gruvarbetare en SHA-256 hash av n och blocken i F_k. Om värdet på denna hash är under en målsvårighet har de hittat en giltig lösning.

Figur 8.4: Att välja slumpmässiga block i en fil i Permacoin.
I detta exempel k_1 = 6 och k_2 =2. I en verklig implementering skulle dessa parametrar vara mycket större.

För att verifiera en lösning krävs följande steg:

- Kontrollera att $F_{M,n}$ var korrekt genereras från gruvarbetare offentliga nyckel K_M och nonce n
- Kontrollera att varje block av av $F_{M,n}$ är korrekt genom att verifiera deras väg i Merkle-trädet till globalt överenskomna roten av F.
- Kontrollera att H ($F_{M,n}$ || n) är mindre än målet svårigheter.

Det ska vara lätt att se varför lösa pusslet kräver gruvarbetare att lagra alla $F_{M,n}$ lokalt. För varje nonce, de gruvarbetare behov att testa hash av en slumpmässig undergrupp av block av $F_{M,n}$, vilket skulle vara oöverkomligt långsam att hämta över nätverket från fjärrlagring.

Till skillnad från fallet med Scrypt finns det inga rimliga tid / minnes avvägningar förutsatt att k_2 är stor tillräckligt. Om en gruvarbetare lagras endast halv av F_M lokalt, och k_2 =20, de måste prova en miljon nonces innan de hittade en som inte krävde att några block skulle hämtas över nätverket. Så minska deras lagringsbördan med en konstant faktor ökar deras beräkningsbörda exponentiellt. Självklart,

inställning k_2 vara för stor kommer inte att vara mycket effektiv, eftersom k_2 Merkle trädstigar måste överföras och verifierad i någon giltig lösning.

Det finns också en avvägning i inställning k_1. Den mindre k_1 är, desto mindre lagring behövs för att fungera som en gruvarbetare och därför är gruvdriften mer demokratisk. Men detta betyder också att större gruvarbetare har nej incitament att lagra mer än k_1 block av F, även om de har förmågan att lagra mer.

Som vanligt är detta en liten förenkling av hela Permacoin-förslaget, men detta är tillräckligt för att förstå de viktigaste designkomponenterna. Den största praktiska utmaningen är förstås att hitta en passande stor fil det är viktigt, offentligt och i behov av ytterligare replikering. Det finns också betydande komplexitet om filen F förändras över tiden, liksom med att justera gruv svårigheten över tiden.

Långsiktiga utmaningar och ekonomi. För att sammanfatta det här avsnittet, proof-of-användbar-arbete är en mycket naturligt mål, men det är ganska utmanande att uppnå det med tanke på de andra kraven på en vara beräkningspussel för ett konsensusprotokoll. Även om åtminstone två exempel är kända som är tekniskt genomförbara, Primecoin och Permacoin, har båda vissa tekniska nackdelar (främst längre verifieringstid för påstådda lösningar). Båda ger dessutom ganska ringa allmännytta jämfört med omfattningen av ansträngning som vi har sett tas ut vid Bitcoin-brytning med miljontals dollar kapital och megawatt förbrukad el.

Det finns ett intressant ekonomiskt argument att fördelen med alla bevis-på-användbart-arbete bör vara en ren *kollektiv nyttighet* . Inom ekonomi är en allmän nytta en som inte kan uteslutas, vilket betyder att ingen kan vara det förhindrad från att använda den, och icke-rivalerande, vilket innebär att varans användning av andra inte påverkar dess värde. Det klassiska exemplet är en fyr.

Några av de exempel vi diskuterade här, som proteinveckning, kanske inte är en ren allmännytta eftersom vissa företag (som stora läkemedelsföretag) kan dra mer nytta av ökad kunskap om proteinveckning än andra. I grund och botten skulle gruvdrift vara billigare för dessa parter eftersom de får mer nytta av de offentliga förmånerna än vad andra skulle göra.

8.4 Pussel som inte går att lägga ut på entreprenad

Låt oss vända oss till ett annat potentiellt designmål för alternativa gruvpussel: att förhindra bildandet av gruvpooler. Som vi diskuterade i kapitel 5 och på andra ställen, bryter de flesta Bitcoin-gruvarbetare som en del av en pool snarare än självständigt. Detta har resulterat i ett fåtal stora pooler som tillsammans representerar det mesta gruvkraften. Eftersom varje pool drivs av en central pooladministratör, anser vissa att detta är en farlig trend bort från Bitcoins kärndesignprincip om decentralisering och kan äventyra dess

säkerhet.

Medan en gruvpool med majoritetsandel är ett uppenbart problem, alla stora centralt förvaltade pooler kan implementera en icke-standard gruvstrategi och attackera nätverket. Sådana pooler är också en saftig mål för hackare att försöka kompromissa för att omedelbart kontrollera en stor mängd gruvkraft. De pooloperatörer kan samarbeta för att censurera transaktioner eller genomdriva höga transaktionsavgifter. Åtminstonde,
att ha flest gruvarbetare i pooler betyder också att de flesta gruvarbetare inte kör en fullständigt validerande nod.

228

Intressant nog har dessa bekymmer en analogi när det gäller röstning. Det är olagligt i USA och många andra nationer för individer att sälja sin röst. Antagligen deltar i en pool som kontrolleras av någon annan liknar att sälja din röst i Bitcoin-konsensusprotokollet.

Tekniska krav för pooler. Kom ihåg att gruvpooler verkar vara ett framväxande fenomen. Det finns inga bevis för att Satoshi tänkte på gruvpooler vid tidpunkten för Bitcoins ursprungliga design. Det var inte uppenbart på några år att effektiva pooler kunde drivas mellan många individer som inte känner eller litar på varandra.

Som vi såg i kapitel 5 fungerar gruvpooler vanligtvis genom att utse en pooloperatör med en välkänd offentlig nyckel. Var och en av de deltagande gruvarbetarna bryter som vanligt men skickar in andelar till pooloperatören.
Dessa andelar är "nästan misstag" eller "dellösningar", vilket skulle vara giltiga lösningar vid en lägre svårighetsgrad. Detta visar pooloperatören hur mycket arbete gruvarbetaren utför. Närhelst en av pooldeltagarna hittar ett giltigt block, pooloperatören delar sedan ut belöningarna mellan pooldeltagarna baserat på antalet aktier de lämnat in. Som vi diskuterade i kap 5, det finns många formler för att dela upp intäkterna, men alla gruvpooler följer detta grundläggande strukturera.

Förekomsten av pooler är alltså beroende av minst två tekniska egenskaper hos Bitcoin. Det första är att det är det lätt för en gruvarbetare att bevisa (sannolikt) hur mycket arbete de utför genom att lämna in aktier. Förbi genom att välja en tillräckligt låg tröskel för aktier kan gruvarbetare enkelt bevisa hur mycket arbete de är presterar med godtycklig precision oavsett den faktiska svårigheten att hitta ett giltigt block. Detta aspekten av gruvpussel verkar vara svår att ändra, med tanke på att vi behöver ett pussel som kan skapas med godtycklig svårighet.

För det andra kan poolmedlemmar enkelt bevisa för pooloperatören att de följer reglerna och arbetar med att hitta giltiga block som skulle belöna poolen som helhet. Detta fungerar eftersom poolen är offentlig nyckel är förpliktad till i myntbastransaktionen som ingår i blockets Merkle-träd av transaktioner. När en gruvarbetare väl hittar ett block eller till och med en del, kan de inte ändra vilken offentlig nyckel som är
mottagare av de nypräglade mynten.

Blockera kasseringsattacker. Det finns en svaghet i detta system för att implementera gruvpooler: där är inget att tvinga fram att deltagande gruvarbetare faktiskt skickar in giltiga block till poolchefen i händelsen att de hittar dem. Anta att det finns en poolmedlem som är upprörd över en stor gruvdrift slå samman. De kan delta i poolen genom att bryta och skicka in aktier precis som vanligt, men i

Om de faktiskt hittar ett giltigt block som skulle belöna poolen kasserar de det helt enkelt och gör det inte berätta för pooloperatören om det.

Denna attack minskar poolens totala gruvkraft eftersom inget av angriparens arbete bidrar för att hitta giltiga block. Men angriparen kommer fortfarande att belönas som de ser ut att vara skicka in giltiga aktier och helt enkelt ha otur att inte hitta några giltiga block. Om gruvpoolen är utformad för att vara intäktsneutral (det vill säga alla gruvbelöningar omfördelas tillbaka till deltagarna) denna attack kan göra att poolen går med förlust.
229

Denna attack kallas ibland en *vigilante* eller *sabotage* attack och anses vara en form av vandalism eftersom attacken verkar vara kostsam för både angriparen och poolen. Angriparen förlorar pengar eftersom varje block de kastar skulle ha lett till att någon del av blockbelöningarna hade blivit återvände till dem. Naturligtvis får angriparen fortfarande belöningar för andra pussellösningar som hittas.

Det verkar som att en rationell angripare inte skulle använda denna strategi, eftersom de skulle förlora pengar utan att få något påtagligt. Det visar sig (ganska överraskande) att det finns fall där detta strategi kan vara lönsam, som diskuteras i rutan nedan. Men vi vill i alla fall designa en helt ny formulering av gruvpussel som säkerställer att denna strategi alltid är lönsam.

Sidofält: blockera kasseringsattacker mellan pooler. Folk har antagit i flera år att det inte kan vara så lönsamt för en deltagare att kassera giltiga block som hittats för poolens räkning. Det visar sig detta strategi kan vara lönsam om en gruvpool använder den för att attackera en annan. Detta föreslogs apokryfiskt många gånger och först noggrant analyserat i en artikel av Ittay Eyal 2015.

Låt oss överväga ett enkelt fall: anta att två gruvpooler, A och B, vardera har 50 % av den totala gruvdriften kapacitet. Anta nu att B använder hälften av sin gruvkraft (25 % av den totala kapaciteten) för att bryta som en medlem i pool A, men kasserar alla hittade block. Vi kan visa, i en förenklad modell, att B kommer tjänar nu 5/9 av de totala belöningarna, större än de 50 % den skulle tjäna genom att bryta normalt. I denna enkelt fall, att dedikera hälften av sin gruvkraft till attacker kan visa sig vara det optimala strategi för pool B.

Situationen blir mer komplicerad med flera pooler. Blockkastning har inte observerats i praktiken i stor skala när detta skrivs. Men det är fortfarande möjligt att i det långa loppet, attacker som denna kommer att ifrågasätta livskraften för stora gruvpooler.

Belöningssabotage. Vårt designmål är att göra det så att gruvarbetare uppmuntras att bryta i en pool men inte skicka in giltiga block till poolansvarig. För närvarande kan endast poolchefen samla in mining belöningar eftersom chefen kräver att alla deltagare inkluderar en specifik publik nyckel i myntbastransaktion av block som de bryter. Korrekt inkludering kan enkelt checkas in dellösningar. Poolchefen är den enda part som känner till den privata nyckeln och kan därför avgöra var de nypräglade mynten går.

Men tänk om vi krävde att alla deltagare också kände till den privata nyckeln (och därmed kunde omdirigera medel efter att ha brutit ett block?). För att göra detta behöver vi ett pussel där varje lösningsförsök kräver kunskap om den privata nyckeln i myntbastransaktionen. Vi kan ändra pusslet från "hitta ett block vars hash är under ett visst mål”att”hitta ett block där hash av *en signatur* på blocket är

under ett visst mål." Denna signatur måste beräknas med samma publika nyckel i myntbasen transaktion.

Ett sådant pussel lämnar pooloperatörer med två ohållbara val. De kan distribuera privat nyckel till alla pooldeltagare, i vilket fall vilken som helst av dem kan stjäla alla medel. Växelvis,

---

de kan utföra signaturerna på uppdrag av pooldeltagare. Att beräkna en signatur är beställningar av storleken dyrare än att beräkna en hash, men i det här fallet skulle poolchefen göra det göra det mesta av det tunga arbetet. Det vore bättre för poolchefen att helt enkelt vara solo gruvarbetare.

För- och nackdelarna med gruvdrift som inte kan läggas ut på entreprenad. Eftersom detta pussel inte effektivt kan outsourcas till en opålitlig deltagare gör det det mycket mer utmanande, om inte direkt omöjligt, att bilda en gruvpool med opålitliga deltagare. Det förhindrar effektivt *alla* pooler, även insatser som P2Pool till göra en decentraliserad pool utan poolansvarig.

Det finns ett argument som driftsätta ett sådant pussel kan perversely leda till *mer* centralisering, inte mindre, eftersom det skulle avskräcka små gruvarbetare från att delta på grund av den höga variansen de skulle göra ansikte. Detta skulle bara lämna kvar stora gruvverksamheter. För närvarande, medan pooler nominellt kan kontrollera a stor mängd gruvkraft, är det inte klart att de kan använda detta för att starta en attack utan att se många av deras medlemmar hoppar av. Det är fortfarande en öppen fråga vilken risk som är värst - den med stor gruvdrift pooler, eller att begränsa gruvdrift till operatörer som är tillräckligt stora för att leva med en hög varians.

Den heliga gralen skulle vara att utforma ett konsensusprotokoll som är "naturligt" lågvariant genom att belöna gruvarbetare en liten summa för pussel med lägre svårighetsgrad. Detta skulle innebära att gruvarbetare inte behöver bilda pooler och ändå kan små gruvarbetare fortfarande delta. Det går inte att bara minska den genomsnittliga tiden mellan blocken arbete — det skulle behöva minskas med en faktor på 1 000 eller mer för att den resulterande variansen ska vara motsvarande dagens stora gruvbassänger. Men då skulle fördröjningen mellan blocken vara mindre än a andra och antalet inaktuella block skulle bli kaotiskt högt. Det är fortfarande en öppen fråga om det finns är en alternativ version av konsensusprotokollet som skulle möjliggöra enklare gruvpussel utan kräver nästan omedelbar sändning av alla lösningar.

8.5 Proof-of-Stake och virtuell gruvdrift

För att avsluta detta kapitel, låt oss titta på idén att ersätta beräknings pussel med *virtuella brytning* . Denna term hänvisar till en olik uppsättning tillvägagångssätt men de har alla det gemensamt att de bara kräver en mindre utgifter för beräkningsresurser av deltagande gruvarbetare.

Sluter slingan på gruvdrift. Som ett tankeexperiment, anta Bitcoin eller en annan kryptovaluta blir den dominerande betalningsformen globalt. Gruvarbetare skulle börja med några initiala innehav av kryptovaluta, använda den för att köpa gruvutrustning och el, förbruka dessa resurser och in

processen, skaffa ny kryptovaluta i form av gruvbelöningar. Denna process brinner kontinuerligt
energi och råvaror.

231

Figur 8.5: Cykeln för Bitcoin-brytning

När gruvhårdvara blir en vara och elektricitet är en vara (som det i allmänhet redan
är), skulle ingen gruvarbetare ha en betydande fördel jämfört med någon annan gruvarbetare när det gäller hur effektivt de är
kunde konvertera sina ursprungliga kryptovalutainnehav till gruvbelöningar. Med undantag för mindre variationer i
effektivitet, den som investerar mest i gruvdrift kommer att få flest belöningar.

Den grundläggande frågan som motiverar virtuell gruvdrift är: vad skulle hända om vi tog bort steget av
spendera pengar på kraft och utrustning? Denna process används trots allt främst för att bevisa vem som har
investerat mest i gruvdrift. Varför inte helt enkelt allokera gruvkraft direkt till alla valutainnehavare
i proportion till hur mycket valuta de faktiskt har?

Kom ihåg att det ursprungliga målet med Bitcoin-brytning var att möjliggöra en form av omröstning om blockets tillstånd
kedja, med gruvarbetare med mer datorkraft som får fler röster. Vi kunde istället designa vår
”röstnings”-system så att rösterna bestäms av hur mycket valuta man för närvarande innehar.

Fördelar med virtuell gruvdrift. Den primära fördelen med detta tillvägagångssätt är uppenbar: det tar bort
slösaktig högra hälften av gruvcykeln från figur 8.5, vilket lämnar oss med ett "stängt" system som visas i
Figur 8.6.

Figur 8.6: Den virtuella gruvcykeln

Förutom enkelhet skulle detta tillvägagångssätt dramatiskt minska Bitcoins miljöavtryck. Det
skulle inte minska energiförbrukningen till noll, eftersom gruvarbetare alltid kommer att behöva spendera en del
beräkningsresurser för att kommunicera med nätverket och validera. Lite virtuell gruvdrift
232

---

system kräver också en liten mängd beräkningsbrytning. Men i båda fallen, det stora
majoriteten av gruvarbetet som utförs i Bitcoin kan potentiellt elimineras.

Virtuell gruvdrift kan också minska trenden mot centralisering. Eftersom det inte finns någon gruvhårdvara
involverat finns det ingen oro för en ASIC-fördel; alla gruvarbetare kan bryta lika "effektivt" som
alla andra. Alla virtuella gruvpussel uppnår alla målen för ASIC-resistenta pussel.

Kanske viktigast av allt, virtuell gruvdrift kan lösa problemet som vi diskuterade i sammanhanget

av ASIC-resistenta pussel, nämligen att gruvarbetare inte får investeras i den långsiktiga hälsan hos valuta. Alla som har några bitcoins är faktiskt en intressent i valutan och en mäktig virtuell gruvarbetare (som en som har 51 % eller mer av all valuta) är en mycket stor intressent. De ha ett incitament att göra saker som skulle gynna systemet som helhet eftersom det ökar värdet av mynten som de har. Detta argument är ännu starkare än argumentet att en gruvarbetare som sitter på en stort lager av gruvutrustning vars värde beror på valutans framtid kommer inte att fungera uppsåtligt.

Det är där begreppet *proof-of-spel* kommer ifrån. Till och med mer än att eliminera gruvdrift och sparande energi, är kanske den mest grundläggande motivationen för virtuell gruvdrift att se till att gruvdrift sker av intressenter i valutan som har de starkaste incitamenten att vara goda förvaltare av systemet.

Implementera virtuell gruvdrift: Peercoin. Det finns många varianter av virtuell gruvdrift som vi kommer att göra
beskriv några av de vanligaste idéerna. Vi bör betona att dessa idéer ännu inte har gjorts studerat på ett vetenskapligt och rigoröst sätt, och de har inte heller genomgått den nivå av praktiska tester som proof-of-work beror på Bitcoins popularitet.

Till att börja med kommer vi att överväga tillvägagångssättet från Peercoin, som lanserades 2012 som det första altcoin med proof-of-stake. Peercoin är en hybrid proof-of-work/proof-of-stake-algoritm där "insats" betecknas med "myntålder". Myntåldern för en specifik outnyttjad transaktionsutgång är produkten av det belopp som innehas av den produktionen och antalet block som produktionen har kvar outnyttjad. Nu, för att bryta ett block i Peercoin måste en gruvarbetare lösa ett SHA-256-baserat beräkningspussel precis som i Bitcoin. Men svårigheten för detta pussel justeras ner baserat på hur mycket myntålder de är villiga att konsumera. För att göra detta inkluderar blocket en speciell "coinstake" transaktion där vissa transaktioner spenderas helt enkelt för att nollställa deras myntålder. Summan av myntåldrarna konsumeras i coinstake-transaktionen avgör hur svårt proof-of-work-pusslet är att göra en givet block giltigt.

Det är möjligt för gruvarbetare att bryta med mycket liten insats och en stor mängd beräkningskraft, men svårighetsformeln är vald för att göra det dramatiskt lättare att hitta ett block om någon myntålder är det förbrukad. Effekten av beräkningspusslet är främst att säkerställa att processen är randomiserad om två gruvarbetare försöker konsumera en liknande mängd myntålder.

Många virtuella mining altcoins har antagit lite olika design, inklusive Nxt, BitShares, BlackCoin och Reddcoin. I var och en av dessa används en viss insats för att göra en beräkning

---

pusslet mycket lättare, påstås till den grad att beräkningspusslet inte längre är det huvudsakliga utmaning inom gruvdrift.

Alternativa former av insats. Ett par alternativ till denna hybridmodell som är värda att diskutera:

- **Proof-of-stake.** Den renaste formen av proof-of-stake är helt enkelt att göra gruvdrift lättare för dem som kan visa att de kontrollerar en stor mängd valuta. Detta liknar Peercoins bevis-på-myntålder, endast med ålder som inte beaktas. Nackdelen med detta tillvägagångssätt är det till skillnad från myntåldern som återställs efter framgångsrik gruvdrift, ges alltid de rikaste deltagarna det enklaste gruvpusslet.
- **Depositionsbevis** . I denna formulering, när mynt används av en gruvarbetare för att prägla ett block, de

bli fryst för ett visst antal block. Detta kan ses som en spegel av myntåldern:
istället för att belöna en gruvarbetare för att han innehar mynt som inte har använts under lång tid i
Tidigare belönar detta system gruvarbetare som är villiga att få mynt att förbli oberörda under en lång tid
in i framtiden. I båda metoderna kommer gruvarbetarnas andel i praktiken från alternativkostnaden
att inte kunna använda mynten för att utföra andra handlingar.

Problemet med ingenting som står på spel. Virtuell gruvdrift är ett aktivt område av pågående forskning och det finns
stora öppna problem. Medan några kryptovalutor har lanserats och överlevt med virtuell gruvdrift,
de har mött samma press som Bitcoin att stå emot motiverade angripare.

Den generiska sårbarhet virtuella gruvsystem är vad som ofta kallas *ingenting-at-spel*
problem eller *stavs-slipning attacker.* Antag att en angripare med en proportion $\alpha < 50\%$ av insatsen är
försöker skapa en gaffel av k block. Som vi har diskuterat tidigare kommer denna attack att misslyckas med hög
sannolikhet som är exponentiellt ökar i k . I traditionell gruvdrift har en misslyckad attack en betydande
alternativkostnad, eftersom den gruvarbetaren kunde ha tjänat gruvbelöningar under brytningen
process istället för att slösa gruvresurser på dess misslyckade attack.

Med virtuell gruvdrift existerar inte denna alternativkostnad. En gruvarbetare kan använda sin insats för att bryta i
nuvarande längsta kedjan samtidigt som man försöker skapa en gaffel. Om deras gaffel lyckas kommer den att göra det
har förbrukat en stor del av sin insats. Om det misslyckas kommer inte registreringen av att det misslyckas att reflekteras
den så småningom längsta kedjan.

Således kan rationella gruvarbetare ständigt försöka splittra kedjan. Olika försök har gjorts
gjorts för att ta itu med denna fråga. De flesta virtuella gruvsystem har varit mycket mer aggressiva
använder checkpointing för att förhindra långa gafflar, men som diskuterats tidigare är detta lite av en slutkörning
kring ett decentraliserat konsensusprotokoll.

För Ethereum (ett altcoin lanserat i mitten av 2015 som vi kommer att diskutera i kapitel 10), ett förslag som heter
Slasher tillåter straff för gruvarbetare som försöker dela kedjan. I Slasher använder du insats för att bryta
kräver att det aktuella blocket signeras med den privata nyckeln som motsvarar de transaktioner som ingår
gruvarbetarens insats. Om en gruvarbetare någonsin använder samma insats för att signera två inkonsekventa kedjor (ingen av
som är ett prefix till den andra), tillåter Slasher gruvarbetare att ange dessa två signaturer senare i
blockera kedjan som bevis på dåligt uppförande och samla in en del av denna insats som en belöning. Medan detta dyker upp

---

För att tillhandahålla en effektiv lösning är detaljerna i protokollet ganska komplicerade och det har det ännu inte varit
distribueras framgångsrikt.

En sista motåtgärd som kan finnas är att, som vi har sett för traditionella gruvsystem, gruvarbetare
kanske helt enkelt inte har ett starkt incitament att attackera eftersom detta skulle skada systemet och
underminera deras insats, även om attacken är framgångsrik.

Andra nackdelar med virtuell gruvdrift. Två andra nackdelar är värda att snabbt nämna. Den första är att vissa former av virtuell gruvdrift, även i frånvaro av stakslipning, kan göra vissa typer av attacker lättare eftersom det är möjligt att "spara ihop" till en explosion av gruvkraft. Till exempel en stor mängden myntinsats kan slås samman för att möjliggöra en dramatisk ökning av gruvdrift för att kanske införa en gaffel. Detta är möjligt även om ett system som Slasher används för att motverka gruvdrift på två kedjor samtidigt. För att motverka denna typ av attack begränsar Peercoin åldersparametern till 90 dagar vid datoranvändning mynt.

En andra fråga är att om en gruvarbetare i ett virtuellt gruvsystem får 51 % av den tillgängliga insatsen, kan de upprätthålla det för alltid genom att bara bryta ovanpå sina egna block, i huvudsak ta kontroll över blocket kedja. Även om nya insatser uppstår från gruvbelöningar och transaktionsavgifter kommer gruvarbetaren på 51 % att få
denna nya andel och deras andel av den totala andelen kommer sakta att närma sig 100 %. I traditionell gruvdrift, till och med
om det finns en gruvarbetare på 51 % är det alltid möjligt att någon ny gruvarbetare kommer att dyka upp med mer brytning
utrustning och energi och minska majoriteten gruvarbetare. Med virtuell gruvdrift är det mycket svårare att undvika detta problem.

Kan virtuell gruvdrift verkligen fungera? Virtuell gruvdrift är fortfarande något kontroversiell i mainstream Bitcoin-gemenskap. Det finns ett argument att säkerhet i grunden kräver att man bränner verkliga resurser, kräver riktig beräkningshårdvara och förbrukar verklig elektrisk kraft för att hitta pussel
lösningar. Om man tror på detta argument, så kan det uppenbara slöseriet med proof of work-systemet vara tolkas som kostnaden för den säkerhet som du får. Men detta argument har inte bevisats, precis som säkerheten för virtuell gruvdrift har inte bevisats.

Sammanfattningsvis finns det många saker man skulle vilja ändra på Bitcoins gruvpussel, och detta har varit ett område med rasande forskning och innovation. Än så länge dock inget av alternativen tycks både ha visat teoretisk soliditet och funnit praktisk adoption. Till exempel,
även om scrypt har varit ett populärt val i altcoins, har det faktiskt inte uppnått ASIC-motstånd, och dess användbarhet är oklar. Det är fullt möjligt att alternativa gruvpussel hittar fler framgång i framtiden. Trots allt kom Bitcoin själv efter decennier av misslyckade försök att skapa en kryptovaluta och lyckades träffa den söta punkten mellan principiell design och praktiska avvägningar.
235

Vidare läsning

Papperet som definierar minneshårda funktioner och föreslår kryptering:

Percival, Colin. Starkare nyckel härledning via sekventiella minneshårda funktioner . Självpublicerad, 2009.

Tidigare uppsatser om minnesbundna funktioner:

Abadi, Martin, Mike Burrows, Mark Manasse och Ted Wobber. Måttligt hårt, minne bundna funktioner . ACM Transactions on Internet Technology, 2005.

Dwork, Cynthia, Andrew Goldberg och Moni Naor. På minnesbundna funktioner för att bekämpa skräppost .
In Advances in Cryptology—Crypto, 2003.

Gökcykelns förslag:

. Tromp, John Cuckoo Cycle: ett minne hård proof-of-work-system . IACR Cryptology ePrint-arkiv 2014.

Permacoins förslag:

Miller, Andrew, Ari Juels, Elaine Shi, Bryan Parno och Justin Katz. Permacoin: Återanvända Bitcoin arbete för data bevarande . I IEEE Security and Privacy, 2014.

Det här dokumentet diskuterar olika hashfunktionsdesigner och SHA-3-tävlingen:

Preneel, Bart. De första 30 åren av kryptografiska hashfunktioner och NIST SHA-3 tävling . I Ämnen i kryptologi — CT-RSA, 2010.

Förslaget om pussel som inte kan läggas ut på entreprenad:

Miller, Andrew, Elaine Shi, Ahmed Kosba och Jonathan Katz. Nonoutsourceable Scratch-Off pussel att Motverka Bitcoin Mining koalitioner . Förtryck 2015.

Kapitel 9: Bitcoin som plattform

I tidigare kapitel utvecklade vi den tekniska grunden för Bitcoin och såg hur den kan användas som en valuta. Nu ska vi titta på program *annan valuta än* att vi kan bygga med hjälp av Bitcoin som central komponent. Vissa av dessa förlitar sig på Bitcoin som det är idag, utan några modifieringar, och många andra skulle endast kräva små modifieringar.

Vi har valt dessa applikationer för en kombination av praktisk användbarhet och intellektuellt intresse. Den här listan är inte på något sätt uttömmande, utan ser hur dessa applikationer fungerar (eller skulle kunna fungera, sedan
många är bara idéer eller förslag) kommer att ge dig insikt i de många sätt som Bitcoins funktionalitet kan återanvändas.

9.1. Bitcoin som en logg med endast tillägg

Det är bra att tänka på Bitcoin som *append endast log* - en datastruktur som vi kan skriva ett nytt data, och det som när vi väl har skrivit data, är manipuleringssäkert och tillgängligt för alltid. Vi har också en säker uppfattning om beställning: vi kan se om en databit skrevs till loggen före eller efter en annan bit. Denna ordning uppstår från blockhashpekarna, inte blockets tidsstämplar — a blockets tidsstämpel kan faktiskt vara ett lägre (tidigare) värde än dess föregångare. Det är för att gruvarbetare

kan ljuga om tidsstämplar, gruvarbetarnas klockor kanske inte är synkroniserade och det finns latens på nätverk. Som sagt, om en blockeringstidsstämpel verkar vara avstängd med mer än några timmar, då annat gruvarbetare kommer att avvisa det, så vi kan lita på att tidsstämplarna är ungefär korrekta. Som vi kommer att se, dessa
egenskaper visar sig vara ganska användbara.

Säker tidsstämpling. APPEND endast log kan användas för att bygga en säker tidsstämpling system från Bitcoin. Vi vill kunna bevisa att vi vet något värde x vid någon viss tidpunkt T. Vi kanske inte vill faktiskt *avslöja* x vid tiden T. Istället vi bara vill avslöja x när vi faktiskt gör bevis, som kan vara mycket senare än T (och naturligtvis om vi visste att det på T, fortfarande vet vi att det efter T också).
Men när vi väl har gjort beviset vill vi att bevisen ska vara permanenta.

Kom ihåg från kapitel 1 att vi kan använda hash-funktioner för att övergå till data. Istället för att publicera uppgifter x att vi vill bevisa att vi vet, kan vi publicera bara hash $H(x)$ till blocket kedjan. De egenskaper hos hash funktionsgaranti att vi inte senare kan hitta några olika värde y med samma värde, $y \neq x$ så att $H(x) = H(y)$. Vi förlitar oss också på den bekväma egenskapen att hash av x avslöjar inte någon information om x, så länge som x är vald från en fördelning med hög min-entropi, det vill säga den är tillräckligt oförutsägbar. Om x inte har denna egenskap, då kan vi välja en slumptal r med hög min-entropi och användning $H(r \mid x)$ som åtagande, som vi såg i kapitel 1.

Grundtanken är att vi kan publicera bara hash $H(r \mid x)$ vid tiden T, och sedan någon gång senare vi kan avslöja r och x. Vem som helst kan titta på loggen med endast tillägg och vara övertygad om att vi måste ha det

---

kända x vid den tidpunkt vi publicerade $H(r \mid x)$, eftersom det inte finns någon annan möjlig väg för att ha genererat
den datan.

Tillämpningar av tidsstämpling. Vad kan vi göra med den här typen av säkra tidsstämpling? Ett möjlig användning är att bevisa förkunskaper om någon idé. Anta att vi ville bevisa att vissa uppfinning som vi lämnade patent på var faktiskt i våra huvuden mycket tidigare. Vi skulle kunna göra detta genom att publicera
hashen av ett designdokument eller schematisk när vi först tänkte på uppfinningen - utan avslöjar för någon vad tanken är. Senare, när vi lämnar in vårt patent eller när vi publicerar idé, vi kan publicera originaldokumenten och informationen och vem som helst kan se bakåt i tiden och bekräfta att vi måste ha känt till det tidigare när vi publicerade åtagandet om det.

Vi kan också bevisa att någon annan har fått ett meddelande vi skickat till dem. Antag att Alice anställer Bob för att utföra ett programmeringsjobb; deras kontrakt kräver att Bob lämnar in sitt arbete till Alice av en specifik tid. Båda parter vill se till att om det blir en tvist senare om huruvida Bob lämnat in arbetet eller om koden utförs enligt specifikation, de har bevis på vad som var lämnas in och när. För att säkerställa detta kan de ömsesidigt komma överens om att publicera en hash av Bobs inskickade
arbete undertecknat av båda parter. Om endera parten senare ljuger om vad som lämnades in eller när, den andra part kan bevisa att de har fel (säg i en skiljedomstol) genom att avslöja input till hashen.

Många andra intressanta saker kan byggas bara från säker tidsstämpling. Det finns till och med en hel public-key signaturschema (kallat Guy Fawkes signaturschema) som bara använder hashfunktioner och en logg som endast kan läggas till. Det kräver inte någon av den tunga kryptografi som vanligtvis används för offentliga nyckelsignaturer.

Angrepp på Proofs-of- "Clairvoyance". En sak som vi *inte kan* göra med säker tidsstämpling ensam - även om det skulle vara mycket trevligt om vi kunde - är att bevisa *klärvoajans* , förmågan att förutsäga framtiden. Detta kan tyckas möjligt. Tanken skulle vara att publicera ett åtagande om en beskrivning av en händelse som är på väg att inträffa (som resultatet av ett sportevenemang eller ett val) och sedan avslöja den informationen för att bevisa att vi förutspådde händelsen i förväg. Men fungerar detta?

I slutet av 2014, under den sista matchen av VM, använde någon den här metoden för att "bevisa" det FIFA, organisationen som driver fotbolls-VM, var korrupt. Efter att matchen var över, en Twitter konto fick stor uppmärksamhet för att ha twittrat om flera händelser som inträffade under spelet, tidsstämplas *innan matchen ens började* . Till exempel twitrade den korrekt det Tyskland skulle vinna i förlängning och att Mario Götze skulle göra mål. Tydligen bevisar detta det heller ägaren till detta Twitter-konto kunde berätta om framtiden eller att matchen var riggad. Men i själva verket konto hade twittrade *alla möjliga utfall* innan matchen började. För varje spelare som är involverad i matchen kom det en tweet som förutspådde att han skulle göra mål; det fanns en tweet för alla tänkbara slutresultatet av spelet; och så vidare (se figur 9.1). Innan matchen slutade, alla falska förutsägelser togs bort, vilket lämnade Twitter-kontot med endast sanna "förutsägelser".

Samma grundläggande attack kan utföras mot vilket säkert tidsstämplingssystem som helst. Du binder dig helt enkelt
till en mängd möjliga resultat, och sedan bara avslöja de åtaganden som visar sig vara sanna. Detta

---

innebär att om du faktiskt *göra* har förmågan att förutsäga framtiden och vill bevisa det, måste du bevisa att du är tidstämpling *en specifik förutsägelse* snarare än att flera förutsägelser. Om du publicerar hashbaserade åtaganden är detta svårt att göra. Detta gäller särskilt i Bitcoin, eftersom vårt säkra tidsstämplingssystem binder inte åtaganden till någon individs offentliga identitet. Om du avslöja dem inte, det är lätt att publicera ett stort antal åtaganden och de du aldrig avslöjar kan inte lätt spåras tillbaka till dig.

Figur 9.1: Ett Twitter-konto som försökte "bevisa" att 2014 FIFA World Cup final riggades genom att "förutspå" matchens resultat. Den första, tredje och fjärde tweeten slutade eftersom det är sant, raderades resten efter matchen.

Fäst tidstämpling gammaldags sätt. Här är ett enkelt lågteknologiska sättet att göra säker tidsstämpling: publicera hash av dina data i en tidning eller något annat media som är allmänt ses av allmänheten, genom att köpa en annons. Arkiv över gamla tidningsnummer upprätthålls på bibliotek och online. Denna metod ger en hög grad av säkerhet att du kände till dessa uppgifter dagen då tidningen gavs ut. Senare, när du vill avslöja de uppgifter du begått, du kan till och med ta ut en andra annons för att publicera uppgifterna i samma tidning.

Figur 9.2 : **En tidsstämpling tjänst (Guard) som publicerar hashar i en dagstidning i stället än Bitcoin-blockkedjan.** Kunder till företaget kan betala för att få sina data inkluderade i tidsstämpel. Kom ihåg från kapitel 1 att vi kan använda Merkle-träd för att kapsla in många bitar av data i en enda hash och ändå effektivt bevisa att någon av dessa data är inkluderad i hashen.

Säker tidsstämpling i Bitcoin. Om vi vill använda Bitcoin istället för tidningar för tidsstämpling, var ska vi placera hash-åtagandet? Någonstans i en transaktion? Eller direkt i ett block?

Den enklaste lösningen (och den som folk kom på först) är istället för att skicka pengar till hashen av en offentlig nyckel, skicka den bara till hashen för dina data. Detta "bränner" dessa mynt, det vill säga gör dem oförbrukning och därmed förlorad för alltid, eftersom du inte känner till den privata nyckeln som motsvarar det adress. För att hålla kostnaderna nere skulle du vilja skicka ett mycket litet belopp, till exempel en satoshi (den lägsta möjliga transaktionsvärde i Bitcoin).

Även om detta tillvägagångssätt är mycket enkelt, är behovet av att bränna mynt en nackdel (även om mängden bränt är förmodligen försumbar jämfört med transaktionsavgifterna). Ett större problem är det Bitcoin-gruvarbetare har inget sätt att veta att transaktionsutmatningen är oanvändbar, så de måste spåra den evigt. Samhället rynkar på näsan åt denna metod av denna anledning.

En mer sofistikerad metod som kallas CommitCoin kan du koda dina data i *privata* nyckel. Kom ihåg att vi i kapitel 1 sa: "Med ECDSA är en bra källa till slumpmässighet viktig eftersom en dålig källa till slumpmässighet kommer sannolikt att läcka din nyckel. Det är intuitivt vettigt att om du använder dåligt slumpmässighet vid generering av en nyckel, kommer nyckeln som du genererar sannolikt inte att vara säker. Men det är en egenhet ECDSA att, även om du använder dålig slumpmässighet bara för att göra en signatur, med din perfekta nyckel, som också kommer att läcka din privata nyckel."

CommitCoin utnyttjar den här egenskapen. Vi genererar en ny privat nyckel som kodar vårt engagemang och vi härleder dess motsvarande publika nyckel. Sedan skickar vi en liten transaktion (på till exempel 2000 satoshi) till det adress, och skicka sedan tillbaka den i två bitar om 1000 satoshi vardera. Avgörande, när vi skickar tillbaka det använd samma slumpmässighet båda gångerna för att underteckna transaktionen. Detta gör att alla tittar på blockkedja för att beräkna den privata nyckeln, som innehåller åtagandet, med hjälp av de två signaturerna.

Jämfört med att koda ditt engagemang i den offentliga nyckeln, undviker denna CommitCoin behovet av att bränna mynt och för gruvarbetare att spåra en outnyttjad produktion för alltid. Det är dock ganska komplicerat.

Unspendable utgångar. Från och med 2014, är det föredragna sättet att göra Bitcoin tidsstämpling med en OP_RETURN transaktion som resulterar i en bevisligen outnyttjad produktion. OP_RETURN-instruktionen returnerar omedelbart med ett fel så att detta skript aldrig kan köras framgångsrikt, och data du inkluderar ignoreras. Som vi såg i kapitel 3 kan detta användas både som bevis på brännskador och för att koda godtyckliga data. Från och med 2015 tillåter OP_RETURN 80 byte data att pushas, vilket är mer än tillräckligt för en hashfunktionsutgång (32 byte för SHA-256).

OP_RETURN <H(data)>

Figur 9.3: Ett bevisligen "oförbrukbart" transaktionsutdataskript som bäddar in ett dataåtagande

Denna metod undviker uppsvällning i den outnyttjade transaktionsutmatningen eftersom gruvarbetare kommer att beskära OP_RETURN
utgångar. Kostnaden för ett sådant åtagande är i huvudsak kostnaden för en transaktionsavgift. Genom hela 2014 är en typisk transaktionsavgift mindre än ett öre. Kostnaden kan minskas ytterligare genom att använda en enda engagemang för flera värden. I slutet av 2014 finns det redan flera webbplatstjänster som hjälp med detta. De samlar ett gäng åtaganden från olika användare och kombinerar dem till en stort Merkle-träd, som publicerar en outnyttbar produktion som innehåller Merkle-trädets rot. Detta fungerar som en
engagemang för all data som användare ville tidsstämpla den dagen.

Olagligt innehåll. En nackdel med att kunna skriva godtycklig data i blocket kedjan är att människor kan missbruka funktionen. I de flesta länder är det olagligt att inneha och/eller distribuera vissa typer av innehåll, särskilt barnpornografi, och påföljder kan vara stränga. Upphovsrättslagar begränsar också distribution av visst innehåll.

Visst har flera personer försökt göra saker som detta för att "sörja" (dvs för att trakassera eller irritera) Bitcoin-gemenskap. Till exempel har det förekommit rapporter om länkar till pornografi publicerade i Bitcoin blockkedja. Målet med dessa sörjande är att göra det farligt att ladda ner blockkedjan på din hårddisk och att köra en full nod, eftersom att göra det kan innebära att lagra och överföra material vars innehav eller spridning är olagligt.

Det finns inget bra sätt att hindra människor från att skriva godtyckliga data i bitcoin-blockkedjan. Ett möjlig motåtgärd är att bara acceptera *Pay-to-Script-Hash* transaktioner. Detta skulle göra det lite lite dyrare att skriva i godtyckliga data, men det skulle ändå inte förhindra det direkt.

Lyckligtvis är lagen inte en algoritm. Det är frestande att försöka "hacka" lagen med tekniska medel för att ger oväntade eller oavsiktliga resultat, men detta är inte lätt. Lagar är avsedda att tolkas av människor och inkluderar faktorer som avsikt. Till exempel, US Code 2252, avsnittet i US federal lag som hänför sig till innehav, distribution och mottagande av barnpornografi, använder Formuleringen ” *medvetet* besitter, eller *medvetet* accesser med avsikt för att visa” när man beskriver förbjudet aktiviteter (betoning vår).

Det är också värt att notera att på grund av storleksbegränsningarna vi diskuterade ovan, data som bilder (förutom kanske små) kan inte skrivas direkt in i Bitcoin-blockkedjan. Det kommer de heller måste vara värd externt, med endast länkar inskrivna i blockkedjan, eller vara kodade i en besvärligt sätt över flera transaktioner. Slutligen, de flesta Bitcoin-klienter skickar inte med förmågan att avkoda och visa data som skrivits in i transaktioner, än mindre data som är kodad över flera transaktioner.

Overlay valutor. På den positiva sidan, eftersom vi *kan* skriva vad data vi vill ha till Bitcoin, vi kan också bygga ett helt nytt valutasystem *ovanpå Bitcoin* utan att behöva utveckla en ny

konsensusmekanism. Vi kan helt enkelt använda Bitcoin som det finns idag som en logg som endast kan läggas till och skriva
all data som vi behöver för vårt nya valutasystem direkt in i Bitcoin-blockkedjan. Vi ringer
detta tillvägagångssätt en "överlagringsvaluta". Bitcoin fungerar som det underliggande substratet, och data för överlagringsvaluta skrivs in i Bitcoin-blockkedjan med outnyttjade transaktionsutdata.

Naturligtvis kommer Bitcoin gruvarbetare faktiskt inte *bekräfta* vad du skriver i blocket kedjan eftersom de vet inte (och bryr dig inte!) om uppgifterna du skriver är giltiga enligt reglerna för din nya valuta. Vem som helst kan skriva vad som helst där som är villig att betala transaktionsavgifterna för Bitcoin. Istället måste du utveckla mer komplicerad logik för att validera transaktioner i den nya valutan, och denna logik måste finnas i varje slutanvändarklient som deltar i att skicka eller ta emot denna valuta.

Till exempel, i en överlagringsvaluta kan gruvarbetare inte längre avvisa dubbla utgifter. Istället, varje användare av överlagringsvalutan måste titta på historien om vad som har skrivits i blockkedjan. Om en överläggstransaktionen försöker spendera ett överläggsmynt som redan har spenderats, sedan den andra transaktionen bör helt enkelt ignoreras. Av denna anledning finns det inget sådant som en lätt SPV-klient för överlagringsvalutor.

Motpart är en framträdande överlagringsvaluta. Alla motpartstransaktioner skrivs in i Bitcoin blockkedja. Under 2014 genomfördes mellan 0,5 % och 1 % av alla Bitcoin-transaktioner Motpartsdata. Den stöder också en mycket större och rikare funktionsuppsättning än Bitcoin. Tanken är den eftersom Motparten inte behöver utveckla en ny konsensusalgoritm, och eftersom Bitcoin-gruvarbetare behöver inte känna till Motpartsreglerna, de kan istället fokusera på att utveckla intressant funktioner som smarta kontrakt, användardefinierade valutor och mer. Motpartens API kan vara mycket större än Bitcoin API eftersom Bitcoin-gruvarbetare inte behöver förstå det eller godkänna det.

Potentialen att utveckla en ny valuta utan att behöva skapa ett nytt konsensussystem är mycket lockande. Du behöver inte ens uppmuntra nya gruvarbetare att gå med i ditt system, och du kan lägga till nya

---

funktioner utan att behöva ändra Bitcoin. Sådana system är dock fortfarande beroende av Bitcoin - för till exempel är de föremål för samma avgiftskrav som andra Bitcoin-transaktioner. Detta tillvägagångssätt kan också vara ineffektivt, eftersom noder på överlagringsvalutan kan behöva bearbeta mycket data, eftersom Bitcoin-noder filtrerar inte dessa transaktioner åt dig.

9.2 Bitcoins som "smart egendom"

Nu ska vi prata om att använda bitcoins för att representera något annat än en valutaenhet i Bitcoin systemet.

Minns från kapitel 6 att du kan spåra ägande av värde i Bitcoin-systemet över tid, helt enkelt genom efter transaktionsdiagrammet. Tänk på varningen: det finns inget sådant som ett "bitcoin" i sig - bara outnyttjade transaktionsutdata, som vi kallar mynt. Varje bitcoin har en historia som vem som helst kan se i blockkedjan. Ett mynts historia går ända tillbaka till en eller flera myntbaser transaktioner där mynt ursprungligen präglades. Som vi nämnde tidigare är detta dåligt för anonymitet, eftersom du ofta kan spåra ägande av mynt på detta sätt.

Utbytbarhet . Detta leder också till en intressant iakttagelse: Bitcoins inte *utbytbara* . Inom ekonomi, a

fungibel goda är en där alla individuella enheter är likvärdiga och kan ersättas med varandra. Till exempel är guld fungibelt eftersom ett uns (rent) guld kan ersätta vilket annat uns av guld. Men detta är inte alltid sant för Bitcoin eftersom varje bitcoin är unik och har en annan historia.

I många sammanhang kanske denna historia inte spelar någon roll, men om historien är meningsfull för någon du vill handla med, kan det betyda att din 1.0 bitcoin inte är samma som deras 1.0 bitcoin. Kanske de skulle inte vara villiga att byta ut deras med ditt eftersom de föredrar historien om deras mynt framför det av ditt mynt. Till exempel, precis som myntsamlare värderar gamla mynt, kan bitcoinsamlare en dag göra det sätt särskilt värde på mynt som har sitt ursprung i genesis-blocket eller något annat tidigt block i Bitcoins historia.

Smart Property. Kan detta icke-utbytbarhet egendom vara *användbar* ? Vi har redan sett varför det kan vara dåligt för integritet på grund av möjligheten att avanonymisera användare. I det här avsnittet ska vi titta på varför det kan också vara bra att ge *mening* till historien av en Bitcoin.

Låt oss fundera på vad det skulle innebära att ge mening till historien om vanliga fysiska offline valuta. Anta att vi ville lägga till metadata till offlinevalutan. Faktum är att vissa människor redan gör det detta. Till exempel gillar de att skriva olika meddelanden på sedlar, ofta som ett skämt eller ett politiskt protest. Detta påverkar i allmänhet inte sedelns värde och är bara en nyhet.

Men tänk om vi kunde ha *autentiseras* metadata fäst vår valuta - metadata som inte kan lätt dupliceras? Ett sätt att uppnå detta är att inkludera en kryptografisk signatur i metadata vi skriver, och knyta denna metadata till *serienumret* på sedeln.

Figur 9.4: Ett exempel på att lägga till användbar metadata till vanliga sedlar

Vad skulle detta kunna användas till? Säg att ett basebolllag vill använda dollarsedlar som biljetter. På så sätt nej längre måste gå igenom besväret med att skriva ut sina egna biljetter och se till att ingen kan skriva ut förfalskade biljetter. New York Yankees kunde helt enkelt hävda att dollarsedeln med en specifik serienumret representerar nu en biljett till ett specifikt spel och i en specifik plats. Dessa dollarsedlar skulle fördelas på samma sätt som pappersbiljetter normalt distribueras, såsom genom att vara skickas till fansen när de köper biljetter online. Den som har den lappen har rätt att gå in i stadion, sitt på den tilldelade sätet och titta på matchen, utan några andra frågor. Sedeln själv är biljetten!

För att lägga till autenticitet kunde jänkarna använda digitala signaturer. De skulle skriva under ett meddelande som innehåller specifikt speldatum, platsnummer och serienummer på notan - och stämpla meddelandet och signaturen på räkningen. En 2D-streckkod skulle vara en bekväm form för denna data (se figur 9.4). Alternativt kan arenan upprätthålla en databas som listar serienummer och motsvarande platsnummer för varje spel. De kunde kontrollera databasen efter denna information när du försökte

gå in i porten. Detta undviker behovet av att stämpla sedlarna.

Vad köper detta oss? Nu kan valuta representera många saker. Förutom exemplet på en sport biljett, det finns många andra applikationer. Vi ärver egendomen mot förfalskning som sedlar redan har. Regeringar arbetar mycket hårt för att se till att det är svårt att duplicera en sedel! Också, sedelns underliggande valutavärde bibehålls. Efter att fansen har löst in sin biljett, sedel är perfekt användbar som vanlig valuta. Det kan vara ett problem om alla vill fysiskt stämpla metadata på valuta, men detta problem försvinner om vi använder databasmetoden.

Naturligtvis är alla användbara betydelsen av denna nya metadata bara så bra som vår tillit till *utgivaren* som signerade den. Någon måste veta att det finns en specifik nyckel som används för att signera giltiga Yankees-biljetter — eller
ladda ner Yankees databas — för att känna igen dess värde som en biljett. Till någon annan, det

---

Sida 46

skulle bara se ut som en dollarsedel. Men det är okej. Det är faktiskt en önskvärd egenskap, eftersom en gång biljetten
har uppfyllt sitt syfte kan den återgå i omlopp som en vanlig dollarsedel.

Färgade mynt . Kan vi göra detta digitalt ovanpå Bitcoin? Vi skulle vilja behålla Bitcoins trevliga funktioner såsom
som förmågan att göra transaktioner online, snabb transaktionsavveckling och att man inte litar på en bank.

Figur 9,5:. Färgade mynt Transaktionen visade diagrammet illustrerar emission och utbredning av färg

Huvudidén är att stämpla några Bitcoins med en "färg" och spåra den färgstämpeln även som myntet byter ägare, precis som vi kan stämpla metadata på en fysisk valuta. En bitcoin stämplad med en färg fungerar fortfarande som en giltig bitcoin, men bär dessutom denna metadata.

Sida 47

För att uppnå detta, i en transaktion, kallad "emissionstransaktionen", infogar vi lite extra metadata som förklarar att några av utgångarna har en specifik färg. Ett exempel illustreras i figur 9.5. I en transaktion ger vi ut fem "lila" bitcoins i en transaktionsutgång, medan den andra utgången fortsätter att vara normala ofärgade bitcoins. Någon annan, kanske med en annan signeringsnyckel, har problem "gröna" bitcoins i en annan transaktion. Vi kallar dessa färger för intuitivitet, men i praktiken färger är bara bitsträngar. Den enda egenskapen som spelar roll är att mynt av samma färg och samma värde är det likvärdig.

Nu har vi bitcoins med olika färger kopplade till dem. Vi kan fortfarande göra alla vanliga saker vi gör med bitcointransaktioner. Vi kan ha en annan bitcoin-transaktion som kräver flera ingångar: några gröna mynt, några lila mynt, några ofärgade mynt, och blandar runt dem. Det kan det ha vissa utgångar som bibehåller färgen. Det kan behöva finnas lite metadata i transaktion för att avgöra vilken färg som går till vilken transaktionsutgång. Vi kan dela upp en transaktion utmatning av fyra gröna mynt till två mindre gröna mynt. Senare kunde vi kombinera flera gröna mynt till ett stort grönt mynt.

OpenAssets . Från och med 2015 är det mest populära förslag för att genomföra detta i Bitcoin kallas OpenAssets. Tillgångar utfärdas med en speciell Pay-to-Script-Hash-adress. Om du vill utfärda färgade mynt väljer du först en P2SH-adress att använda. Alla mynt som överförs via den adressen och kommer in utan färg lämnas med den färg som anges av den adressen. För att detta ska vara meningsfullt, du måste publicera den adressen någonstans. Det finns olika börser som spårar vilka adresser ger mynten vilka färger. Eftersom mynt sekventiellt kan passera genom mer än ett färgutgivande adress kan de ha mer än en färg, och det är bra.

Varje gång du har en transaktion som involverar färgade mynt måste du sätta in en speciell markör produktion. Detta är en bevisligen outnyttjad utdata, liknande vad vi använde för tidsstämpling av data åtaganden. Metadata som är inbäddade i markörutgången kodar detaljer om hur inkommande färgvärde ska delas mellan de olika utgångarna.

Som vi noterade tidigare är detta kompatibelt med Bitcoin. Eftersom det inte kräver att du ändrar Bitcoin, grupp av gruvarbetare tenderar att inte avskräcka eller störa dessa system. Det tillåter vem som helst att deklarera vilken färg de vill utan att behöva fråga en central myndighet om rätt att utfärda färgade mynt. Om det finns andra som förstår och håller sig till den betydelse du tillskriver färgen du problem kan dina färgade mynt få ytterligare värde utöver deras nominella bitcoinvärde. Till exempel, om Yankees ger ut färgade mynt, kommer dessa mynt att kunna fungera som biljetter till ett spel som tillhandahålls
stadionoperatörerna förstår deras innebörd och släpper in dig baserat på biljetter med färgade mynt.

En nackdel med detta schema är att vi måste lägga in den outnyttjade markörutgången i varje transaktion. Detta lägger till lite extra kostnader, eftersom vi måste förlora lite pengar varje gång vi vill handla ett färgat mynt. En andra nackdel är att gruvarbetare inte kontrollerar giltigheten av färgade mynt, bara de underliggande bitcoins. För att verifiera att ett färgat mynt du får är giltigt måste du kontrollera *Hela transaktionshistorik* att myntet var inblandad i, eller lita på en tredje part att göra kontrollen för

---

du. I synnerhet kan du inte använda en tunn SPV-klient som du kan för vanliga Bitcoin. Det gör det svårare att använda färgade mynt på beräkningsmässigt begränsade enheter som mobiltelefoner.

Användning av färgade mynt och smarta egendom . *Stock i ett företag.* En citerade ofta motivation för smart egendom är aktie i ett företag. Ett företag som vill ge ut färgade mynt som lager skulle göra publicera dess utfärdande adress, och bitcoins som är färgade med denna adress fungerar som aktier. Ett satoshi kan representera en aktie i företaget. Aktieägare kan sedan handla aktien på blockkedja utan att behöva en centraliserad mellanhand som en börs. Naturligtvis aktieägare kommer att behöva lita på att bolaget kommer att hedra aktierna. Till exempel kan företaget lova att dela ut utdelningar proportionellt till varje aktie eller för att ge aktieägarna rösträtt i bolagets beslut. Med traditionella aktier upprätthålls dessa löften lagligt. Från och med 2015, färgade mynt eller andra blockkedjebaserade tillgångar har inte juridiskt erkännande i någon jurisdiktion.

Fysisk egendom. En annan potentiell användning är att färgade mynt kan utgöra ett anspråk på en del verklig egendom. Ett färgat mynt kan till exempel förknippas med ett hus eller en bil. Kanske du har en sofistikerad bil som faktiskt spårar ett specifikt färgat mynt på blockkedjan, och startar och kör automatiskt för alla som äger det färgade myntet. Då kan du sälja din bil,

eller åtminstone överföra kontrollen över det, helt enkelt genom att göra en enda transaktion i blockkedjan. Vi får se in
Kapitel 11 hur detta potentiellt kan implementeras såväl tekniskt som det sociala och juridiska hinder för att få det att hända. Men drömmen om färgade mynt och smart egendom är att någon verklig egendom skulle kunna representeras i Bitcoins värld och överföras eller handlas lika enkelt som bitcoins själva.

Domännamn . Som ett sista exempel, överväg att använda färgade mynt för att utföra några av funktionerna det befintliga domännamnssystemet: spåra ägande och överföring av internetdomännamn som samt kartläggning av domännamn till IP-adresser. Domännamnsmarknaden har en mängd olika intressanta egenskaper: det finns ett potentiellt oändligt antal namn, dessa namn har brett olika värden baserat på deras minnesbarhet och andra faktorer, och samma namn kan ha mycket olika nytta för olika människor. Det är möjligt att använda färgade mynt för att hantera domännamn registrering och de funktioner vi listade. Men att stödja denna applikation har också varit i fokus av ett framträdande altcoin som heter Namecoin, som vi kommer att titta på i detalj i nästa kapitel. Varje tillvägagångssätt har fördelar: med färgade mynt får du säkerheten för Bitcoins blockkedja medan med en altcoin är det lättare att implementera den komplexa logiken som behövs för ägande av domännamn, överföring,
och IP-adressmappning.

9.3: Säkra flerpartslotterier i Bitcoin

Nu ska vi prata om att vara värd för ett "coin flip"-spel i Bitcoin. Återigen, vi börjar med att beskriva offlineversionen av det vi försöker bygga.

Alice och Bob vill satsa fem dollar. De båda går med på vadet i förväg och metoden för avgöra vinnaren. Bob kommer att slå ett mynt i luften, och medan det roterar ropar Alice "Heads" eller

---

"Svansar". När myntet landar har de båda omedelbart en klar förståelse för vem som vann vadet, och de båda har försäkran om att resultatet var slumpmässigt och att ingen av dem kunde påverka resultatet.

Stegsekvensen i den här ceremonin samt myntslagningens fysik spelar en avgörande roll i övertyga båda parter om att spelet är rättvist. En brist med detta system är att båda parter måste vara närvarande på samma plats vid samma tidpunkt. Dessutom måste båda parter fortfarande lita på det den som förlorar får betala. I onlinevärlden skulle vi vilja kunna ha ett lotteri som är precis som "rättvist", men löser också problemet med att se till att förloraren betalar.

Till en början kan detta verka som en ganska märklig och begränsad tillämpning för att studera i detalj. Roligt nog är Bitcoin-baserade vadslagningstjänster som Satoshi Dice – som förlitar sig på en pålitlig part, till skillnad från
systemet vi skulle vilja designa — har visat sig vara mycket populärt, ibland representerar det en stor del av det alla Bitcoin-transaktioner på nätverket.

Den verkliga anledningen till att vi vill studera kryptografisk myntvändning är dock att det visar sig att om vi kan designa ett säkert protokoll för det, kan vi använda dessa tekniker för att bygga många andra intressanta och användbara protokoll. Kryptografer studerar "säker flerpartsberäkning" där två eller flera ömsesidigt

otillförlitliga parter har var och en vissa data och vill beräkna ett resultat som beror på alla deras data, men utan att avslöja uppgifterna för varandra. Tänk på en auktion med förseglat bud, men utan en pålitlig auktionsförrättare. Ofta måste dessa beräkningar randomiseras, till exempel för att bryta banden. Slutligen, vi
kanske vill resultatet av *beräkningen* för att fastställa en *monetär* utfallet i ett oåterkalleligt sätt. Kanske vill vi säkerställa att den vinnande budgivaren i auktionen betalar säljaren; kanske vi till och med vill säkerställa att säljarens (smarta) egendom som auktioneras ut automatiskt överförs till vinnande budgivare. Alternativt kanske vi vill straffa parter om de avviker från protokollet.

Med andra ord, ett säkert flerpartslotteri är en enkel miljö att studera ett extraordinärt kraftfullt paradigm: ömsesidigt opålitliga deltagare med känsliga input som tillsammans kör ett program som har makten att manipulera inte bara bitar utan också pengar.

Slantsingling Online. Den första utmaningen är att ersätta ”singla slant” mekanism med vissa online likvärdig. Låt oss säga att vi nu har tre parter, Alice, Bob och Carol, som alla vill välja en nummer 1, 2 eller 3 med lika stor sannolikhet. Här är ett försök till ett sådant protokoll. Var och en av dem väljer en
stort slumptal — Alice väljer x, Bob y och Carol Z. De berättar sina siffror för varandra och de beräknar utdata som (x + y + z) % 3.

Om alla valde sina slumptal oberoende av varandra skulle detta verkligen fungera. Men kom ihåg att vi gör det här över internet, och det finns inget sätt att insistera på att de alla skickar sina nummer "samtidigt". Alice kanske väntar tills hon hör Bobs och Carols nummer innan hon sänder hennes. Om hon gör detta kan du se hur det är trivialt för henne att göra den slutliga produktionen vad hon vill. Vi kan inte utforma protokollet för att övertyga alla parter om att ingen av de andra partierna fuskat.

För att lösa detta problem kan vi återigen använda hash-åtaganden. Först väljer var och en av dem en stor slumpmässigt nummer och publicerar en hash av detta nummer. När detta är gjort avslöjar var och en av dem nummer de valde. De andra kontrollerar sedan att de avslöjade siffrorna hash till de värden som publicerats i det första steget, och beräkna det slutliga resultatet från de tre slumptal som visas i figur 9.6.

Omgång 1:
Varje part väljer en stor slumpmässig sträng — Alice väljer x, Bob väljer y och Carol väljer z.
Parterna publicerar H(x), H(y), H(z) respektive.
Varje part kontrollerar att H(x), H(y), H(z) alla är distinkta värden (annars avbryter protokollet).
Runda 2:
De tre partierna avslöjar sina värden, x, y och z.
Varje part kontrollerar att de avslöjade värdena överensstämmer med hasharna som publicerades i omgång 1.
Resultatet är (x + y + z) % 3.

Figur 9.6: Använda hash-åtaganden för att implementera en rättvis slumptalsgenerator.
Detta protokoll kan enkelt utökas för att stödja valfritt antal parter.

Anledningen till att detta protokoll fungerar är tvåfaldigt. För det första, eftersom hash-ingångarna x, y och z är stora slumpmässiga
siffror kan inget parti förutse de andras ingångar efter första omgången. För det andra, om (säg) Alice väljer hennes input slumpmässigt enligt protokollet, kan hon vara säker på att den slutliga utmatningen kommer att vara slumpmässig
oavsett om eller inte Bob och Carol välja sina ingångar slumpmässigt.

Rättvisa . Vad händer om någon misslyckas med att avslöja sitt engagemang? I omgång 2 av protokollet,
Anta att Carol väntar tills Alice och Bob har avslöjat sina hemligheter. Carol, innan hon avslöjar sin,
inser att hon kommer att förlora om hon gör det. Så hon kanske vägrar att publicera sitt slumpmässiga nummer - hon
kan påstå sig ha glömt det eller låtsas vara offline. Alice och Bob skulle förmodligen vara misstänksamma, men de skulle inte ha någon bra utväg.

Vad vi skulle vilja är ett system där den som gör ett åtagande tvingas avslöja det inom vissa
tidsgräns. I kryptografi kallas denna egenskap för rättvisa. Bitcoin ger oss en utmärkt
mekanism för detta.

Låt oss säga att Alice vill göra ett tidsbestämt åtagande, och Bob är den enda andra personen som är det bekymrad över det. Först lägger Alice upp en obligation, i form av ett Bitcoin-transaktionsutmatningsskript som anger att den kan användas på ett av två sätt. Ett sätt är med en undertecknad transaktion från både Alice
och Bob. Det andra sättet att spendera det är med en signatur från just Alice, men bara om hon också avslöjar henne
slumpmässigt nummer. Om Alice slumptalssträng är **x** , sedan scriptPubkey faktiskt innehåller värdet **H (x)** .

Därefter undertecknar Alice och Bob båda en transaktion som betalar obligationen till Bob (vilket är ett av de två sätten att
kan spenderas). Varför skulle Alice gå med på detta? Transaktionen medför en ***nLockTime*** värde som

---

garanterar Bob kan inte göra anspråk lånet före en tid **t** . Eftersom Alice planerar att avslöja hennes engagerade värde innan dess och återfå obligationen, är det säkert för henne underteckna denna transaktion.

Nu om Alice lämnar utan att avslöja sitt värde, kan Bob anspråk på bindningen vid tidpunkten **t** . Detta inte *kraft* Alice att avslöja hennes engagemang, men hon *kommer att* förlora hela band som hon satte upp. Så garantin att hon kommer att avslöja sitt hemliga värde beror på hur mycket pengar hon är villig att lägga i obligationen.

scriptPubKey:
OP_IF
<AlicePubKey> OP_CHECKSIGVERIFY <BobPubKey> OP_CHECKSIG
OP_ELSE
<AlicePubKey> OP_CHECKSIGVERIFY OP_HASH <H(x)> OP_EQUAL
OP_ENDIF

scriptSig för fall 1:
<BobSignature> <AliceSignature> 0
scriptSig för fall 2:
x <AliceSignature> 1

Figur 9.7: Transaktionsutgångens scriptPubkey och scriptSigs som används i ett tidsbestämt hashåtagande.

Hur kan vi använda detta tidsinställda hash-åtagande för att genomföra vårt säkra lotteri? Vi kommer att ha nästan
exakt samma struktur som tidigare, förutom i stället för att använda de enkla hash-åtagandena, kommer vi att använda dessa
tidsinställda åtaganden. Den som inte avslöjar sitt slumpmässiga värde före deadline kommer att förlora en deposition som används för att kompensera de andra två spelarna. Att avslöja det slumpmässiga värdet är nu helt enkelt en fråga om att återvinna band genom att ge den rätta hemliga ingången **x** .

Detta lotterischema kan implementeras ovanpå Bitcoin. Men det är lite komplicerat och tidsbestämt hashåtaganden kräver flera icke-standardiserade transaktioner. När det finns **n** parter i lotteri, **n** 2
åtaganden behövs eftersom varje part måste sätta upp en förbindelse för varandra.
Spelarna måste deponera mer pengar totalt än de ens satsar. Men det är rimligt att a få deltagare, och det finns varianter med bättre effektivitet. Viktigast av allt, det tjänar som ett existensbevis på att till synes omöjliga protokoll som att vända ett virtuellt mynt över internet och att straffa en part för att ha avbrutit protokollet är möjligt i Bitcoin-världen.

9.4: Bitcoin som offentlig slumpmässig källa

I det sista avsnittet visade vi hur en grupp människor tillsammans kan välja ett rättvist slumpmässigt värde. I denna
avsnitt ska vi prata om hur du använder Bitcoin att generera slumpmässiga värden som är rättvist att *någon* i allmänheten.

Varför skulle vi vilja detta? Låt oss diskutera några exempel på applikationer som redan är beroende av allmänheten
källor till slumpmässiga värden.
250

Sida 52

NBA utkast lotteri. Ett exempel som förekommer varje vår i USA är NBA utkast lotteri. Alla 30 lag i NBA träffas och väljer slumpmässigt — med viss viktning baserat på hur varje lag spelade under föregående säsong — ordningen i vilken lagen får välja de bästa amatörspelarna i landet som är redo att bli professionell. Detta gjordes första gången 1985. Lotteriet var genomfördes över direktsänd tv, och involverade att plocka kuvert efter att de blandats i en transparent snurrande trumma. Det här lotteriet skapade lite kontrovers då, eftersom New York Knicks vann det första året och kunde utarbeta den mycket eftertraktade centern Patrick Ewing (en eventuell medlem från Basketball Hall of Fame). Sedan lotteriet filmades i New York City, några fans av andra team påstod att processen var riggad till förmån för Knicks.

Det finns många konspirationsteorier för hur NBA kan ha riggat den här processen, till exempel berömda teorin om "böjt hörn" som tyder på att Knicks kuvert hade sitt hörn böjt så att kommissionären kunde skilja den från de andra genom beröring. En annan teori antyder att Knicks kuvertet förvarades i en frys och kommissarien tog helt enkelt tag i det ena kalla kuvertet. Dessa teorier visar varför det är väldigt svårt att hålla en sådan här teckning och bevisa att den var rättvis — det finns

många rimliga sätt att fuska. Tänk bara på vilka professionella trollkarlar som kan dyka upp
att göra! Än idag sker detta lotteri varje år och varje gång leder det till en mängd olika konspirationer
teorier och rykten om att lotteriet inte är en rättvis slumpmässig dragning.

Bild **9.8:** Bilder från 1969 (Vietnamkriget) militära draglotteriet.

USA: s militära utkast lotteri. Ett allvarligare exempel kommer från 1969, då det fanns en värnplikt
lotteri i USA för att avgöra vilka unga män som skulle behöva gå med i de beväpnade
tjänster. De flesta av dem skickades för att slåss i Vietnamkriget. Ett förfarande som liknar NBA-lotteriet
användes, utfördes av flera representanter från den amerikanska kongressen och sändes i direktsändning
TV (Figur 9.8). De dumpade små kapslar märkta med varje dag på året i en stor
plasttrumma, och turades sedan om att sträcka sig in med händerna för att dra ut siffrorna. Män berättigade till
vara utarbetade fick ett prioritetsnummer baserat på den dag på året deras födelsedag inföll. De
prioritetsnummer avgjorde i vilken ordning de skulle utarbetas.

Detta utkast från 1969 var första gången detta lotteriförfarande användes i nationell skala. Målet var att
göra processen mer rättvis (genom att ta den ur händerna på tusentals lokala nämnder) och till
visa för allmänheten att det var en slumpmässig process. Tyvärr misslyckades lotteriet.
Inom en vecka märkte statistiker som tittade på data ett avvikande mönster (illustrerat i figur
9,9). Dagar sent på året fick låga dragsiffror. Även om avvikelsen är mycket subtil så är den det
statistiskt signifikant och mycket osannolikt att det har hänt på grund av slumpen. När de gick tillbaka till
granska banden visade det sig att de roterade trumman exakt ett jämnt antal gånger, så att
kapslarna som började på toppen tenderade att fortfarande vara på toppen. Det fanns inte tillräckligt med
blandning
gör det till en statistiskt slumpmässig dragning.

Figur 9.9: Statistisk snedvridning av 1969 års lotteriutkast. Årets dag (x-axel) kontra lottnummer
(y-axel).

Vad båda dessa exempel visar är att det är väldigt svårt att skapa offentlig slumpmässighet och övertyga
allmänheten att du faktiskt har gjort ett bra jobb. Det finns en risk att processen inte är riktigt
slumpmässigt och fritt från inflytande. Det finns också en risk att även om processen *är* slumpmässigt,
allmänheten kommer inte
tror dig.

Kryptografiska ”Beacons”. Offentlig visning av slumpmässighet med hjälp av ett hjul, vända mynt, kasta tärning,
och så vidare har varit så populära genom historien eftersom de är billiga och lätta att förstå.
Men de klarar sig inte så bra med storskaliga scenarier eftersom de är väldigt svåra för människor att granska.
Även om videon av förfarandet verkar legitim kan människor rimligen misstänka att
lotterikonduktören har utfört en del trick för att rigga processen.

Kan vi göra det bättre kryptografiskt? Låt oss använda termen "kryptografisk beacon" för att referera till en tjänst
som ger en offentlig källa till slumpmässighet. Tanken är att fyren ska publiceras kontinuerligt

nya slumpmässiga data i regelbunden takt som ingen kan förutse i förväg. Hoppas alla håller med
252

---

att det inte finns något sätt för någon att förutsäga vad beaconen kommer att skicka ut härnäst, så alla kan lita på det som ett rättvist slumpmässigt värde.

Om en perfekt kryptografisk ledstjärna fanns, skulle den kunna användas för vilket som helst av de offentliga lotterierna vi
tittade på. Även om du bara ville spela bingo på din lokala sociala klubb, skulle du inte behöva använda en stor trumma av siffror. Om alla litade på fyren skulle du spara mycket ansträngning jämfört med med hjälp av fysiska uppvisningar av slumpmässighet.

Kryptografer har föreslagit många andra tillämpningar av offentlig slumpmässighet, inklusive röstning system, nollkunskapsbevis och skär-och-välj-protokoll. Många av dessa kan göras mycket enklare och mer effektivt om du har en perfekt kryptografisk ledstjärna. Tyvärr har vi inte gjort det hittat ett perfekt sätt att implementera en sådan fyr än.

NIST ledstjärna . National Institute of Standards and Technology (NIST) har sedan 2011 drivit sitt egen fyrtjänst. De påstår sig generera sina slumpmässiga siffror genom ett komplicerat laboratorium uppställning som involverar två intrasslade fotoner. Tanken är att ge starka garantier för att siffrorna är det slumpmässiga eftersom de genereras från ett kvantmekaniskt fenomen. Om du accepterar Heisenbergs osäkerhetsprincip och andra allmänt ansedda fysiklagar, då borde detta verkligen vara slumpmässigt och oförutsägbart. Tjänsten är inställd så att den producerar ny slumpdata var sextionde sekunder tillsammans med en digital signatur över datan. NIST beacon ger ett bekvämt gränssnitt för programmatiska applikationer: siffrorna kan enkelt läsas ut från ett webbflöde.

Denna kvantmekaniska procedur är i någon mening "gränsen" för fysiska uppvisningar av slumpmässighet. Men det gör ingenting för att lindra det väsentliga problemet med tillit - du måste lita på att NIST faktiskt är det genomföra förfarandet som de hävdar. Du måste lita på det någonstans i en byggnad i Maryland NIST har sitt faktiska laboratorium som producerar dessa siffror och att de inte bara iscensätter procedur. Du måste också tro att de inte förbehåller sig möjligheten att medvetet skriva över några av de slumpmässiga värdena innan de publicerar dem.

Andra potentiella sätt att bygga en fyr: naturfenomen. Hur en alternativ metod där vi använder något naturfenomen som alla kan observera? Vi kanske kan använda detaljer om vädret, som vilken temperatur det kommer att vara imorgon på en specifik plats, eller hur hård vinden kommer att vara, eller om det kommer att regna eller inte. Naturligtvis har vi en viss förmåga att förutsäga
väder i förväg, men inte exakt, så kanske vi kan använda de "minst betydande bitarna" av uppmätta värden. Begränsningen här är att alla deltagare måste vara på samma plats för att få samma mått.

För att undvika detta kan vi vända oss till solfläckar, som är utbrott av aktivitet på solens yta. Annan exempel är kosmisk bakgrundsstrålning, som är brus som du kan lyssna på med en radioantenn från vilken punkt som helst på planeten; alla borde kunna läsa samma värde. Dessa är fenomen som händer i så stor skala att det är lätt att övertyga sig själv om att ingen kommer att göra det lyckas rigga processen. Det är långsökt att föreställa sig att någon skulle flyga en rymdfarkost mot solens yta för att på något sätt manipulera den bara för att rigga lite lotteri igen

Jorden. Så dessa tillvägagångssätt har flera goda egenskaper: offentlig observerbarhet, säkerhet mot manipulation och, i de flesta fall, en acceptabel nivå av oförutsägbarhet.

Ett problem med dessa tillvägagångssätt är att de är ganska långsamma. Till exempel, om din slumpmässiga signal är
den dagliga höga temperaturen, då får du bara en avläsning per dag. Det gör inte solens yta
byt för ofta. I många kryptografiska applikationer används slumpmässiga bitar som indata till en *pseudoslumpgenerator* (PRG). För att PRG ska vara säker måste ingången vara 80 bitar (eller mer) in längd. Det kan ta ett tag för 80 bitar av slumpmässighet att samlas med källor baserat på väder och astronomi.

Figur 9.10: NASA-bild av solfläckar.

Dessutom kräver det expertis för att mäta solfläckar, så du skulle verkligen behöva lita på några pålitliga observatör att publicera mätningarna. Det kan dock finnas många betrodda observatörer, och det kan vi hoppas att de skulle "hålla varandra ärliga". Applikationer som konsumerar beacons, eller användare av sådana applikationer, kunde välja vilken av observatörerna de skulle vilja förlita sig på. De kan också enkelt byta observatörer när som helst. Den här egenskapen kallas "trust agility" och är utan tvekan överlägsen att ha en singel enhet som NIST som producerar fyren.

Det finns ett djupare problem, ett som vid första anblicken kan verka trivialt. Hur gör vi en verklig värld observation - en temperatur, ett fotografi av solfläckar - till en sträng av bitar på ett sådant sätt att kommer varje observatör att få samma bitsträng? Vi skulle kunna försöka kvantisera mätningen: för Exempelvis skulle vi kunna uttrycka temperaturen i Fahrenheit och använda den första decimalsiffran som beacon-utgång. Men om inte varje observatörs termometer är orealistiskt exakt, kommer det att finnas tider när vissa observatörer kommer att läsa temperaturen som (säg) 62,7 och andra kommer att läsa den som 62,8. Det verkar
att oavsett vilket naturfenomen vi väljer och vilket protokoll vi använder så kommer det alltid att finnas "hörnfodral". För en kryptografisk beacon, även en liten sannolikhet för inkonsekventa mätningar kan vara oacceptabelt eftersom det kommer att göra att de slumpmässiga bitarna som matas ut av en PRG blir fullständigt
annorlunda.

Finansiella data . En liknande idé är att använda flöden av finansiella data såsom börskurser. Återigen, dessa är offentligt observerbara värden. Till skillnad från naturfenomen rapporteras de som digitala värden, så problemet med inkonsekventa observationer försvinner. Det finns en stark anledning att tro att det är väldigt svårt

att förutsäga lågnivåfluktuationerna i aktiekurserna: om du kunde förutsäga inom ett öre vad slutgiltigt priset på en specifik aktie kommer att vara på New York-börsen i morgon, kan du göra en hel del vinst som daytrader. Någon skulle kunna försöka påverka priset genom att köpa eller sälja aktien för att driva det till ett visst värde, men det har en verklig kostnad som du inte kan undvika.

Men detta tillvägagångssätt har också problemet med att förlita sig på en betrodd part, nämligen aktien utbyta. Även om börsen har ett starkt incitament att slå fast att det är ärligt och i god tro kan det fortfarande finnas misstankar om att de kan försöka ändra priset på en aktie med en slant (till exempel genom att lägga in sin egen order i orderboken) om det skulle låta dem rigga en värdefullt lotteri.

Med alla tillvägagångssätt vi har tittat på verkar det vara svårt att undvika att ha någon pålitlig part som har inflytande över någon avgörande del av processen.

Med hjälp av Bitcoin som Beacon . Lyckligtvis har ett tema hittills i hela boken varit att Bitcoin är en lovande teknik för att ta bort centraliserat förtroende från protokoll på sätt som vi tidigare inte trodde var möjliga. Kan vi använda Bitcoin som en slumpmässig ledstjärna? Vi skulle vilja extrahera slumpmässiga data från
Bitcoin blockkedja samtidigt som man behåller alla de decentraliserade egenskaper som gör Bitcoin själv så attraktiv.

Kom ihåg att gruvarbetare måste beräkna massor av slumpmässiga hash-värden medan de försöker hitta en vinnande block. Kanske betyder detta att ingen kan förutsäga eller påverka vad nästa blockhash kommer vara utan att faktiskt göra gruvarbetet. Naturligtvis kommer de första bitarna av blockhash att vara noll, men det visar sig att under lämpliga antaganden är det enda sättet att förutsäga de återstående bitarna skulle vara att påverka dem genom att hitta ett vinnande block och selektivt kassera det.

Figur 9,11: Extrahera offentliga slumpmässighet från hashar av block i blockkedjan.

Det gör det enkelt att förvandla blockkedjan till en slumpmässighet. För varje block i kedjan, vi tillämpar en "slumpmässig extraherare" på värdet på blockhuvudet. En slumpextraktor, ungefär talat, är som en hash-funktion som är utformad för att klämma in all den slumpmässiga entropin av inmatningen i den ena likformigt slumpmässiga strängen. Varje gång ett block publiceras har vi ny beacon-utgång.
255

Sida 57

Utvärdering av säkerheten i en Bitcoin ledstjärna. Låt oss säga att du deltar i ett lotteri vars utgång bestäms av utsignalen från Bitcoin-fyren för något förutbestämt framtida block på höjden H in blockkedjan. Det finns N spelare i detta lotteri, och var och en av dem satsar B bitcoins. Om du är också en gruvarbetare kan du ha tur och hitta en hashpussellösning för block H. Då har du valet huruvida blocket ska publiceras eller inte. Om du inte gillar lotteriresultatet som skulle bli resultatet av när du publicerar blocket du hittade kan du helt enkelt slänga det och låta lotteriet avgöras av vem som helst publicerar block B. Däremot skulle du förlora de intäkter du kan tjäna på det blockera.

Låt oss räkna ut hur stor insatsen B måste vara för att du ska hitta den selektiva kastningsstrategin värt. Du hittar framgångsrikt ett block på blockhöjd H och inser att om du publicerar det kommer du att göra det förlora definitivt lotteriet, medan om du kasserar blocket har du fortfarande en 1/N chans att vinna B * N bitcoins. Det betyder att det kommer att vara rationellt att kassera blockeringen om din förväntade utbetalning på 1/N * B * N
bitcoins (dvs B bitcoins) är större än belöningen för att bryta ett block (ungefär 25 bitcoins 2015, ignorerar transaktionsavgifter). Så attacken är lönsam om B > 25. I mitten av 2015 är 25 bitcoins värt över

5 000 amerikanska dollar. Så om insatsen per spelare är under $5 000, kommer lotteriet att vara säkert mot denna attack
om spelarna är rationella.

En av fördelarna med detta schema är att det är en helt decentraliserad ledstjärna, utan någon central punkt förtroende. Jämfört med vissa andra beacon-förslag går det ganska snabbt. Det kan skapa en utgång ungefär varje tio minuter. Det är också användbart att kunna uppskatta kostnaden för en angripare att manipulera fyren utgångar med vår enkla modell ovan.

En nackdel med att använda Bitcoin som ledstjärna är att dess timing är något oprecis. Säg att vi vill läsa värdet av fyren i morgon kl. Vi vet inte exakt vilket block som blir det senaste blocket vid den tiden. Även om ett block i genomsnitt kommer att publiceras inom 10 minuter före eller efter kl. det finns en viss variation. Vi måste också planera att tolerera lite mer förseningar om vi vill minska sannolikheten för att blocket vi tittar på försvinner i en kort gaffel. Som vanligt i Bitcoin skulle vi vilja vänta på ungefär sex kvarter kvar innan vi tror att beaconvärdet verkligen har satt sig.

En annan nackdel är att kostnaden för att manipulera beaconvärdet kan vara för låg för vissa applikationer vi bryr oss om. Om vi faktiskt körde NBA-draften, där det finns tiotals miljoner dollar på spel kan det plötsligt se lönt ut för ett av lagen att börja muta Bitcoin-gruvarbetare för att manipulera denna process. Det är fortfarande en öppen fråga om vi kan förlänga detta konstruktion för att göra det säkert när miljontals dollar står på spel.

Slutligen ignorerar vår säkerhetsutvärdering några verkliga faktorer. Till exempel en gruvarbetare som ingår i en gruvpool förlorar inte mycket på att kassera ett block, eftersom de belönas på basis av aktier snarare än block. För nu är Bitcoin beacons en intressant men oprövad idé.

Skriptstöd för fyrar. Vad händer om vi utökat Bitcoin s skriptspråk med en speciell opkod att läsa beacon-värden? För närvarande finns det inget sätt att ha någon slumpmässighet i Bitcoin-skript. Det är förbi

---

design, eftersom gruvarbetare måste verifiera skript och de vill alla komma överens om huruvida ett skript är giltigt eller
inte. Men om vi använder beaconvärdet är det en offentlig källa till verifierbar slumpmässighet. Vi skulle kunna använda
beacon för att lägga till slumpmässighet i transaktionsskript som alla gruvarbetare kan komma överens om.

Anta att vi hade en opkod som skulle fatta ett slumpmässigt beslut baserat på beacon-utgången från föregående block. Vi skulle kunna ersätta hela det komplicerade lotteriprotokollet med bara ett skript som läser pejlvärde och tilldelar den utgång till en av **n** nycklar. Det skulle inte kräva en multi-round protokoll, säkerhetsinsättningar eller tidsinställda hashåtaganden.

En nackdel med denna idé är att det nu skulle vara möjligt för gruvarbetare att helt enkelt manipulera lotteriet genom att skjuta upp lotteritransaktionen till en senare blockering om de finner att inklusive transaktionen i block som de bryter skulle få dem att förlora lotteriet. Det kräver inte längre förverka block belöningar. Det är möjligt att göra en variant av beacon-opkoden som undviker denna attack. Istället för Med hänvisning till föregående block anger du att beaconvärdet ska användas vid en viss blockhöjd.

9.5: Prediction Markets och Real World Data Feeds

För det sista ämnet i detta kapitel kommer vi att titta på hur man implementerar en förutsägelsemarknad i en decentraliserat sätt att använda kryptovalutor och det relaterade ämnet för att föra in verklig data i Bitcoin.

En förutsägelsemarknad låter människor träffas och satsa på framtida evenemang som sport
spel eller val. Deltagare på en förutspådd marknad köper, säljer och handlar specifikt med "aktier".
resultatet av sådana händelser.

Team Tyskland
Argentina
Brasilien
Förenta staterna
England
Nederländerna
Före-turnering
0,12
0,09
0,22
0,01
0,05
0,03
Efter gruppspelet
0,18
0,15
0,31
0,06
0,00
0,05
Inför semifinalerna
0,26
0,21
0,45
0,00
0,00
0,08
Inför finaler
0,64
0,36
0,00
0,00
0,00
0,00
Slutlig
1
0
0
0
0
0

Tabell 9.12: Priser i dollar på en hypotetisk förutsägelsemarknad för ett urval av lag under VM 2014. Priset på en aktie som satsade på att det amerikanska laget skulle vinna cupen steg från 1 cent till 6 cent efter att USA presterade bra i gruppspelet. En aktie i Brasilien steg successivt till 45 cent som Brasilien gick vidare till semifinal och tappade sedan hela sitt värde efter att Brasilien förlorat sin semifinalmatch. Efter turneringen hade bara aktier i Tyskland (som vann turneringen) något värde.

Låt oss gå igenom ett exempel som borde göra koncepten bakom förutsägelsemarknader mer tydliga. VM 2014 hölls i Brasilien. Anta att det fanns en marknad där man kunde köpa och sälja aktier associerade med varje lag, och andelarna för laget som vinner kommer i slutändan att vara värda 1 dollar och alla andra aktier är värda 0. När man går in i turneringen skulle alla lag börja med ett pris som inte är noll, baserat på vad marknaden tror att deras chanser att vinna är. Exempel visas i tabell 9.12 för fem olika lag.

I fasen före turneringen handlas tyska aktier för cirka 12 cent, vilket innebär att marknaden tror ungefär att Tyskland har 12 % chans att vinna. Allt eftersom turneringen fortskrider kommer dessa priserna kommer att fluktuera, vilket återspeglar hur marknadsaktörerna justerar sina övertygelser om hur sannolikt var och en laget ska vinna.

I vårt exempel handlades England initialt för fem cent men gick till noll efter gruppspelet. Det beror på att England slogs ut i gruppspelet. Det finns inte längre något sätt för dem att göra det vinna, och priset återspeglar det; deras aktier är nu värdelösa. Å andra sidan, det amerikanska laget som ansågs initialt ha en mycket dålig chans att överleva gruppspelet visade sig göra mycket väl. Om du hade tänkt köpa amerikanska aktier i början när de var väldigt billiga (en cent), du kunde sälja dem direkt efter gruppspelet för sex cent. Du skulle få tillbaka sex gånger så mycket pengar du satsar. Du skulle inte behöva vänta till efter turneringens slut för att göra vinst. Även om det amerikanska laget inte vann turneringen, skulle du kunna dra nytta av det faktum att du förutsåg en förändring i övertygelsen om deras chanser att vinna efter deras starka uppträdande i gruppspelet.

När vi kommer till semifinal är det bara fyra lag kvar. USA och England slogs ut så deras aktiekurser har redan gått till noll. Nu har varje kvarvarande lag ett relativt högt pris, och deras aktiekurser bör läggas till 1,0. Särskilt Brasilien var favoriserat att vinna, och hade därmed högsta priset. Faktum är att Brasilien förlorade i semifinalen och deras aktiekurs gick till noll. Inom spannet av ett par timmar förändrades marknadens övertygelser dramatiskt. Du skulle ha kunnat tjäna på en mycket kort tidsram om du var säker på att Brasilien var överskattat när du gick in i matchen; du kunde ta en "kort position" på Brasilien och/eller satsa på de andra lagen.

När vi går in i finalen är det bara två lag kvar och deras andelar blir återigen 1,0. Vid själva i slutet av turneringen, naturligtvis, de enda aktierna som till slut har något värde är tyskens laget sedan de vann.

Uppenbarligen skulle ett sätt du kunde ha gjort en vinst ha varit att köpa aktier i Tyskland på börjar för 12 cent och håll dem hela vägen till slutet. Detta är i princip hur traditionella sporter

vadslagning fungerar — du lägger ett spel innan turneringen startar och samlar in utbetalningen efter slutet av turneringen. Men på en förutsägelsemarknad finns det många andra sätt att spela och tjäna pengar.
Du kan investera i vilket lag som helst när som helst, och du kan tjäna enbart på förmågan att förutse att människors
övertygelser kommer att förändras, oavsett det slutliga resultatet.

Här är ett annat exempel, den här gången från en verklig förutsägelsemarknad. Före presidentvalet i USA 2008 valet tillät Iowa Electronic Markets människor att köpa aktier för om Barack Obama eller eller John McCain skulle vinna. I figur 9.13 visas priset på Barack Obama-aktier i blått och McCain visas i rött. Du kan se att när kampanjmånaderna utvecklades, folks övertygelser om vem som skulle vinna fluktuerade. Men dagen före valet fick Obama 90 % chans att vinna. Marknaden var väl medveten om att resultatet i princip var avgjort innan röster avgavs.

Figur 9.13 : Priset på förutspådda marknadsandelar om det amerikanska presidentvalet 2008, från Iowa Elektroniska marknader.

Sidebar: The Power of Prediction Markets. Ekonomer tenderar att vara entusiastiska över förutsägelser marknader. Information som är relevant för att förutse framtida händelser är ofta spridd, och förutsägelsemarknader är en utmärkt mekanism för att aggregera den informationen genom att ge deltagarna ett sätt att dra nytta av sin kunskap. Under lämpliga ekonomiska modeller, marknadspriset på aktier kan tolkas som sannolikheten för resultatet, även om det finns farhågor som är verkliga förutsägelsemarknader lider av fördomar. Empiriskt har förutsägelsemarknaderna hållit sig mycket väl mot andra prognosmetoder såsom opinionsundersökningar och expertpaneler.

Men förutsägelsemarknaderna står inför många osäkerheter och hinder i lagstiftningen. Intrade var mest populär förutsägelsemarknad på internet innan den stötte på problem med efterlevnad av regler i USA och lade ner 2013. Många ekonomer var besvikna över detta eftersom de kände att vi förlorade en värdefullt socialt verktyg som avslöjade användbar information om framtiden.

Decentraliserade förutsägelse marknader . Vad skulle det ta att bygga en *decentraliserad* förutsägelse marknaden?
Det finns några uppgifter som vi måste decentralisera. Vi behöver ett sätt att ta emot pengar och utbetalande utbetalningar, och vi behöver ett sätt att genomdriva att rätt belopp betalas ut enl till resultatet. Vi kommer särskilt att behöva decentraliserad skiljedom. Skiljedom är processen att hävda

---

vilka resultat som faktiskt hände. Oftast när det gäller ett nationellt val eller en sport match är det ganska uppenbart vem som vann och vem som förlorade. Men det finns också många gråzoner. Vi behöver också
att decentralisera orderboken, vilket är ett sätt för människor att hitta motparter att handla aktier med.
Vi kommer att gå igenom var och en av dessa utmaningar i ordning.

Låt oss designa ett hypotetiskt altcoin som heter "Futurecoin" som har explicit stöd för förutsägelsemarknader. Vi skulle behöva några nya transaktionstyper som utför funktioner som är specifika för förutsägande marknader. Det kan
se ut ungefär som figur 9.14.

CreateMarket tillåter alla användare att skapa en förutsägelsemarknad för alla händelser genom att ange en skiljeman
(i termer av en offentlig nyckel) vem som är behörig att deklarera resultatet av den händelsen, och antalet möjliga resultat. Event_id är en godtycklig sträng som binder samman de olika transaktionerna som hänvisa till samma marknad. Futurecoin bryr sig inte om vilken verklig händelse event_id refererar till, inte heller vad resultaten är, och det finns inget sätt att specificera dessa inom systemet. Användare måste få denna information från marknadsskaparen (som vanligtvis är densamma som skiljemannen). Vi kommer att diskutera olika alternativ för skiljeförfarande inom kort.

Betalning och avveckling . BuyPortfolio kan du köpa en portfölj av aktier i någon händelse. För Priset på en futurecoin, kan du köpa en aktie i *varje* möjligt resultat av händelsen. Anta att vi är det satsar på VM 2014. Det finns 32 lag som kan vinna. För ett mynt kan du köpa 32 aktier, en för varje lag — detta är helt klart "värt" exakt ett mynt eftersom exakt ett av lagen kommer att göra det till slut vinna. Alla användare kan ensidigt skapa en BuyPortfolio utan att behöva en motpart. De transaktionen förstör i huvudsak ett framtidsmynt som tillhandahålls av användaren och skapar ett nytt ta del av varje resultat. Det finns också en transaktionstyp för att sälja en portfölj, som låter dig sälja (eller bränna) en andel i varje resultat för att få tillbaka ett framtidsmynt. För ett futurecoin kan du köpa en andel i varje utfall och sedan kan du förvandla en andel i varje utfall tillbaka till ett framtidsmynt.

CreateMarket(event_id, arbitrator_key, num_outcomes)
skapa en ny förutsägelsemarknad, med angivande av skiljedomare och parametrar
BuyPortfolio(event_id)
köp en aktie i varje utfall för 1 futurecoin
TradeShares(...)
överföra aktier i utbyte mot framtida mynt
SellPortfolio(event_id)
lös in en andel i varje utfall för 1 futurecoin
CloseMarket(event_id, outcome_id)
stänga marknaden för den angivna händelsen genom att konvertera alla aktier av det angivna resultatet till 1 nypräglat futurecoin och förstör alla andelar av alla andra utfall i evenemanget
(outcome_id är ett heltal mellan 1 och num_outcomes för händelsen)

Figur 9.14: Nya transaktionstyper i Futurecoin, ett hypotetiskt altcoin som implementerar en decentraliserad förutsägelsemarknad

---

Du kan också handla aktier mot futurecoins, eller en sorts andel mot en annan typ av aktie, så länge som du kan hitta någon att handla med. Det här fallet är mycket mer intressant. Du kan spendera ett framtidsmynt att köpa en andel i varje utfall och sedan sälja av aktierna i utfall du inte tror är sannolikt inträffa. För de lag du inte vill satsa på kan du sälja dessa aktier till någon annan som gör det vill satsa på det laget. När du gör detta har du inte längre en balanserad portfölj i varje lag,

och du kan inte längre automatiskt lösa in din portfölj mot ett futurecoin. Istället måste du vänta tills vadet slutar för att lösa in dina andelar — och om laget/lagen du satsat på inte vann, kanske inte kan lösa in dem mot något alls. Å andra sidan kan du också tjäna direkt genom handel. Du kan köpa en balanserad portfölj, vänta på att priserna ändras och sedan sälja alla aktier direkt för framtida mynt, som du sedan kan byta mot Bitcoin eller valfri annan valuta.

Prediction marknads skiljedom . Hur kan vi göra skiljeförfarande på ett decentraliserat sätt? Hur kan vi göra påståenden om vem som faktiskt vann så att vi kan låta folk lösa in sina vinnande andelar i slutet? De enklaste systemet är att ha en betrodd skiljedomare, vilket är vad CreateMarket ovan gör. Alla användare kan lansera en marknad där de är skiljedomare (eller utse någon annan som skiljeman). Dom kan skapa en transaktion och meddela att de öppnar en marknad för VM-resultaten. De kommer att avgöra vem som vann till slut, och om du litar på dem bör du vara villig att acceptera deras signatur på en CloseMarket-transaktion som bevis på resultatet.

Liksom på många andra marknader föreställer vi oss att vissa enheter med tiden kommer att bygga upp ett rykte som pålitliga
skiljemän. Då skulle de ha ett visst incitament att medla korrekt för att behålla sina värdefulla rykten. Men det finns alltid risken att de kan stjäla mycket pengar - mer än deras rykte är värt - genom att rigga ett vad. Detta skulle vara mycket farligt på en förutsägelsemarknad. För till exempel på VM-marknaden kunde skiljedomaren hävda att Argentina vann, trots att de faktiskt förlorat. Om skiljedomaren hade satsat hårt på Argentina själva, då skulle de kanske kunna tjäna tillräckligt på det för att rättfärdiga att förstöra deras rykte.

Skulle vi kunna ha ett mer decentraliserat skiljedomssystem? Ett alternativ är att ange flera skiljemän, varvid utgången avgörs utifrån majoriteten. Det finns också idéer baserade på röstning — antingen av alla användare som har aktier på marknaden eller av gruvarbetare i kryptovalutan. Förslag i den stilen föreslår ofta att väljarna ska straffas för att de röstar mot majoriteten. Men där
Det finns många potentiella problem med dessa tillvägagångssätt och vi vet helt enkelt inte hur väl de kommer att fungera
arbete i praktiken.

En ytterligare rynka är att verkligheten ibland är komplicerad. Förutom problemet med skiljemän ljuger, kan det finnas en legitim tvist om resultatet av händelsen. Vårt favoritexempel är från Super Bowl 2014. Det finns en tradition vid Super Bowl att det vinnande laget dumpar en hink med Gatorade på deras huvudtränare. Folk gillar att satsa på färgen på Gatorade som det vinnande laget använder för detta firande, och denna vadslagning har hänt i två eller tre decennier. 2014 fanns det satsningar på gul, orange och alla andra färger i Gatorade. Men det året, ett aldrig tidigare skådat resultatet gjorde det svårt att avgöra vadet. När Seahawks vann dumpade de orange Gatorade deras huvudtränare, Pete Carroll. Sen lite senare bestämde sig några andra spelare för att göra det igen och

---

dumpa *en annan* hink med Gatorade på honom. Den första hinken innehöll orange Gatorade och andra hinken innehöll gul Gatorade.

Om du drev en förutsägelsemarknad där folk hade satsat på färgen på Gatorade, hur skulle du hantera detta scenario? Det är inte klart om orange, gul eller båda ska vinna. Vad hände i praxis med flera sportspeltjänster är att de bestämde sig för att det var bättre att förlora lite pengar på för att behålla sitt rykte. Som ett bevis på god tro till sina kunder betalade de ut

vinster för alla som satsa på *antingen* orange eller gult.

Naturligtvis, på en decentraliserad förutsägelsemarknad är detta inte så lätt, eftersom du inte bara kan skapa pengar
ur tomma luften för att betala båda uppsättningarna av parterna. Istället kunde skiljemannen dela vinsterna lika bland både orange och gult. Istället för att stänga till ett värde av 1,0 skulle båda aktierna stänga till ett värde av 0,5. Du kan definiera avtalet noggrant för att undvika denna förvirring, men du kan inte vara säker på att du har gjort det
förutsåg alla möjligheter. Lärdomen här är att skiljeförfarande delvis är ett socialt problem och nej teknisk lösning kommer att bli perfekt.

Dataflöden . Idén med skiljedom leder till ett mer allmänt koncept: att utöka kryptovalutor med en mekanism för att hävda fakta om den verkliga världen. Vi kallar en sådan mekanism för ett dataflöde. Ett faktum kanske
handla om typiska prediktionsmarknadshändelser som vem som vann ett val, eller priset på en aktie eller vara en viss dag, eller andra viktiga data från den verkliga världen. Om vi hade sådana fakta tillgängliga i Bitcoin skulle skriptspråket kunna använda dem som indata. Till exempel kan ett skript vara kunna ladda det aktuella kopparpriset på stacken och fatta beslut baserat på värdet.

Om tillförlitliga dataflöden fanns kunde vi lägga – och automatiskt avgöra – vad på sportmatcher eller det framtida priset på råvaror. En förutsägelsemarknad är bara en applikation som detta skulle möjliggöra. Du kan säkra riskerna i din investeringsportfölj genom att satsa mot priset på aktier du äger. Och du kan härleda en mängd olika finansiella instrument som terminer och terminer som är vanliga handlas på finansmarknaderna. Skulle det inte vara bra om vi kunde göra allt detta inom Bitcoin?

Vi kan skilja den tekniska frågan om hur man *representera* verkliga fakta i Bitcoin (eller altcoin) från den sociotekniska frågan om hur vi kan förbättra vårt förtroende för fodrets korrekthet. Vi har redan tittat på den förra frågan när vi diskuterade alternativ för skiljeförfarande.

Ett smart sätt att koda dataflöden till vanliga Bitcoin kallas Reality Keys. I detta system är skiljemannen skapar ett par signeringsnycklar för varje resultat av varje evenemang de är intresserade av - en nyckelpar för "Ja", och ett nyckelpar för "Nej". De publicerar de offentliga nycklarna när händelsen är först registreras och senare publicera exakt en av de två *privata* nycklar när resultatet regleras. Om Alice satsade mot Bob på att resultatet skulle inträffa, kunde de skicka sina insatser till en Bitcoin utdata som antingen kan göras anspråk på av Alice med en signatur från Alice och från "Ja"-tangenten, eller gjort anspråk på av Bob med en signatur från Bob och från "Nej"-tangenten. Detta är långt ifrån det ideala mål att kunna använda dataflödesvärden som skriptindata på godtyckliga sätt, men det tillåter enkla applikationer som satsningen som beskrivs ovan. Observera att skiljemannen inte behöver veta om eller engagera dig i den specifika satsningen mellan Alice och Bob.

Orderböcker . Den sista delen av en förutsägelse marknaden är ett decentraliserat orderbok. Återigen är detta en ganska allmänt koncept, och att inse det skulle tillåta många andra tillämpningar. Vad är en orderbok?
På verkliga förutsägelsemarknader, eller de flesta finansiella marknader, finns det inte ett enda marknadspris. Istället där
är *bud* och *frågar* som anges i *orderboken* . Ett bud är det högsta priset som någon är villig att köpa en aktie för och *frågar* är den lägsta pris som någon är villig att sälja aktien för. Typiskt

ask är större än budet (annars skulle det finnas två deltagare som skulle matchas, a
handel skulle inträffa, och åtminstone en av orderna skulle inte längre finnas kvar i orderboken). A
deltagare som vill köpa en aktie direkt kan göra det till säljkurs och en deltagare som
vill sälja direkt kan göra det till budpriset. Dessa kallas "marknadsorder" eftersom de
exekvera till marknadspris, till skillnad från de "limitordrar" som finns registrerade i orderboken som
exekvera till det angivna gränspriset (eller bättre).

Traditionellt har detta gjorts på ett centraliserat sätt med en enda orderboktjänst (vanligtvis en
utbyte) som samlar in alla beställningar. Problemet, som är typiskt för centraliserade tjänster, är att en
oärligt utbyte kan tjäna på deltagarnas bekostnad. Om börsen får en marknad
köporder, kanske de själva köper från bästa fråga innan de lägger beställningen de fått, då
vänd om och sälj aktierna som de just köpt till ett högre pris, vilket gör att skillnaden stoppas. Detta
övning kallas frontrunning. Det dyker upp i en mängd olika finansiella miljöer, och det anses vara ett
brottslighet. Centraliserade orderböcker kräver laglig verkställighet för att motverka frontrunning och
säkerställa förtroendet för systemets integritet.

I en decentraliserad orderbok kan vi inte förlita oss på en stark rättstillämpning. Men det finns en smart lösning,
vilket är att helt enkelt glömma frontrunning. Istället för att förklara det som ett brott och försvara sig mot det,
vi kallar det en funktion. Tanken är att vem som helst kan skicka limiterade order till gruvarbetare genom att
sända
transaktioner, och gruvarbetare kan matcha några två order så länge som bud är *större* än eller lika med
fråga. Gruvarbetaren får helt enkelt behålla mellanskillnaden som en form av transaktionsavgift. Nu har
gruvarbetare nej
incitament till frontrun eftersom frontrunning en order aldrig kommer att vara mer lönsamt än helt enkelt
uppfylla det och fånga överskottet.

Detta är ett elegant sätt att bygga en decentraliserad orderbok. Den största nackdelen är gruvarbetaravgifterna som
handlare måste betala. För att undvika att betala den avgiften kan folk lämna mycket mer konservativa order och
kanske inte är villiga att i förväg avslöja det bästa priset som de är villiga att handla till. Detta kan göra
marknaden mindre effektiv. Vi vet ännu inte hur den här typen av orderbok med gruvarbetare matchar order
kommer att fungera i praktiken, men det verkar vara en lovande idé.

Sammanfattningsvis kan Bitcoin som det är idag fungera som en plattform för en mängd olika applikationer. Men
för vissa
applikationer Bitcoin tar oss bara så långt. Den har inte alla funktioner vi behöver för en säker
decentraliserad förutsägelsemarknad eller en decentraliserad orderbok.

Sida 65

Men tänk om vi kunde börja om från början och glömma mjuka gafflar, hårda gafflar och andra utmaningar i
fästa nya funktioner på Bitcoin? Vi har lärt oss mycket sedan 2008 när Bitcoin först kom ut. Varför
inte designa en ny kryptovaluta från grunden och göra allt bättre?

I nästa kapitel ska vi titta på altcoins, som är försök att göra just det. Vi ska prata om alla
lovande idéer och utmaningarna för att starta en ny kryptovaluta.

Vidare läsning

Projektsidor / specifikationer för två av överlagringsprotokollen vi tittade på:

Motpartsprotokollspecifikationen

OpenAssets-protokollet

Det säkra flerpartslotteriprotokollet vi beskrev är från följande papper, som inte är för svagt hjärta!

Andrychowicz, Marcin, Stefan Dziembowski, Daniel Malinowski och Lukasz Mazurek. Säkert fler beräkningar på Bitcoin . I IEEE Security and Privacy 2014.

Dokument från ekonomer om kraften hos förutsägelsemarknader:

Wolfers, Justin och Eric Zitzewitz. Prediction marknader . nr. w10504. National Bureau of Economic Forskning, 2004.

Arrow, Kenneth J. et al. Löftet förutsägelse marknader . 2008.

Förutsägelsemarknadsdesignen vi beskrev är från denna artikel, medförfattad av flera av de närvarande författare:

Clark, Jeremy, Joseph Bonneau, Edward W. Felten, Joshua A. Kroll, Andrew Miller och Arvind Narayanan. På decentralisera Prediction Markets och orderböckerna . I workshop om ekonomi för informationssäkerhet, 2014.

Kapitel 10: Altcoins and the Cryptocurrency Ecosystem

Bitcoin är bara en komponent (om än en viktig sådan) i ett bredare ekosystem av alternativ, men ofta ganska lika, valutor kallas *altcoins* . I det här kapitlet ska vi titta på altcoins och ekosystemet av kryptovalutor.

10.1 Altcoins: Historia och motivation

Bitcoin lanserades i januari 2009. Det var inte förrän i två år till, förrän i mitten av 2011, som det första Bitcoin-liknande härledda systemet, Namecoin, lanserades. Hastigheten för lanseringar av altcoin exploderade
2013, och hundratals har sedan dess följt. Hur många är det totalt? Det är omöjligt att tillhandahålla en exakt antal eftersom det inte är klart vilka altcoins som är värda att räkna. Till exempel om någon tillkännager ett altcoin och kanske släpper lite källkod, men ingen har börjat bryta eller använda den ändå, räknas det? Andra altcoins har lanserats och använts en del, men dog sedan mycket

snabbt efter lanseringen.

Figur 10.1: Altcoins som lanseras per månad (mätt med genereringsblock).

Det är inte heller helt klart vad som är en altcoin i motsats till helt enkelt en annan kryptografisk valuta. där var, trots allt, olika kryptovaluta förslag och system som föregår Bitcoin och dessa är brukar inte kallas altcoins. Många altcoins lånar koncept från Bitcoin, ofta direkt gafflar koden basera eller på annat sätt anta en del av Bitcoins kod. Vissa gör endast mycket små ändringar Bitcoin, såsom att ändra värdet på vissa parametrar i systemet, och fortsätta att införliva ändringar gjorda av Bitcoins utvecklare. Hittills börjar alla altcoins som vi känner till med en ny genesis block och deras egen alternativa syn på transaktionshistorik, snarare än att dela Bitcoins blockkedja efter en viss tidpunkt i historien. För våra ändamål behöver vi inte en exakt definition av ett altcoin. Istället kommer vi löst att referera till någon kryptovaluta som lanserats sedan Bitcoin som en altcoin.

Vi kommer kort att nämna icke-altcoin-system som Ripple och Stellar: dessa är distribuerad konsensus protokoll i den tradition som vi tittade på i kapitel 2. Dessa system uppnår konsensus i en modell där noder har identifierare och måste vara medvetna om varandra. Bitcoin avviker naturligtvis radikalt från denna modell. I både Ripple och Stellar stöder konsensusprotokollet en betalning/uppgörelse nätverk, och varje system har en inhemsk valuta. Trots dessa likheter med altcoins gör vi det inte anser att de ligger inom ramen för denna bok.

Anledningar till att lansera altcoins. Varje altcoin behöver någon sorts historia att berätta. Om en altcoin inte kan göra anspråk
någon egenskap som skiljer den från alla andra, det finns ingen anledning att existera. I den enklaste fall, en altcoin ändrar helt enkelt några av de inbyggda parametrarna till Bitcoin. Detta inkluderar saker som den genomsnittliga tiden mellan blocken, blockstorleksgränsen, schemat för belöningar skapat, eller inflationstakten för altcoin.

Det kan också finnas mer komplexa tekniska skillnader, vilket är ett mer intressant fall. Till exempel det kan finnas tillägg till skriptspråket för att uttrycka olika typer av transaktioner eller säkerhet egenskaper. Gruvdrift kan fungera annorlunda och konsensusalgoritmen kan vara betydligt annorlunda från Bitcoins.

Ibland lanseras även altcoins med ett tema eller en känsla av en gemenskap som altcoinn är avsedda att stödja eller förknippas med, ofta ge medlemmar i denna gemenskap en speciell roll eller förmågor i altcoin. Vi kommer att titta på exempel på alla dessa senare i det här avsnittet.

Hur man lanserar ett altcoin. Låt oss överväga vad som är involverat i processen att lansera ett altcoin och vad som händer efter att ett altcoin har lanserats. Som vi nämnde innebär att skapa en altcoin att skapa en ny referensklient, typiskt genom att splittra den befintliga kodbasen för vissa befintliga, mer väletablerad altcoin, eller av själva Bitcoin. Den enkla delen är att lägga till ett gäng tekniska funktioner eller modifierade parametrar som du tror kommer att fungera bra. Faktum är att det en gång fanns en webbplats som hette Coingen
som skulle automatisera denna process för en liten avgift. Det tillät dig att specificera olika parametrar som genomsnittlig blockeringstid och algoritmen för proof-of-work du ville ha, förutom ett namn på din

altcoin, en valutakod på tre bokstäver och en logotyp. Sedan genom att klicka på en knapp laddar du ner en gaffel Bitcoin med de parametrar du valde, och du (och andra) kunde omedelbart börja köra den.

Det svåra är att bootstrapping antagandet av din altcoin. Du kan punga källkoden och du kan tillkännage det offentligt, men vid det här laget använder ingen din altcoin så den har inget marknadsvärde (sedan ingen vill ha mynten) och ingen säkerhet (eftersom det inte finns gruvarbetare än). I kapitel 7 såg vi det Det finns ett antal intressenter i Bitcoin: utvecklare, gruvarbetare, investerare, handlare, kunder och betaltjänster. Så småningom måste du locka alla dessa typer av deltagare till din altcoin ekonomi för att få igång det.

Alla dessa är viktiga och relaterade till varandra, och analoga med utmaningen med att lansera någon annan plattform och få den antagen. Om du vill lansera ett nytt smartphone-operativsystem, säg, du skulle behöva locka användare, enhetstillverkare, apputvecklare och olika andra intressenter, och var och en av dessa grupper behöver de andra.
266

Sida 68

Att locka gruvarbetare har särskild betydelse för kryptovalutor eftersom de saknar tillräcklig hashkraft bakom en altcoin kan säkerheten misslyckas om dubbelt utgifter och gafflar är möjliga. Faktum är att din altcoin kan köras över helt; vi ska titta på "altcoin barnmord" senare i det här kapitlet. Det finns inte en enkelt recept för bootstrapping adoption, men i allmänhet kommer gruvarbetare när de tror på coinbase-belöningar de får kommer att vara värda ansträngningen. För att uppmuntra detta ger många altcoins tidigt
gruvarbetare större belöningar. Bitcoin, naturligtvis, banade väg för detta tillvägagångssätt, men vissa altcoins har tagit en
mer aggressivt tillvägagångssätt för att belöna tidiga gruvarbetare.

Att få en grupp människor att tro att altcoin är värdefullt är det svåraste tricket. Som vi diskuteras i kapitel 7, även för Bitcoin är det inte klart exakt hur denna process var bootstrappad som den förlitar sig på Tinkerbell-effekten. Detta knyter an till varför altcoins behöver en bra berättelse: att ta sig av markgemenskap av människor måste tro att den nya altcoin verkligen kommer att vara värdefull i framtid (och tror att andra kommer att tro att det är värdefullt, och så vidare).

Med tanke på en gemenskap av människor som är intresserade av att skaffa ett altcoin, kommer gruvarbetare vanligtvis att komma (även om det
kan vara riskabelt om värdet ökar snabbare än gruvarbetare kan byta till att börja bryta valuta). Andra viktiga element kommer vanligtvis att följa i sin tur när värdet uppfattas som att få ditt altcoin noterat på börser och utveckla olika typer av stödjande infrastruktur är användbara, allt från en opinionsbildningsstiftelse till verktyg för att utforska blockkedjan

Pump-and-dump-bedrägerier. När skaparna av ett altcoin har lyckats starta upp en community och en riktig växlingsmarknad har de ofta funnit sig mycket rika. Det är för att de nästan säkert äger en stor mängd mynt - till exempel genom att vara tidiga gruvarbetare innan hashen hastighetsökningar, eller till och med "pre-mining", som vi diskuterar nedan. När altcoins växelkurs stiger, grundarna kommer att kunna sälja sina mynt om de väljer det.

Möjligheten att bli rik har lockat entreprenöriella individer och riskkapital till altcoins, och föga överraskande har det också lockat bedragare. Visserligen är gränsen mellan de två ibland en

lite suddigt. En bedragare kan använda en mängd olika metoder för att överdriva en altcoins potential och trumma öka intresset. De kan hajpa upp dess förmodade tekniska fördelar, fejka utseendet på gräsrötter stöd, köp altcoin på marknaden till höga priser och så vidare.

Faktum är att den här bluffen kan lösas även av någon som inte är grundaren av en altcoin. De skulle måste först köpa upp aktier av något obskyrt altcoin, sedan övertyga allmänheten om detta mynts förmodade oupptäckt potential (dvs. "pumpa" altcoint). Om de lyckas blåsa upp priset på det här sättet, gör de kan lossa sina aktier och skörda en vinst (dvs. "dumpa" sina mynt). Vid denna tidpunkt kommer investerarna förmodligen bli klok på bedrägeriet och priset kommer att rasa, med många människor kvar värdelösa mynt. Den här typen av pump-and-dump-bedrägerier har länge begåtts inom vanliga finanser, använde obskyra, lågprisaktier, och det var vanligt i de första dagarna av altcoins eftersom entusiasm var hög och investerare kämpade för att skilja verkligt innovativa altcoins från "me-too"-system med smart marknadsföring men ingen riktig innovation. Som ett resultat är användare och investerare något trötta på altcoins i dag.
267

Sida 69

Initial tilldelning. I Bitcoin allokeras valuta till användare enbart genom gruvdrift. Men för olika skäl, altcoin-utvecklare har sökt andra sätt för initial valutaallokering utöver gruvdrift.

Utvecklare kan "pre-mine" valutan, det vill säga reservera en del av penningmängden för sig själva eller någon annan utsedd enhet (såsom en ideell stiftelse med en stadga till utveckla valutan). Tanken är att möjligheten till ett oväntat fall ger utvecklare mer av en incitament att lägga tid på att skapa och starta upp en ny kryptovaluta. Ibland går de längre och gör en "förförsäljning", där de säljer dessa pre-mined enheter till andra spekulanter för bitcoins eller fiat valuta. Detta är lite analogt med att investera i en startup: spekulanterna kan bli rika om altcoin gör det stort.

Ett annat motiv för att söka ytterligare metoder för initial tilldelning är att säkerställa att det finns en mångfaldigt gemenskap av tidiga användare som äger valutan och har en del i dess framgång, givet det Idag är gruvdrift ganska centraliserad och kan leda till ett koncentrerat ägande av tillgångar. Ett smart sätt att möjliggöra mångsidigt ägande är att allokera altcoin-enheter till befintliga Bitcoin-ägare.

Hur kan vi tekniskt utforma systemet så att alla som äger bitcoins kan göra anspråk på sin del av altcoin, med detta krav som automatiskt döms? Ett alternativ är ett proof-of-burn, som vi diskuteras i kapitel 3: användare kan göra anspråk på enheter av ett nytt altcoin i proportion till en mängd bitcoins de förstör bevisligen. Ägaren kommer att förbinda sig till vissa uppgifter i beviset på bränningen, till exempel en speciell
sträng som identifierar det specifika altcoin, för att visa att de bränner bitcoins enbart för att tjäna nya enheter av denna specifika altcoin.

Att tilldela altcoins via ett proof-of-burn kallas också en "envägspeg" eller "pristak". Associerar en altcoin enhet (exempelvis) en Bitcoin faktiskt inte gör det *värt* en Bitcoin. Det säkerställer istället att altcoin kommer att vara värt *högst* en Bitcoin, eftersom en Bitcoin alltid kan lösas in för en altcoin, men inte tvärtom.

Figur 10.2: Tilldelning av altcoins via proof-of-burn. Altcoin stöder en GenCoin-transaktion som

tar en *Bitcoin* transaktion som indata. GenCoin är signerat av samma privata nyckel som signerade proof-of-burn (och använder samma signaturschema). Detta säkerställer att samma användare som brände bitcoins skapade också GenCoin. Om tappen förhållandet är 1: 1, sedan v' får inte vara större än v .

Det finns ett mindre tungt alternativ: kräva bevis på ägande av bitcoins, men inte bränna dem, att göra anspråk på altcoins. Specifikt skulle altcoin beteckna en Bitcoin-blockhöjd (kanske sammanfallande

---

med lanseringsdatumet för altcoin) under vilken alla som ägde en oanvänd Bitcoin-transaktion output från det blocket skulle kunna göra anspråk på en proportionell mängd altcoins. I det här systemet, där är inte nödvändigtvis ett fast förhållande mellan priset på en bitcoin och priset på en altcoin, eftersom bitcoins "konverteras" inte till altcoins via proof-of-burn.

Figur 10.3: Allokering av altcoins genom att bevisa ägande av bitcoins. Indata till GenCoin är en eller mer *oanvända Bitcoin transaktions utgångar* vid den utsedda blockhöjden. Det är undertecknat av den privata nycklar som styr dessa outnyttjade utdata, som i alla vanliga Bitcoin-transaktioner. Här är Bitcoin Den visade transaktionen har två outnyttjade transaktionsutgångar, till adresserna B och C, vid det angivna blocket höjd. Ägaren till adress B har gjort anspråk på sina altcoins, men ägaren till adress C har ännu inte gjort det så. Om tappen förhållandet är 1: 1, sedan v' får inte vara större än v_1
.

Naturligtvis, för att få allt detta att hända, måste altcoin-gruvarbetare hålla sig på toppen av Bitcoin-blockkedjan som
väl. Altcoin måste ange vad som räknas som en bekräftad Bitcoin-transaktion. Ett alternativ är att kräva något fast antal bekräftelser, säg 6. Ett annat alternativ är att ange det senaste Bitcoin-blocket i varje altcoin-block. På så sätt blir Bitcoin-transaktioner omedelbart tillgängliga att spendera i altcoin. Detta är analogt med det faktum att inom själva Bitcoin kan transaktionsutgångar spenderas i nästa block eller till och med i samma block. Merge mining, som vi kommer att diskutera i nästa avsnitt, är ett sätt att knyta altcoin-block till Bitcoin-block.

Slutligen, att donera redan tilldelade mynt är ett annat sätt att öka valutans mångfald ägare. En metod är dricks: olika tjänster gör det möjligt att skicka tips till en e-postadress eller en social mediekonto, vilket delvis är ett sätt att uppmuntra mottagaren att lära sig om och ha en andel i valuta. Drickstjänsten förvarar mynten i deposition och mottagaren får ett meddelande som berättar för dem att de har mynt de kan samla in. Mottagaren kan göra anspråk på mynten genom att autentisera sig själv till tjänsten via sin e-postadress eller sociala mediekonto. De måste också installera en plånbok programvara eller möjliggöra ett annat sätt att ta emot mynt. En annan donationsmetod är en kran: dessa är tjänster som ger ut en liten mängd mynt till alla som besöker en webbplats och kanske skriver in ett e-postmeddelande
adress.

10.2 Några Altcoins i detalj

Nu ska vi fokusera på några av de äldsta altcoins och studera deras egenskaper mer i detalj.

Namecoin. Vi har sett hur Bitcoins blockkedja är en säker, global databas. När data har varit skrivet till den är den manipuleringssäker och dess inkludering kan bevisas för alltid. Kan vi ändra Bitcoins design för att stödja andra tillämpningar av säkra globala databaser, till exempel ett namnsystem?

Vi behöver några grundregler för att göra den här databasen mer användbar för applikationer som inte är valutor. Först, vi går med på att se datainmatningar som namn/värdepar, med namn som är globalt unika. Det här tillåter alla att slå upp värdet som är mappat till ett namn, precis som en hashtabell eller en databas med en primärnyckelfält. För att upprätthålla den globala unikheten hos namn, om ett namn/värdepar har samma namn som en tidigare databaspost, då ser vi den som en uppdatering av värdet snarare än en ny post.

För det andra godkänner vi att endast den användare som ursprungligen skapade posten för ett visst namn får göra det göra uppdateringar av det namnet. Vi kan enkelt genomdriva detta genom att associera varje namn med en Bitcoin adress och kräver att uppdateringstransaktionerna signeras av den privata nyckeln för den adressen.

Vi skulle kunna göra allt detta ovanpå Bitcoin, precis som vi sa i kapitel 9 att vi kunde bygga vilket överlägg som helst valuta som använder Bitcoin som en logg med endast tillägg. Men det är enklare att göra det i en altcoin eftersom vi kan ta detta gentleman's agreement och skriv in det i reglerna för altcoin. Dessa regler skulle då vara okränkbar och påtvingad av gruvarbetarna, snarare än att kräva att varje användare (dvs full nod) kontrollerar regler för sig själv och självständigt bestämma vad de ska göra om de överträds. Rätt gjort, det skulle det till och med tillåt SPV-liknande bevis: en lättviktsklient skulle kunna skicka en fråga (dvs ett namn) till en server kör en fullständig nod, och servern skulle returnera ett värde för det namnet, tillsammans med ett bevis på att returnerade värdet i själva verket den senaste uppdateringen för det namnet i databasen.

Det är Namecoin i ett nötskal. Det är en global namn/värdebutik där varje användare kan registrera en eller fler namn (mot en nominell avgift) och sedan utfärda uppdateringar av värdena för något av deras namn. Användare kan också överföra kontrollen över sina namn till andra. Faktum är att du kan göra en transaktion som överför din domän till någon, och samtidigt överför enheter av Namecoin-valutan från dem till du. Eftersom detta är en enda atomär transaktion är det ett säkert sätt att sälja din domän till någon du har aldrig träffat och lita inte på. Från och med 2015 stöder Namecoin inte säkra lättviktsklienter, men en förlängning som stöder detta har föreslagits.

Namecoins mål är att tillhandahålla en decentraliserad version av Domain Name System (DNS), namnen i databasen är domännamn och värdena är IP-adresser. Du kan inte använda detta som standard med en omodifierad webbläsare, men du kan ladda ner en webbläsarplugin för t.ex. Firefox eller Chrome skulle tillåta dig att skriva in en adress som example.bit — vilket domännamn som helst som slutar på .bit — och det kommer att slå upp platsen i Namecoin-registret istället för den traditionella DNS.

Namecoin är tekniskt intressant, och det är också historiskt intressant - det var faktiskt den första altcoinen kommer att lanseras i april 2011, drygt två år efter att Bitcoin lanserades. Den har "sammanfoga gruvdrift" som vi kommer att diskutera senare i detta kapitel.

Namecoin används inte särskilt mycket från och med 2015. De allra flesta registrerade domäner tas av "squatters", i hopp om (men misslyckats hittills) att sälja sina namn med vinst. Namecoin-anhängare brukar hävdar att den befintliga DNS ger för mycket kontroll över en kritisk komponent av Internet i händerna på en enda enhet. Denna uppfattning är populär i Bitcoin-gemenskapen, som du kan föreställa dig, men den
det ser inte ut som att vanliga användare ropar efter ett alternativ till DNS och berövar Namecoin killer-appen den behöver se betydande adoption.

Litecoin. Litecoin lanserades också 2011, en tid efter Namecoin. Under de senaste åren, Litecoin har varit nummer ett altcoin när det gäller övergripande popularitet och användarbas. Det är också mest kluven kodbas. Faktum är att det har klaffats fler gånger än själva Bitcoin.

Den huvudsakliga tekniska skillnaden mellan Litecoin och Bitcoin är att Litecoin har en minneshård gruvpussel (baserat på scrypt), som vi pratade om i kapitel 8. När Litecoin lanserades, Bitcoin-brytning var i GPU-eran, och så målet med Litecoins användning av ett minneshårt gruvpussel var GPU-motstånd. När det lanserades kunde du fortfarande bryta på Litecoin med en CPU, långt efter detta hade blivit meningslöst för Bitcoin. Men sedan dess har Litecoin inte lyckats stå emot övergången till GPU-utvinning och sedan till ASIC. Var och en av dessa gruvövergångar tog lite längre tid i Litecoin än Bitcoin, men det är inte klart om detta beror på att Litecoins pussel faktiskt var svårare att implementera i hårdvara eller helt enkelt för att Litecoins lägre växelkurs gav mindre incitament att göra det.

Hur som helst är prestandaförbättringarna för ASIC:er jämfört med CPU-mining ungefär likadana för Litecoin som de är för Bitcoin. I denna mening misslyckades Litecoin i sitt ursprungliga mål att skapa ett mer decentraliserat system genom att upprätthålla en gemenskap av CPU-gruvarbetare. Men, viktigare, denna berättelse fortfarande
arbetade för bootstrapping Litecoin — det lockade många adoptanter som slutade stanna även efter ursprungliga premissen misslyckades. Litecoin har sedan dess explicit ändrat sin berättelse och säger att dess initiala
allokeringen var mer rättvis än Bitcoins eftersom den motstod ASICs längre.

Litecoin gör också några mindre parameterändringar: till exempel kommer block i Litecoin fyra gånger snabbare än i Bitcoin, var 2,5 minut. Litecoin lånar annars så mycket från Bitcoin som möjligt. Faktum är att dess utveckling har följt Bitcoin, så att patchar och förbättringar har gjorts till Bitcoin har Litecoin även antagit dessa.

Peercoin. Peercoin, ibland kallad PPCoin, lanserades i slutet av 2012 och var det första altcoin som använda proof-of-stake gruvdrift. Vi diskuterade proof-of-stake mining (och Peercoins implementering av det) i kapitel 8, men Peercoin är intressant att diskutera av ytterligare en helt annan anledning: dess administratörer har en betrodd offentlig nyckel som de använder för att tilldela kontrollpunkter för "välsignade" block
då och då. Detta är tänkt att fungera som ett skydd mot gaffelattacker, men det är kontroversiellt eftersom administratörernas förmåga att kontrollera systemet betyder att Peercoin inte är riktigt decentraliserat. Kontrollpunktssystemet är inte naturligt för Peercoin och kan tas bort i framtiden;

dess existens innebär ändå att vi inte kan dra slutsatsen att bevis på insats har lett till ett säkert system i öva. Vi vet inte vad som skulle hända om detta skydd togs bort.

Figur 10.4: En av flera Dogecoin-logotyper. Försäljningsargumentet är humor mer än teknisk innovation.

Dogecoin. Dogecoin har kanske varit den mest färgstarka av alla altcoins hittills. Den släpptes sent 2013, och det som utmärker det är inte i första hand tekniskt (det är en nära gaffel av Litecoin) utan snarare en uppsättning
av gemenskapsvärden: dricks, generositet och att inte ta kryptovaluta på så stort allvar. Det är det verkligen uppkallad efter Doge, ett underhållande internetmeme med en grammatiskt utmanad Shiba Inu-hund. Gemenskapen har haft flera intressanta och framgångsrika marknadsföringskampanjer som att sponsra en NASCAR-förare och sätter Dogecoin-logotyper över hela sin bil. De samlade också in över $30 000 för att stödja Jamaicas nationella Bobsled Team så att de kunde resa och tävla under vintern 2014
OS. Lustigt, detta noggrant speglar tomten till 90-talet filmen *Cool Runnings* .

Kombinationen av samhällets generositet, PR-aktiviteter och det inneboende memevärdet hos Doge innebar att Dogecoin blev väldigt populärt 2014. Det verkar som att många av de tidiga adopterna var det obekant med kryptovalutor före Dogecoin, vilket ger en ny community att starta upp
valutans värde utan att behöva erbjuda en övertygande historia när det gäller fördelar gentemot andra valutor. Dogecoin visade att bootstrapping kan vara framgångsrikt med en icke-teknisk berättelse. Tyvärr, liksom många internetfenomen, har populariteten inte bestått och Dogecoins utbyte
kursen har sedan dess sjunkit.

10.3 Förhållandet mellan Bitcoin och Altcoins

För att få en känsla av den relativa storleken eller effekten av olika altcoins finns det en mängd olika mätvärden vi kan
använda sig av.

Jämföra altcoins: börsvärde. Traditionellt är börsvärde eller börsvärde ett enkel metod för att uppskatta värdet av ett offentligt bolag genom att multiplicera priset på en aktie med det totala antalet utestående aktier. I samband med altcoins är detta marknadsvärde ofta liknande används för att uppskatta det totala värdet av altcoin genom att multiplicera priset på en individuell enhet av altcoin (mätt, kanske, vid de mest populära tredjepartsbörserna) med det totala antalet enheter av valutan för det altcoin som tros vara i omlopp. Med detta mått är Bitcoin överlägset störst — som 2015 står den för över 90 % av det totala marknadsvärdet för alla kryptovalutor tillsammans. Den relativa rankningen av de andra altcoins tenderar att variera ganska mycket, men poängen är att de flesta altcoins
är jämförelsevis små när det gäller penningvärde.

Det är viktigt att inte läsa för mycket i börsvärdet. För det första är det inte nödvändigtvis hur mycket det skulle göra
kostnad för någon att köpa upp alla mynt i omlopp. Den siffran kan vara högre eller lägre, eftersom

stora beställningar kommer att flytta priset på valutan. För det andra, även om beräkningen endast beaktar de mynt som för närvarande är i omlopp, bör vi förvänta oss att marknadsaktörerna tar med i börsen betygsätta det faktum att nya mynt kommer i omlopp i framtiden, vilket ytterligare komplicerar tolkning av numret. Slutligen kan vi inte ens exakt uppskatta det verkliga antalet mynt för närvarande i omlopp eftersom ägarna till vissa mynt kan ha tappat sina privata nycklar, och det skulle vi har inget sätt att veta.

Jämföra altcoins: gruvkraft. Om två altcoins använder samma gruvpussel kan vi direkt jämför dem med hur mycket gruvkraft alla altcoins gruvarbetare har. Detta kallas ofta bara hashfrekvensen på grund av hashbaserade pussel framträdande. Till exempel är Zetacoin ett altcoin som använder SHA-256 gruvpussel, precis som Bitcoin gör, och har en nätverkshashhastighet på cirka 5 Terahashes / sekund (5 * 10^{12} hash/sekund) från och med december 2015. Detta antal är ungefär en hundratusendel av Bitcoins gruvkraft. Det är knepigare att jämföra gruvkraften mellan mynt som använder olika gruvpussel eftersom pusslen kan ta olika lång tid att beräkna. Dessutom kommer gruvhårdvara specialiserad för ett av mynten inte nödvändigtvis att användas för bryta (inklusive attackera) det andra myntet.

Även för en altcoin som använder ett helt unikt gruvpussel kan vi fortfarande lära oss något av relativ förändring i gruvkraft över tid. Tillväxt i gruvkraft indikerar antingen det mer deltagare har gått med eller att de har uppgraderat till mer kraftfull gruvutrustning. Förlust av gruvkraft innebär vanligtvis att vissa gruvarbetare har övergett altcoin och är vanligtvis en olycksbådande skylt.

Jämföra altcoins: andra indikatorer. Det finns flera andra indikatorer vi kan titta på. Ändringar i en altcoins växelkurs över tid ger oss ledtrådar om dess hälsa och tenderar att korrelera med förändringar i sin hashhastighet över långa tidsperioder. Utbytesvolymen på olika tredjepartsbörser är en mått på aktivitet och intresse för altcoin. Å andra sidan, volymen av transaktioner som har gjorts på altcoins blockkedja säger oss inte så mycket, eftersom det helt enkelt kan vara användare blandar runt sina egna mynt i plånboken, kanske till och med automatiskt. Äntligen kan vi också titta på hur många handlare och betalningsprocessorer som stöder altcoin - bara de mest framträdande sådana tenderar att stödjas av betalningsprocessorer.

Den ekonomiska synen på Bitcoin-altcoin-interaktioner. Relationen mellan Bitcoin och altcoins är komplicerad. I en mening konkurrerar kryptovalutor med varandra, eftersom de alla erbjuder ett sätt att göra onlinebetalningar. Om det finns två standarder, protokoll eller format i konkurrens som är ungefär likvärdiga vad gäller vad de erbjuder, så kommer en av dem vanligtvis att dominera, pga vad ekonomer kallar "nätverkseffekter".

Till exempel var Blu-ray och HD-DVD i hård konkurrens i mitten till slutet av 2000-talet för att vara efterföljare till DVD-formatet. Successivt började Blu-ray bli mer populärt, till stor del eftersom den populära PlayStation 3-konsolen fungerade som en Blu-ray-spelare. Detta gjorde Blu-ray till en mer attraktivt format för filmstudior och denna popularitet livnärde sig: allt eftersom fler filmer släpptes för Blu-ray, fler konsumenter köpte fristående Blu-ray-spelare, vilket ledde till fler filmsläpp och så på. På samma sätt, om alla dina vänner har Blu-ray-spelare, skulle du vilja köpa en själv istället för en HD DVD-spelare eftersom du enkelt skulle kunna byta filmer med dem. Inom cirka två år, HD DVD

var en historisk fotnot.

Sidebar: vem vinner loppet? Långt innan HD DVD har det funnits otaliga exempel på tekniska standarder som snabbt förlorade mot en konkurrent och gled in i dunkel, från Betamax analoga videoband till ryska järnvägsspår. Om du aldrig har hört talas om dessa, nätverkseffekter är anledningen till detta. Ibland, som i fallet med Thomas Edisons likström elnät kontra Nikola Teslas växelströmsnät, vinnaren (AC) bestämdes av överväldigande teknisk överlägsenhet. I många andra fall dock, såsom Betamax-band som förlorar mot VHS-band, förloraren kan faktiskt ha varit tekniskt överlägsen, med nätverkseffekter som är starka tillräckligt för att övervinna en liten teknisk nackdel.

Detta resonemang antyder att en kryptovaluta kommer att dominera - förmodligen Bitcoin, vilket är överlägset den mest populära idag - även om vissa efterföljande system skulle kunna hävdas vara det tekniskt överlägsen. Men det vore en överenkling. Det finns åtminstone två anledningar till det konkurrensen mellan kryptovalutor är inte lika fientlig som konkurrensen mellan skivformat.

För det första är det relativt enkelt för användare att konvertera en kryptovaluta till en annan, och för leverantörer att
acceptera mer än en kryptovaluta, vilket innebär att flera kryptovalutor lättare kan samexistera och frodas. I ekonomi termer cryptocurrencies uppvisar relativt låga *kostnader kopplings* . Jämför med DVD-spelare, där de flesta verkligen inte vill ha två skrymmande maskiner i sitt hem och kan inte konvertera sitt befintliga bibliotek med skivor om de byter till en maskin som spelar det andra formatet. [274]

Byteskostnaderna är absolut inte noll för kryptovalutor. Användare kan till exempel köpa hårdvara plånböcker som inte kan uppgraderas. Men i stort sett är det enkelt att byta kryptovaluta eller att använda mer än en på samma gång.

För det andra, som vi sa tidigare, har många altcoins unika egenskaper som ger en distinkt anledning till existerande. Dessa altcoins ska inte ses som bara substitut för Bitcoin; de kan vara ortogonala, eller kanske till och med kompletterande. Sett på detta sätt ökar komplementära altcoins faktiskt användbarheten av Bitcoin snarare än att konkurrera med den. Om Namecoin lyckas, till exempel Bitcoin-användare
har ytterligare en användbar sak de kan göra med sina bitcoins.

Men denna bild av lyckligt samarbete är också en överförenkling. Vissa altcoins, som Litecoin, helt enkelt försök att uppnå samma funktionalitet som Bitcoin men på ett annat, kanske mer effektivt sätt. Även när ny funktionalitet erbjuds kan ofta dessa användningsfall faktiskt uppnås inom Bitcoin i sig, om än på ett mindre elegant sätt (vi kommer att ha mer att säga om detta i kapitel 11). Supportrar av do-it-on-top-of-Bitcoin-modellen hävdar att att ha många altcoins delar hashkraften tillgänglig och gör varje valuta mindre säker.

Anhängare av altcoins hävdar däremot att altcoins tillåter marknadskrafterna att avgöra vilka funktioner är värda att ha, vilka system som är tekniskt överlägsna och så vidare. Det hävdar de vidare att ha många altcoins begränsar skadan av ett potentiellt katastrofalt fel i ett system. De påpekar också att Bitcoin-utvecklare är mycket riskvilliga och att lägga till nya funktioner Bitcoin via en mjuk eller en hård gaffel är långsam och svår. Å andra sidan är det lätt att testa en ny idé via ett altcoin; altcoins kan ses som en testbädd för forskning och utveckling för potentiella Bitcoin

funktioner.

Det praktiska resultatet är att det finns en viss spänning mellan anhängare av Bitcoin och de av altcoins, men också en känsla av samarbete.

10.4 Sammanfoga gruvdrift

I det här avsnittet och nästa kommer vi att lägga frågor om kultur, politik och ekonomi åt sidan. Det gör vi istället fokusera på den tekniska interaktionen mellan Bitcoin och altcoins.

Barnmord på Altcoin. Från och med 2015 överstiger Bitcoins hashkraft den för alla andra altcoins. Sannerligen, där
är kraftfulla gruvarbetare och gruvpooler som kontrollerar mer gruvkraft än hela altcoins. En sådan gruvarbetare eller entitet skulle lätt kunna utföra en attack mot en liten altcoin (om den använder samma SHA-256 gruvpussel som Bitcoin), vilket orsakar gafflar och allmän förödelse som ofta räcker för att döda altcoin. Vi kallar detta fenomen *altcoin barnamord* .

Varför skulle någon göra detta, med tanke på att de måste använda sin värdefulla gruvkraft för att göra det och kommer inte att göra det
få en betydande monetär belöning? Ta fallet med 2012 års attack på en liten altcoin som heter CoiledCoin: Operatören av Bitcoin gruvpool Eligius beslutade att CoiledCoin var en bluff och en

---

kränkning av kryptovalutans ekosystem. Så Eligius riktade sina gruvresurser mot CoiledCoin, gruvdrift block som vänder på dagars värde av CoiledCoin-transaktionshistorik samt utvinner en lång kedja med tomma block, vilket effektivt orsakade en överbelastningsattack som hindrade CoiledCoin-användare från göra några transaktioner. Efter en ganska kort belägring övergav användare CoiledCoin, och det existerar inte längre. I detta exempel och i andra altcoin-barnmordsattacker motiveras angriparen av
något annat än direkt vinst.

Slå samman gruvdrift. Som standard — säg om en altcoin klaffar Bitcoin-källkoden men inte gör någon annan ändringar — gruvdrift på altcoin är exklusiv. Det vill säga, du kan försöka lösa gruvpussellösningen till hitta ett giltigt block för altcoin eller för Bitcoin, men du kan inte försöka lösa båda pusslen samtidigt. Av naturligtvis kan du dela upp dina gruvresurser för att dedikera en del till gruvdrift på altcoin och en del till gruvdrift på Bitcoin. Du kan till och med dela mellan flera olika altcoins och du kan justera dina allokering över tid, men det finns inget sätt att få din gruvkraft att utföra dubbelt arbete.

Med exklusiv gruvdrift kan nätverkseffekter göra det svårt för en altcoin att starta upp. Om du ville för att lansera en altcoin och övertyga dagens Bitcoin-gruvarbetare att delta i ditt nätverk, skulle de göra det måste sluta bryta Bitcoin (med åtminstone några av deras resurser) vilket kommer att innebära en omedelbar förlust
av belöningar för Bitcoin-gruvdrift. Detta betyder att din altcoin sannolikt kommer att förbli liten när det gäller hashkraft
och mer sårbara för attacker i form av barnmord från Bitcoin-gruvarbetare.

Kan vi designa ett altcoin så att det är möjligt att bryta block både på altcoin och på Bitcoin vid samma tid? För att göra det måste vi skapa block som inkluderar transaktioner från både Bitcoin och altcoin, vilket gör dem giltiga i båda blockkedjorna. Det är enkelt att designa altcoin så att det tillåter Bitcoin

transaktioner i dess block, eftersom vi kan skriva reglerna för altcoin hur vi vill. Motsatsen är svårare. Var kan vi placera altcoin-transaktioner i Bitcoin-block? I kapitel 3 och senare i 9 kap vi har sett hur man lägger in godtyckliga data i Bitcoin-block, men bandbredden för dessa metoder är mycket begränsad.

Det är ett trick, men: även om vi inte kan sätta *innehållet* i altcoin transaktioner i Bitcoin block, kan vi sätta en *sammanfattning* av altcoin transaktioner till Bitcoin block i form av en hash pekare till altcoin-blocket. Att hitta ett sätt att sätta en enda hash-pekare i varje Bitcoin-block är enkelt. Kom ihåg att varje Bitcoin-block har en speciell transaktion som kallas myntbastransaktionen det är där gruvarbetaren skapar nya mynt som en blockbelöning. scriptSig-fältet för denna transaktion har ingen betydelse och kan därför användas för att lagra godtyckliga data (det finns inget behov av att signera Coinbase-transaktion eftersom den inte spenderar några tidigare transaktionsutdata). Så i en sammansmältning altcoin är gruv uppgift att beräkna *Bitcoin* block vars Coinbase scriptsig innehåller en hash pekare till ett altcoinblock.

Det här blocket kan nu göra dubbla uppgifter: för Bitcoin-klienter ser det ut precis som vilket annat Bitcoin-block som helst, med ett
hash i myntbastransaktionen som kan ignoreras. Altcoin-klienter vet hur man tolkar blocket genom att ignorera Bitcoin-transaktionerna och titta på altcoin-transaktionerna som hashen gör i myntbastransaktionen. Observera att även om detta inte kräver några ändringar av Bitcoin, så gör det det kräva att altcoin specifikt förstår Bitcoin och accepterar merge-mined block.

Om vår altcoin är merge-mined, hoppas vi att många Bitcoin-gruvarbetare kommer att bryta det, för att göra det gör det inte
kräver ytterligare hashkraft. Det kräver en del ytterligare beräkningsresurser för bearbetning av block och transaktioner, och gruvarbetare behöver veta och bry sig tillräckligt mycket om vår altcoin för att
bry sig till och med att bryta det. Låt oss säga att 25 % av Bitcoin-gruvarbetarna med hashkraft bryter vår altcoin. Detta
innebär att i genomsnitt 25% av Bitcoin-blocken innehåller pekare till altcoin-block. Det verkar alltså som om i vårt altcoin ett nytt block skulle mineras i genomsnitt var 40:e minut. Värre, medan altcoin är stilla är bootstrappad och bråkdelen av Bitcoin-gruvarbetare som bryter den är mycket liten, tiden mellan blocken kommer att vara timmar eller dagar, vilket är oacceptabelt.

Kan vi säkerställa att block av ett sammanslagna altcoin skapas i en jämn takt, lika hög eller låg som vi vill, oavsett andelen Bitcoin-gruvarbetare som bryter det? Svaret är ja. Tricket är att till och med om gruv uppgift för altcoin är densamma som i Bitcoin, *gruvmålet* behöver inte vara. Altcoin-nätverket beräknar målet och svårigheten för sina block oberoende av Bitcoin nätverk. Precis som Bitcoin justerar sitt gruvmål så att block hittas var tionde minut I genomsnitt skulle altcoin justera sitt eget mål så att block i altcoin hittas var 10:e minuter eller något annat fast värde.

Altcoin block

Bitcoin-block utvunna av altcoin merge-miners

Bitcoin-block utvunna av icke-altcoin-gruvarbetare

Försök med Bitcoin-block som hittats av altcoin merge-miners som mötte altcoins svårighetsgrad
mål men inte Bitcoins mål
Figur 10.5: sammanslagning av gruvdrift.

Detta innebär att altcoins mål vanligtvis kommer att vara mycket mindre än Bitcoins mål, och några (eller till och med
de flesta) altcoin-block kommer inte att pekas på av giltiga Bitcoin-block. Men det är okej! Du borde tänka på Bitcoin-blockkedjan och altcoin-blockkedjan som två parallella kedjor, med enstaka pekare
från ett Bitcoin-block till ett altcoin-block. Detta illustreras i figur 10.5. I det här exemplet, 60 % av Bitcoin-gruvarbetare bryter altcoin, och altcoins tid mellan blocken är 5 minuter. Detta innebär att

---

altcoins svårighetsgrad är 60% * 5 / 10 = 30% av Bitcoin. Observera att 40 % av Bitcoin-blocken inte gör det innehåller hash-pekare till altcoin-block.

Omvänt är varje giltigt altcoin-block ett resultat av ett försök att bryta ett Bitcoin-block, men bara 30 % av dem uppfyller faktiskt Bitcoins svårighetsmål. För de övriga 70% av altcoin-blocken, altcoin nätverk måste kunna verifiera gruvpussellösningen. Det enkla sättet att göra detta är att sända Bitcoin-nära-blocket utöver altcoin-blocket. Men ett smartare sätt är att sända bara rubriken på Bitcoin-nära-blocket och Merkle-beviset på inkludering av Coinbase-transaktionen i Bitcoin-blocket.

Det är också möjligt (men sällan sett) för altcoin att faktiskt ha en *svårare* pussel än Bitcoin. Detta är ovanligt eftersom de flesta altcoins vill ha block hittade oftare än en gång per 10 minuter, men om du av någon anledning ville ha en lägre hastighet skulle detta också vara lätt att uppnå. I denna fall, du skulle se några Bitcoin-block som gruvarbetaren hoppades skulle också vara ett altcoin-block, men kommer att avvisas på altcoin-nätverket eftersom de inte uppfyllde det svårare målet.

Slutligen, notera att valfritt antal altcoins samtidigt kan slås samman med Bitcoin, och varje miner är fri att välja en godtycklig delmängd av altcoins för att slå samman mina. I det här fallet, Coinbase scriptSig
skulle i sig vara ett Merkle-träd av hashpekare till olika altcoin-block. Notera komplexitetsnivåerna: verifiering av inkluderingen av en altcoin-transaktion kräver verifiering av bland annat: (1) en Merkle bevis på inkludering av altcoin-transaktionen i altcoin-blocket (2) ett Merkle-bevis för inkludering av altcoin block hash i Coinbase scriptSig och (3) ett Merkle bevis på inkludering av Coinbase scriptSigga i Bitcoin-blocket eller nära-blocket!

Sammanfoga gruvdrift och säkerhet. Merge mining är en blandad välsignelse. Det gör bootstrapping lättare, som
vi har diskuterat, och den resulterande ökningen av din altcoins totala hashkraft ökar dess motståndskraft mot ge sig på. En motståndare som vill köpa datorkraft för att förstöra din altcoin kommer att behöva göra en enorm förhandsinvestering.

Å andra sidan kan man hävda att detta är en falsk känsla av säkerhet, eftersom en sådan motståndare

skulle förmodligen få tillbaka kostnaden för sin investering genom att bryta Bitcoin, och marginalkostnaden till attackera din altcoin är trivialt. Detta är lättare att uppskatta om vi tänker på en motståndare som redan är en stor Bitcoin-gruvarbetare. Det var faktiskt CoiledCoin, altcoint som beskrevs tidigare och som drabbades av barnmord
sammansmälta. Eligius-gruvpoolen och dess deltagare behövde inte stoppa Bitcoin-brytningen i ordning att attackera den. Faktum är att pooldeltagarna inte ens var medvetna om att deras datorresurser var det används i attacken!

Sidofält: trender inom gruvpussel med altcoin. Från och med 2015 lanseras få altcoins med samma SHA-256 gruvpussel som Bitcoin, med eller utan merge mining, vilket antyder att det kanske är det anses vara en säkerhetsrisk. Scrypt är ett mycket mer populärt val, vilket gör Bitcoin ASICs värdelösa för att bryta eller attackera sådana altcoins. Naturligtvis krypterar ASICs som tillverkas för Litecoin gruvdrift kunde användas för att attackera dem.
278

När vi tänker på en rationell gruvarbetare som bestämmer sig för att slå ihop min eller inte, hittar vi mer problem med säkerheten för merge mining. Minns att grovt sett är gruvdrift vettigt om förväntad belöning är lika med eller överstiger de förväntade kostnaderna. För Bitcoin-brytning är kostnaden i första hand den för
hashberäkning. Men för någon som redan är en Bitcoin-gruvarbetare som bestämmer sig för att slå samman eller inte
bryta ett altcoin, det tillkommer ingen extra kostnad från hash. Istället uppstår merkostnaderna från två faktorer: beräkningen, bandbredden och lagringen som behövs för att validera altcoin-transaktionerna, och behöver hålla mjukvaran uppdaterad och kanske fatta välgrundade beslut om altcoin är under hårda eller mjuka gafflar.

Detta resonemang ger två insikter. För det första har merge mining starka stordriftsfördelar, eftersom alla gruvarbetare ådrar sig ungefär samma kostnader oavsett deras hashkraft. Detta står i skarp kontrast till Bitcoin där kostnaden är proportionell mot hashkraft, till en första approximation. Så för ett lågt värde altcoin, en liten solo miner kommer att tycka att det är olönsamt att slå samman min det eftersom kostnaden överstiger den magra belöningen de
kommer att göra på grund av deras låga hashkraft. Tänk på att från och med 2015 kommer de potentiella intäkterna från
att bryta altcoins är fortfarande en liten del av Bitcoin-gruvintäkterna. Detta förutspår att jämfört med Bitcoin, merge-mined altcoins kommer att ha en större centralisering eller koncentration av gruvkraft.

En relaterad förutsägelse är att de flesta gruvarbetare kommer att välja att lägga ut sin transaktionsvalidering på entreprenad. De
mindre altcoin, desto större incitament att lägga ut på entreprenad. Det naturliga sättet att göra detta är att gå med i en
Bitcoin gruvpool. Det beror på att pooler vanligtvis tar dessa beräkningar ur gruvarbetarnas händer. De pooloperatören sätter ihop ett Bitcoin-block som innehåller block från (noll eller fler) altcoins, efter validerar transaktionerna i Bitcoin-blocket såväl som alla dessa altcoin-block. Gruvarbetaren bara försöker lösa för nonce. Dessa förutsägelser bekräftas i praktiken. Till exempel, GHash.IO, at en gång den största Bitcoin gruvpool, tillåter sammanslagning av gruvdrift Namecoin, IXCoin och DevCoin. Så dessa blev de mest populära sammanslagna altmynten.

Den andra insikten från det ekonomiska resonemanget är kanske ännu mer oroande för säkerheten än koncentration av gruvkraft. När gruvarbetarnas primära kostnad är bevis på arbete, finns det ingen möjlighet för gruvarbetare att "fuska". Det finns ingen genväg till gruvdrift med tanke på säkerheten för hashfunktioner, och Dessutom kan och kommer andra gruvarbetare lätt att verifiera beviset på arbetet. Båda antagandena misslyckas när
kostnaden är den för transaktionsvalidering. En gruvarbetare kan anta att transaktioner de hört talas om är giltigt och hoppas komma undan med det. Dessutom för andra gruvarbetare att validera ett block och dess transaktioner
lika mycket arbete som det var för gruvarbetaren som hittade den. Av dessa skäl bör vi förvänta oss att kl åtminstone för små sammanslagna gruvarbetare finns det ett incitament att snåla med validering. Förekomsten av felaktigt
validering av gruvarbetare gör attacker lättare eftersom en illvillig gruvarbetare kan skapa ett block som kommer att orsaka
resten av gruvarbetarna att vara oense om vad den längsta giltiga grenen är.

Sammanfattningsvis löser merge mining ett säkerhetsproblem men skapar många andra, delvis pga ekonomin för merge mining skiljer sig på viktiga sätt från ekonomin för exklusiv gruvdrift.
Sammantaget är det långt ifrån klart att merge mining är en bra idé för en ny altcoin som är oroad över gruvdrift attacker.

10.5 Atomic Cross-chain Swaps

I Bitcoin är det enkelt att skapa en enda transaktion som byter valuta eller kontrollerade tillgångar av olika personer eller enheter. Detta är intuitionen bakom Coinjoin, som vi studerade i kapitel 6. Det är också användbar för handel med smart egendom, som vi tittade på kort i kapitel 9 och kommer att återkomma till i
Kapitel 11. Samma idé gör det möjligt att sälja domännamn i Namecoin, som nämnts tidigare i detta kapitel.

Men i alla dessa fall är swaptransaktionerna begränsade till en enda blockkedja, även om de involverar olika typer av tillgångar inom den blockkedjan. I allmänhet är en transaktion på en altcoin helt oberoende av och har inget sätt att referera till en transaktion som sker på andra altcoins transaktionshistorik. Men är detta en grundläggande begränsning, eller finns det något sätt att byta en typ av mynt för en annan? Dvs om Alice vill sälja en kvantitet *ett* av altcoin till Bob i utbyte för en kvantitet *b* av hans bitcoin, kan de göra det på ett atomärt sätt, utan att behöva lita på varandra eller förlita sig på en mellanhand, till exempel en utbytestjänst? Vid första anblicken verkar detta omöjligt, eftersom det inte finns sätt att tvinga transaktioner på två olika blockkedjor att ske samtidigt. Om någon av dem, säg Alice, genomför sin överföring före den andra, vilket hindrar honom från att avstå från sin sida förhandla?

Lösningen är smart och involverar kryptografiska åtaganden och tidslåsta insättningar, båda vilket är tekniker vi har sett tidigare. Figur 10.6 beskriver protokollet. För tillfället, anta att block i de två blockkedjorna genereras i låssteg: ett block genereras varje tidsenhet.
Låt T representerar den tid i början av protokollet.

1.
Alice genererar en deposition på *ett* altcoins enligt följande:
1,1 Alice genererar en slumpmässig sträng x och beräknar hash $h = H(x)$
1,2 Alice genererar **DepositA** såsom visas nedan, men inte publicera den ännu
1,3 Alice genererar **RefundA** , och får Bobs signatur på det
1,4 När Bob signerar **RefundA** , publicerar hon DepositA (men inte offentliggör **RefundA**)
2.
Bob genererar en deposition på b Bitcoins enligt följande:
2,1 Bob genererar **DepositB** såsom visas nedan, men inte publicera den ännu
2,2 Bob genererar **RefundB** , och får Alices signatur på det
2,2 När Alice tecknar **RefundB** , publicerar han **DepositB** (men inte offentliggör **RefundB**)
3.
Fall 1: Alice går igenom bytet
3,1 Alice hävdar de Bitcoins vid tid T_1 , avslöjande x till Bob (och alla) vid förfarandet
3,2 Bob hävdar de altcoins med tiden T_2
Fall 2: Alice ändrar sig, gör inte anspråk på de altcoins, avslöjar inte x till Bob
3,1 Bob hävdar hans altcoin återbetalning vid tiden T_1
3,2 Alice hävdar her Bitcoin återbetalning vid tiden T_2

InsättningA [Altcoin block chain]

Inmatning:
Alices mynt av värde a
ScriptPubkey: Lös in genom att tillhandahålla
antingen ($sIgA$ och $sigB$)
eller $sigB$ och x st $H(x) = <h>$

→
RefundA [Altcoin block chain]

Inmatning:
DepositionA
Produktion:
AddrA
Timelock:
T_2
ScriptSig:
sigA, sigB

DepositB [Bitcoin blockchain]

Inmatning:

Bobs mynt av värde *b*
ScriptPubkey: Lös in genom att tillhandahålla
antingen (*sIgA* och *sigB*)
eller *sIgA* och *x* st *H (x)* = <*h*>

→
ÅterbetalningB [Bitcoin blockchain]

Inmatning:
InsättningB
Produktion:
AddrB
Timelock:
T_1
ScriptSig:
sigA, sigB

Figur 10.6: Atomic cross-chain swap-protokoll

I steg 1, Alice insättningar altcoins värde *ett* så som kan lösas in i ett av två sätt (en ”insättning” innebär helt enkelt att skicka dessa mynt till en ScriptPubkey som anger två möjliga villkor för spendera det). För det första, om Alice och Bob kommer överens kan de lösa in det. Faktum är att Alice publicerar sätta in först efter att ha sett till att få en återbetalningstransaktion undertecknad av Bob – detta gör att hon kan lösa in
hennes insättning om 2 tidsenheter förflutit och den inte redan har begärts.

Det andra sättet att göra anspråk på Alices insättning, när som helst, är genom att tillhandahålla Bobs underskrift samt värdet
x som öppnar hash engagemang *h.* Observera att vi skriver <*h*> i *DepositA* som tyder på att Alice bokstavligen skriver värdet på *H* i ScriptPubkey. Eftersom *x* är känd endast till Alice, vid slutet av steg 1 ingen av parterna kan göra anspråk på depositionen på detta sätt. Idén är att Bob kommer att lära sig värdet *x* , gör det möjligt för honom att göra anspråk på altcoins, om och bara om Alice gör anspråk på sina bitcoins, som vi kommer att se.

Steg 2 är ungefär motsatsen till steg 1: Bob insättningar Bitcoins värde *b* så att den kan lösas ut i ett av två sätt. Den viktigaste skillnaden är att han inte väljer en ny hemlighet; istället använder han detsamma hashvärde *h* (han skulle bara kopiera värdet från *DepositA* transaktionen till *DepositB* transaktion). Detta är nyckeln till att knyta samman transaktioner på de två blockkedjorna.

Vid det här laget är bollen hos Alice. Hon kunde ändra sig om swap - om vid tiden T_1
Alice
har inte gjort något för att avslöja *x* Bob, kommer han helt enkelt hävdar sin insättning och avsluta protokollet. Alices
andra alternativet är att krav Bobs Bitcoins innan tiden T_1
. Men hon kan bara göra detta genom att skapa och
sända en ScriptSig som innehåller värdet *x* ; Bob kan lyssna på den här sändningen och använda värdet samma *x* krav Alices altcoins, slutföra swap.

Observera att om Alice försöker krav Bobs Bitcoins en smula för sent (efter tiden T_1 men innan tiden T_2), Bob
skulle kunna göra anspråk *både* insättningar. Likaså om Alice gör anspråk på Bobs bitcoins i tid men Bob väntar också
281

---

länge, kanske Alice kan åka hem med båda insättningarna. Men detta är inget problem: vi är glada som så länge det inte finns något sätt för en spelare som avviker från protokollet att lura den andra spelaren.

Slutligen, block i Bitcoin eller någon altcoin kommer inte fram i fasta tidssteg, vilket introducerar en del stökighet, särskilt eftersom de två kedjorna kanske inte är synkroniserade. Låt oss säga att båda blockkedjorna har en
genomsnittlig tid på 10 minuter mellan blocken. Då skulle vi vilja välja en "tidsenhet" på säg 1 timme. I Med andra ord, skulle vi vill ha T_1 vara minst current_altcoin_block + 12 och T_2 vara åtminstone
current_bitcoin_block + 6, möjligen med en större säkerhetsmarginal.

Tyvärr finns det en liten men inte noll chans att nästa 12 altcoin-block kommer att hittas innan de nästa 6 Bitcoin-blocken. I det här fallet kanske Alice kan göra anspråk på båda insättningarna. Detta sannolikhet kan göras godtyckligt liten genom att öka tidsenheten, men på bekostnad av hastigheten.

Detta är ett snyggt protokoll, men från och med 2015 använder ingen det. Istället handlas kryptovalutor på traditionella, centraliserade utbyten. Det finns många anledningar till detta. Den första är komplexiteten, olägenhet och långsamhet i protokollet. För det andra, även om protokollet förhindrar stöld, kan det inte förhindra en denial of service. Någon kanske annonserar erbjudanden till fantastiska växelkurser, bara för att sluta efter steg 1 eller steg 2, slösa bort alla andras tid. För att mildra detta och att aggregera och matcha människors erbjudanden behöver du förmodligen en centraliserad växel ändå -- om än en du inte behöver lita på att inte stjäla dina mynt -- vilket ytterligare minskar användbarheten av protokollet.

10.6 Bitcoin-backed Altcoins, "sidokedjor"

Tidigare i detta kapitel talade vi om två sätt på vilka vi kan allokera enheter av ett nytt altcoin till befintliga ägare av bitcoins: antingen kräver bevisligen brännande bitcoins för att skaffa altcoins, eller helt enkelt allokera altcoins till befintliga innehavare av bitcoins baserat på bitcoinadresser som äger outnyttjade transaktionsutdata. Som vi såg tillåter ingen av dessa att bilateralt koppla priset på altcoin till det för Bitcoin. Utan sådan pegging kommer priset på ett altcoin sannolikt att vara volatilt under dess bootstrapping-fasen. Motivationen för sidokedjor är uppfattningen att denna prisvolatilitet är problematisk: det är en distraktion och gör det svårt för altcoins att konkurrera på sina tekniska meriter.

Här är vad vi behöver när det gäller tekniska funktioner för att faktiskt kunna koppla altcoins pris till Bitcoin är till en fast växelkurs. Först bör du kunna lägga en bitcoin som du äger i några sorts deposition och prägla ett altcoin (eller en fast mängd altcoins). Du borde kunna spendera detta altcoin normalt på altcoin-blockkedjan. Slutligen bör du kunna bränna ett altcoin som du äger och lösa in en tidigare spärrad bitcoin. Detta liknar Zerocoin, dit vi deponerar basmynt till skapa nollmynt, men skillnaden är att här måste vi göra det över två olika blockkedjor.

Den dåliga nyheten är att så vitt vi vet finns det inget sätt att uppnå detta utan att modifiera Bitcoin, eftersom Bitcoin-transaktioner inte kan vara beroende av händelser som händer i en annan blockkedja. Bitcoin

skriptet är helt enkelt inte tillräckligt kraftfullt för att verifiera en hel separat blockkedja. Den goda nyheten är att det
kan aktiveras med en relativt praktisk soft-fork modifiering till Bitcoin, och det är tanken bakom
282

Sidokedjor. Sidokedjornas vision är många blomstrande altcoins som snabbt förnyar och
experiment med Bitcoin som en sorts reservvaluta. Från och med 2015 är det bara ett förslag, men ett så
arbetas aktivt på och har ett seriöst grepp i Bitcoin-gemenskapen. Förslaget finns fortfarande kvar
flux, och vi tar oss friheten att förenkla vissa detaljer för pedagogiska ändamål.

Det uppenbara men opraktiska sättet att utöka Bitcoin för att tillåta konvertering av mynt från en sidokedja tillbaka till
bitcoins är detta: koda alla sidokedjans regler i Bitcoin, inklusive validering av alla
sidokedjans transaktioner och kontrollera sidokedjans arbetsbevis. Anledningen till att detta är opraktiskt är
att de resulterande tilläggen till Bitcoins skript skulle vara för komplexa, och verifieringsarbetet
behövs för Bitcoin-noder skulle vara oöverkomligt. Dessutom skulle komplexiteten och ansträngningen växa med
antalet fästa sidokedjor.

SPV-tricket. Tricket för att undvika denna komplexitet är att använda "SPV-bevis." Minns från kapitel 3 att
Förenklad betalningsverifiering används av lättviktsklienter som mobilappar för Bitcoin. SPV
noder validerar inte transaktioner de inte är intresserade av; de verifierar bara blockrubriker. Istället
för att oroa den längsta *giltiga* gren, SPV klienter bara leta efter bevis för att transaktionen
de bryr sig om är i den längsta grenen, giltig eller inte, och att den har fått något antal
bekräftelser. De antar att gruvarbetarna som skapade dessa block inte skulle ha ansträngt sig
att bryta dem utan att validera transaktionerna i dessa block.

Kanske skulle vi då kunna utöka Bitcoins skript med en instruktion för att verifiera ett bevis på att en viss
transaktion (säg en som förstörde ett mynt) skedde i sidokedjan. Bitcoin-noderna gör detta
verifiering skulle fortfarande vara fullt validerande när det gäller Bitcoins blockkedja, men de skulle göra det
gör relativt lätt SPV-verifiering av händelser i sidokedjan.

Bestrider en överföring. Det här är bättre, men fortfarande inte idealiskt. För att göra ännu förenklad verifiering, Bitcoin
noder skulle fortfarande behöva ansluta till sidokedjans peer-to-peer-nätverk (för varje kopplad
sidokedje!) och spåra alla sidokedjeblockhuvuden så att de kan avgöra den längsta
sidokedjegren. Vad vi istället vill ha är detta: när en transaktion försöker konvertera ett mynt till ett
sidkedja tillbaka till en bitcoin, den innehåller all information som Bitcoin-noder behöver för att verifiera
dess legitimitet, det vill säga att verifiera att en viss sidokedjetransaktion inträffade. Detta är uppfattningen om
ett "SPV-bevis."

Här presenterar vi ett sätt på vilket det skulle kunna fungera, med den varning som denna komponent i Sidechains är
fortfarande ett forskningsområde. För att referera till en sidokedjetransaktion i Bitcoin måste användaren tillhandahålla bevis
av (1) inkludering av sidokedjetransaktionen i ett sidokedjeblock och (2) sidokedjeblockhuvuden
visar att detta block har fått ett visst antal bekräftelser som kumulativt representerar
en viss mängd bevis på arbete. Bitcoin-noder kommer att verifiera dessa påståenden, men kommer inte att göra
några försök

verifiera att kedjan av blockrubriker som visas är den längsta. Istället kommer de att vänta på en viss definierad period, säg en dag eller två, för att tillåta andra användare att presentera bevis på att blockhuvudena presenteras i steg 2 ovan är *inte* på den längsta grenen. Om sådana bevis presenteras inom definierad period, kommer godkännandet av sidokedjetransaktionen i Bitcoin att ogiltigförklaras.

Skälet är att om ett SPV-bevis har presenterats bör det inte accepteras eftersom transaktionen inte på den längsta grenen, måste det finnas *någon* sidokedja användare som kommer att skadas av accepterandet av detta bevis. Denna användare kommer att ha incitament att presentera bevis för att ogiltigförklara bevis. Om det inte finns någon användare som kommer att skadas (kanske var det en gaffel eller omorganisation av sidan kedjan, men transaktionen i fråga fanns också i den andra filialen) så är det ingen skada i acceptera beviset.

Mer generellt försöker systemet inte vara skottsäkert mot problem i sidokedjor, och det kommer det inte hindra dig från att skjuta dig själv i foten. Om du överför din bitcoin till en sidokedja som har trasig krypto, till exempel, någon annan kanske kan stjäla ditt mynt på sidokedjan och konvertera den tillbaka till en bitcoin. Eller så kan all gruvdrift på sidokedjan kollapsa på grund av buggar, med låsta bitcoins förlorade för alltid. Men vad förslaget säkerställer är problem med sidokedjor kan inte skada Bitcoin. I synnerhet finns det inget sätt att lösa in samma mynt två gånger från en sidokedja oavsett hur buggigt det kan vara — det vill säga sidokedjor tillåter dig inte att prägla bitcoins.

Kompakta SPV-korrektur via provprover. Det finns en sista svårighet. Några av sidokedjorna kan ha en hög blockeringshastighet, kanske ett block med några sekunders mellanrum. I det här fallet, även verifiera SPV
Bevis kan vara för betungande för Bitcoin-noder. Det visar sig att vi kan använda en smart statistik teknik för att minska mängden beräkningar som krävs för att kontrollera N blockera bekräftelser från $O(N)$ till ett tal som växer mycket långsammare än linjärt.

Intuitionen är denna: när vi verifierar att ett block är begravt djupt i blockkedjan, är vi verifiera att varje block som bygger på den uppfyller målet svårigheter, det vill säga det uppfyller $hash < mål$. Nu hash-värdena för dessa block kommer likformigt fördelad i intervallet $(0, mål)$, vilket innebär att statistiskt, ca 25% av dessa block kommer i själva verket uppfyller $hash < mål / 4$. Faktum är att mängden arbete behövs för att hitta $N / 4$ block som vart och tillfredsställa $hash < mål / 4$ är densamma som den mängd arbete som behövs för att beräkna N blockerar varje uppfyller $hash < mål$. Det finns naturligtvis inget speciellt siffran 4; vi kan ersätta den med vilken faktor som helst.

Figur 10.7: en överhoppningslista för bevis på arbete. Block innehåller pekare både till föregående block och till närmaste blocket som tillfreds $hash < mål / 4$. Konceptet skulle kunna tillämpas rekursivt, med en tredje nivå av pekare till block som uppfyller $hash < mål / 16$, och så vidare.

Vad detta betyder är att om vi hade något sätt att veta vilka block i kedjan nöjda $hash < target / 4$, och kontrollerade endast de block (eller block headers), skulle vi göras, efter att ha lagt i endast en fjärdedel av verifieringsarbetet! Hur skulle vi veta vilka block tillfreds $hash < mål / 4$? De

blockera sig själva kan berätta för oss. Detta visas i figur 10.7, varje block skulle innehålla en pekare båda med sin föregångare såväl som till den senaste block som nöjda *hash <mål / 4* .

Hur långt kan vi driva detta? Kan vi välja godtyckligt stora multipler? Inte riktigt. Logiken här är liknande till pooled gruvdrift, men omvänt. Vid pooled mining verifierar pooloperatören andelar, som är block med lägre svårighetsgrad (det vill säga ett högre målvärde). Gruvarbetare hittar många fler aktier än block, så operatören måste göra extra arbete för att verifiera dem. Fördelen med att göra det är förmågan att uppskatta miner's hash power mycket mer exakt — variansen för uppskattningen är lägre.

Här ser vi den motsatta avvägningen. Eftersom vi gör mindre och mindre arbete för att uppskatta den totala mängden arbete
som har gått till att bygga kedjan kommer vår uppskattning att ha en större och större varians. Här är en exempel. Antag *N = 4* , så att utan skiplist lösningen, skulle vi kontrollera att det finns 4 block som tillfreds *hash <mål* . Den förväntade mängden arbete som en motståndare måste göra för att lura oss är fyra gånger så mycket
genomsnittlig mängd arbete som krävs för att hitta ett block.

Anta att motståndaren bara gör hälften av detta arbete. Om vi räknar, visar det sig att detta motståndaren har en chans 14% av att finna 4 block som uppfyller *hash <mål* . Å andra sidan, med en skiplistlösning med en faktor 4, skulle motståndarens uppgift vara att hitta ett enda block som uppfyller *hash <mål / 4* . I det här scenariot, den lata motståndaren som bara gör halva förväntade mängden arbete kommer att kunna lura oss med en sannolikhet på 40 % istället för 14 %.

10.7 Ethereum och smarta kontrakt

Vi har sett flera sätt att använda Bitcoins skriptspråk för att stödja intressanta applikationer, såsom en depositionerad betalningstransaktion. Vi har också sett hur Bitcoin-skriptet är något begränsat, med en liten instruktionsuppsättning som inte är Turing-komplett. Som ett resultat föreslår några nya altcoins att läggas till
applikationsspecifik funktionalitet. Namecoin var det första exemplet men många andra har föreslagit kryptovalutor liknar Bitcoin men stödjer spel, aktieemissioner, förutsägelsemarknader och
så vidare.

Tänk om vi, istället för att behöva lansera ett nytt system för att stödja varje applikation, byggde en kryptovaluta som skulle kunna stödja *alla* program vi kan drömma upp i framtiden? Detta vad Turing-fullständighet handlar om: så vitt vi vet, en Turing-komplett programmering
språk låter dig specificera alla funktioner som är möjliga att specificeras av vilken annan dator som helst. Till i viss mån, situationen idag harkens tillbaka till de tidiga dagarna av datorer själva i
1940-talet: allt mer komplicerade maskiner byggdes för olika specifika tillämpningar under Andra världskriget (som brute-force-nycklar som används av mekaniska chiffermaskiner eller fastställande av skjutning
banor för sjöartilleri), som motiverar forskare att bygga den första omprogrammerbara
allmänna datorer som kan användas för alla tänkbara tillämpningar.

Figur 10.8: En ombyggd Bombe-maskin belägen vid Bletchley Park-museet. Bomben var en specialdator designad av Alan Turing för att knäcka tyska Enigma-chiffer. Kommer Ethereum att göra det till applikationsspecifika altcoins vad den allmänna datorn gjorde med Bombe-liknande föremål?

Ethereum är ett ambitiöst altcoin som syftar till att tillhandahålla ett Turing-komplett programmeringsspråk för skriva manus eller "kontrakt". Medan det finns andra förslag för att göra detta, är Ethereum det mest anmärkningsvärt: det introducerade flera nya tekniska idéer, höll en framgångsrik crowdfunding-kampanj, höjning 20 miljoner dollar under flera månader, och antog aggressiva val för parametrar som blockeringstid.
I det här avsnittet ger vi en kort översikt av Ethereum – även om systemet är tillräckligt komplext att vi lätt skulle kunna ägna en hel andra lärobok åt det!

Smart kontraktsprogrammeringsmodell. Termen *smarta kontrakt* användes först för att beskriva användningen av
datorsystem (eller andra automatiserade medel) för att genomdriva kontrakt. Som exempel kan man tycka av en varuautomat som ett mekaniskt smart kontrakt som upprätthåller ett avtal mellan dig och maskinens ägare involverar köp av en godisbit.

I Ethereum är ett kontrakt ett program som lever på blockkedjan. Vem som helst kan skapa ett Ethereum kontrakt, mot en liten avgift, genom att ladda upp sin programkod i en speciell transaktion. Detta kontrakt är
[286]

---

skriven i bytecode och exekveras av en speciell Ethereum-specifik virtuell maskin, vanligtvis bara kallad EVM. När det har laddats upp kommer kontraktet att leva på blockkedjan. Den har sin egen balans av medel, andra
användare kan göra proceduranrop genom vilken API som helst som programmet exponerar, och kontraktet kan skicka och ta emot pengar.

Ett enkelt exempel: Namecoin i Ethereum. Vi hävdade att Ethereum kan användas för att implementera alla applikationsspecifika altcoins funktionalitet. Som ett enkelt exempel kan vi visa hur man implementerar Funktionalitet i Namecoin-stil i ett mycket enkelt Ethereum-kontrakt.

Ett exempel på implementering visas i figur 10.8. Det är kodat i Solidity, Ethereums höga nivå programmeringsspråk för att definiera kontrakt. Detta kontrakt implementerar en rå namn/värdebutik eller namnregister, där namn tilldelas värden en gång för alla. Kontraktet definierar en data variabel, registryTable, som är en mappning från 32-byte-strängar till publika nycklar. initialt kartlägger den varje sträng till nolladressen 0x0000000000...000. Detta kontrakt definierar också en enda ingångspunkt, kallad "claimName". Denna ingångspunkt accepterar ett enda argument, namn. För det första säkerställer kontraktet det den som ringer har skickat ett värde på minst 10 wei, wei är den minsta valutaenheten i Ethereum. Om otillräckliga medel har skickats, kontraktet avslutas med ett fel (det gör "kast"-satsen detta) och inga åtgärder vidtas. Om tillräckliga medel skickas och namnet ännu inte är upptaget, så är det det permanent tilldelad värdet för vilken adress som anropade denna funktion.

```
kontrakt NameRegistry {
mapping(bytes32 => adress) offentlig registertabell;
```

```
function claimName(bytes32 name) {
if (msg.value < 10) {
kasta;
}
if (registertabell[namn] == 0) {
registryTable[namn] = msg.sender;
}
}
}
```

Figur 10.8: Ett enkelt Ethereum smart kontrakt som implementerar ett namnregister.

Det är allt det här kontraktet kan göra på 8 rader kod. Men vi kan lägga till alla andra funktioner i Namecoin med lite mer jobb. Till exempel skulle vi kunna lagra mer data med varje mappning än bara adressen till den enhet som gjorde anspråk på det. Vi kan kräva att namnägare regelbundet omregistrerar sig genom att lagra en "senast uppdaterad" tid och låta andra användare göra anspråk på namn som inte har uppdaterats
på länge.

Vi kanske också vill lägga till en andra funktion så att pengarna kan tas ut. Som för närvarande programmerat, kommer pengarna bara att samlas på kontraktet för alltid, i huvudsak tas bort från omlopp. Naturligtvis, i funktionen som tillåter att pengar kan tas ut, bör vi se till att göra det

---

kontrollera att den som ringer är ägare till kontraktet. Vem som helst kan anropa vilken funktion som helst på ett Ethereum
kontrakt, men samtalen är undertecknade så att vi säkert kan identifiera vem som ringer.

Gas, incitament och säkerhet. Till skillnad från Bitcoin stöder Ethereum loopar, även om vi inte behövde dem i vårt första exempel. Det borde genast ringa varningsklockorna. Om det finns slingor kan det vara oändligt slingor. I allmänhet kan Ethereum-kontrakt löpa för evigt av olika anledningar. Ett berömt resultat i datavetenskap (stoppningsproblemets oavgörbarhet) säger att det inte finns någon algoritm som kan titta på ett programs källkod och alltid korrekt avgöra om det kommer att köras för alltid eller inte. Så hur kan vi förhindra att kontrakt löper för alltid?

Mer generellt behöver vi något sätt att begränsa kontrakt som tar lång tid att löpa, även om det är ändligt. Ethereum använder en mekanism som kallas *gas* för att uppnå detta. I huvudsak köra varje virtuell maskin instruktion kostar en liten summa pengar, så kallad gas. Olika operationer kostar olika mycket. Grundläggande operationer som tillägg eller jämförelse kostar 1 gas, medan beräkning av en SHA-3-hash (tillgänglig som
en inbyggd instruktion) kostar 20 gas och att skriva ett 256-bitars ord till beständig lagring kostar 100 gas. Varje transaktion kostar också 21 000 gas direkt. Du kan tänka på Ethereum som att flyga på en ultrarabattflygbolag: du betalar för att komma ombord och du betalar extra för allt du gör därifrån. Den kompletta listan över instruktioner tillgängliga i Ethereum och gaskostnaden för var och en är fast; skiftande dessa skulle kräva en hård gaffel precis som att ändra semantiken för Bitcoins skriptspråk.

Gas betalas för att använda Ethereums inbyggda valuta, kallad eter. Det kallas bara gas när man är van betala för kontraktsutförande. Varje transaktion kan specificera "gaspriset", det vill säga hur mycket eter det är

kommer att betala per förbrukad enhet gas. Gaspriset som erbjuds är som transaktionsavgiften i Bitcoin: gruvarbetare
är fria att publicera transaktioner med vilket gaspris som helst, och varje gruvarbetare kan självständigt bestämma sin avgift
strukturera. Detta bör resultera i ett marknadspris för gas som återspeglar utbud och efterfrågan. Från och med början av 2016,
Nätverket förblir dock experimentellt och har samlats runt en standard på 50 gigawei per
enhet gas. 50 gigawei är 5×10^{-8}
eter, eller omkring 3×10^{-10}
BTC givet eter-BTC-växelkursen i
januari 2016.

Varje samtal måste i förväg ange hur mycket gas den är villig att spendera ("gasgränsen"). Om detta värde träffas (den tar slut på bensin), körningen stannar, alla ändringar av programmets tillstånd ångras och gruvarbetaren stoppar gasen ändå. Så det är väldigt viktigt att inte få slut på bensin.

Gaskravet gör att mycket dyra beräkningar inte lämpar sig för Ethereum. De
Systemet är inte utformat för att vara en molntjänst där du ska betala andra för att göra en svår
beräkning som du inte kan göra själv. Tjänster som Amazons Elastic Compute Cloud eller
Microsofts Azure ger miljontals gånger mer valuta för pengarna. Å andra sidan är Ethereum det
lämplig för implementering av säkerhetsprotokolllogik. I huvudsak tillhandahåller den en tjänst som två (eller flera)
anonyma parter kan räkna med att bete sig enligt vad som anges.

Säkerheten för Ethereums blockkedja är inte alls lika väletablerad som Bitcoins. Teoretiskt sett
Systemet är mycket mer komplext och därför svårare att resonera om matematiskt. Praktiskt taget,
Ethereum har inte funnits särskilt länge och har inte varit föremål för samma typ av granskning som

---

Bitcoin. I synnerhet finns det farhågor om att kostnaden för transaktionsbearbetning kastar Bitcoin-stil incitamentsargument ur smällen, liknande vår diskussion om merge mining. När transaktionen
bearbetning är en icke-trivial bråkdel av en gruvarbetares totala kostnad, systemet gynnar större gruvarbetare eftersom detta
kostnaden är oberoende av hashkraft. Ännu viktigare, gasbetalningen går bara till gruvarbetaren som inkluderar initialt transaktionen i ett block. Men alla gruvarbetare som bygger på det blocket måste också validera transaktion, och de får inte betalt för att göra det. Detta innebär att de har ett incitament att hoppa över validering. Som vi såg tidigare kan detta vara farligt för blockkedjans hälsa.

Ett andra exempel: schack i Ethereum. Vi har fortfarande inte sagt så mycket om vad du kan göra med Ethereum som är nytt, så låt oss titta på ett andra exempel. Anta att Alice vill utmana Bob till en schackspel med pengar på spel. Det enda problemet är att Alice och Bob bor på olika sätt
länder och ingen av dem litar på att varandra betalar om de förlorar. Detta är ett problem som Ethereum kan lösa!

Alice kommer att skriva ett Ethereum-program som implementerar schackreglerna och ladda upp det till Ethereum.
Hon kommer att skicka kontraktet en mängd eter som motsvarar det belopp hon vill satsa. Bob kan se det här kontrakt, och om han bestämmer sig för att acceptera utmaningen kan han starta spelet genom att skicka sin egen vadslagning

insats i avtalet. Innan han gör detta bör Bob se till att kontraktet är korrekt skrivet i det det implementerar schack och kommer i slutändan att skicka allt sitt värde till den vinnande spelaren.

När båda spelarna har skickat in sin insats ska kontraktet kontrollera att insatserna är lika, förutsatt att de gör en jämn satsning. Vid det här laget är spelet igång, och det borde inte finnas något sätt för någon spelare att ta ut pengarna från kontraktet utan att faktiskt vinna spelet, eller för någon annan att ta ut pengarna under alla omständigheter.

Alice och Bob kommer att turas om att skicka transaktioner till kontraktet som indikerar nästa drag de gör gillar att leka. Kontraktet måste naturligtvis säkerställa att varje drag endast skickas in av den spelare vars tur det är att flytta, och inte av den andra spelaren eller av någon helt annan. Kom ihåg att varje transaktionen (som får kontraktet att utföra en funktion) undertecknas av den som ringer, så kontraktet kan säkerställa detta. Kontraktet kommer också att behöva kontrollera alla schackregler. Om en spelare försöker flytta en
panta tre mellanslag, som måste avvisas.

Så småningom kommer spelet att ta slut. Efter varje drag måste kontraktet kontrollera om någon av spelarna är parad, eller om
spelet är oavgjort genom dödläge eller något av de andra dragningsvillkoren i schack. Spelare borde också kunna skicka in en flytt som tyder på deras avgång. När spelet slutar kan kontraktet avsluta sig själv och skicka alla pengar till den vinnande spelaren. eller dela upp pengarna vid oavgjort.

Konceptuellt är detta en enkel tillämpning av Ethereum, men det finns finesser. Vad händer om en spelare i en att förlora position går helt enkelt iväg? Kontraktet kommer att behöva en mekanism som tilldelar pengarna till motståndare om en spelare inte har skickat ett giltigt drag under en viss tidsperiod.

Vilken spelare får flytta först? "Att spela vitt" ger en liten fördel i schack, så båda spelarna vill ha denna fördel. Detta pekar på en svårighet som många Ethereum-kontrakt står inför: det finns ingen inbyggd källa till slumpmässighet. Detta är ett svårt problem, eftersom slumptalsgeneratorn måste kunna verifieras

---

av alla gruvarbetare (så att de kan kontrollera att kontraktet utfördes korrekt) men borde inte vara det förutsägbar för någon av spelarna (eller så kanske de vägrar att gå med om de vet att de kommer att behöva spela andra).

Detta är problemet med slumpmässighetsfyrar. Som vi diskuterade i avsnitt 9.4 kan kontraktet hash värdet av nästa block i blockkedjan efter att båda spelarna har gått med. För våra specifika applikation är problemet lite lättare, eftersom bara Alice och Bob behöver övertygas om att myntet flip är slumpmässigt, inte hela världen. Så de kan använda tillvägagångssättet från avsnitt 9.3: de båda skicka in hashen för ett slumpmässigt värde, sedan avslöjar både ingångarna och härleder den slumpmässiga biten från
ingångar. Båda tillvägagångssätten har setts i praktiken.

Andra applikationer. Att spela schack kan vara kul, men den verkliga spänningen för Ethereum handlar om finansiella ansökningar. Många av applikationerna vi har diskuterat i texten hittills, inklusive förutsägelse marknader, smart egendom, spärrade betalningar, mikrobetalningskanaler och blandningstjänster kan alla vara implementeras i Ethereum. Det finns finesser i alla dessa applikationer, men de är alla möjliga och i de flesta fall är de mycket enklare att implementera än de typer av bolt-on-protokoll vi har sett med

Bitcoin. Det finns också en mängd andra applikationer, som auktioner och orderböcker, som vi inte har talat om men som folk är entusiastiska över att använda Ethereum för att implementera.

Stats- och kontosaldon i Ethereum. I kapitel 3 diskuterade vi två sätt att designa en reskontra: kontobaserade och transaktionsbaserade. I en transaktionsbaserad reskontra som Bitcoin, lagrar blockkedjan endast transaktioner (plus en liten mängd metadata i blockhuvudena). För att göra det lättare att validera transaktioner, Bitcoin behandlar mynt som oföränderliga, och transaktionsutdata måste spenderas i deras helhet, med ändringsadresser som används vid behov. I själva verket fungerar transaktioner på en global stat som är en lista över UTXO, men detta tillstånd görs aldrig explicit i Bitcoin-protokollet och är helt enkelt något gruvarbetare skapar på egen hand för att påskynda verifieringen.

Ethereum, å andra sidan, använder en kontobaserad modell. Eftersom Ethereum redan lagrar en data strukturkartläggning avtalsadresser till stat är det naturligt att även lagra kontosaldo på varje vanlig adress (även kallad ägd adress) i systemet. Det betyder att istället för att representera betalningar med hjälp av en acyklisk transaktionsgraf där varje transaktion spenderar några insatser och skapar vissa utgångar lagrar Ethereum bara ett saldo för varje adress som en traditionell bank kan lagra saldo på varje kontonummer.

Datastrukturer i Ethereum. I kapitel 3 sa vi att en kontobaserad reskontra skulle krävas snygga datastrukturer för journalföring. Ethereum har just sådana datastrukturer. Närmare bestämt varje blocket innehåller en sammanfattning av det aktuella tillståndet (saldo och transaktionsantal) för varje adress samt tillståndet (saldo och lagring) för varje kontrakt. Varje kontrakts lagringsträd mappar godtyckliga 256-bitars adresser till 256-bitars ord, vilket leder till en jättestor $2^{256} \times 256 = 2^{264}$ byte av lagring! Självklart du skulle aldrig kunna fylla upp hela denna lagring, men det är det teoretiska utrymmet. Smältningen gör det lätt att bevisa att en given adress har ett givet saldo eller lagringstillstånd. Till exempel kan Alice bevisa för Bob vad hennes balans är utan att Bob behöver skanna hela blockkedjan för att verifiera beviset.

Det enkla binära Merkle-trädet som används i Bitcoin skulle fungera för detta ändamål eftersom det tillåter effektiva bevis av inkludering (förutsatt att gruvarbetare säkerställer att inget träd kommer att innehålla två olika tillstånd för samma adress). Men vi vill också ha snabba uppslagningar och möjlighet att effektivt uppdatera en adress värde. Att göra denna Ethereum använder en något mer komplicerad trädstruktur som kallas en *Patricia träd* , även känd som en prefixträd, trie eller radixträd. Varje Ethereum-block innehåller roten av ett Merkle Patricia-träd förbinder sig till staten för varje adress, inklusive avtalsadresser. Varje kontrakts tillstånd, i sin tur, inkluderar ett träd som förbinder sig till hela tillståndet för dess lagring.

Ett annat knepigt problem med en kontobaserad reskontra är att förhindra reprisattacker. I Bitcoin, eftersom varje transaktionen förbrukar sina inmatade UTXO:er, samma signerade transaktion kan aldrig vara giltig två gånger. Med Ethereums design måste vi se till att om Alice undertecknar en transaktion som säger "betala 1 eter till Bob", Bob kan inte sända transaktionen om och om igen förrän Alices konto är tömt. För att undvika detta, varje konto i Ethereum har en transaktionsräknare som spårar hur många transaktioner det har skickat. De

uttalande Alice verkligen tecken är ”Jag godkänner min *n* th transaktion att vara en betalning på ett eter till Bob.” Detta
Transaktionen kan inte spelas upp igen eftersom Alices transaktionsräknare kommer att öka efter att den har bearbetats
och är en del av den globala staten.

För att sammanfatta, Ethereum använder kraftfullare datastrukturer än Bitcoin som en del av sin reskontra. Även om vi inte har tittat på detaljerna, tillåter det effektiva bevis för en mängd olika typer av påståenden om konton, kontrakt och transaktioner.

Ethereum-projekt. Ethereum beskrevs ursprungligen i slutet av 2013 och lanserade sin första release, dubbad Frontier, 2015. Ethereum använde en förförsäljning, vilket gjorde enheter av etervalutan allmänt tillgängliga för ett fast pris i Bitcoin, med hela intäkterna till Ethereum Foundation.

Detta är en långsammare utvecklingstakt jämfört med många altcoins, men det återspeglar detta faktum Ethereum är mycket mer komplext. Förutom EVM, en ny programmeringsmodell och ny data strukturer gjorde Ethereum också betydande förändringar i Bitcoins konsensusprotokoll. Blocktiden är inriktat på 12 sekunder istället för 10 minuter. För att minska påverkan av inaktuella block, som består av en större andel av blocken i Ethereum än i Bitcoin, Ethereum använder ett alternativt protokoll som kallas GHOST för att beräkna konsensusgrenen. Den använder också ett annat arbetsbevis. För närvarande är det en blandning
av hashfunktioner utformade för att vara minneshårda, även om Ethereum i framtiden planerar att byta till en proof-of-stake-system.

Detta representerar ytterligare en viktig avvikelse inom filosofin från Bitcoin. Ethereum-projektet är förvaltas av en ideell stiftelse och är relativt centraliserad i planering och beslutsfattande.
Det finns ett tillkännagivet schema för framtida versioner av protokollet som kommer att införa ändringar baserat på tidig Ethereum-upplevelse. Dessa kommer att vara hårda gafflar genom design, och dessutom varje Ethereum kontraktet kommer att förstöras mellan versionerna. Så Ethereum är fortfarande mycket experimentellt system med stora förändringar planerade. Från och med 2015 är det för tidigt att investera för mycket i att bygga riktigt
applikationer ovanpå Ethereum. Men det är ett mycket lovande system. Kanske framtida versioner av detta läroboken kan till och med kallas "Ethereum and Cryptocurrency Technologies."

För att avsluta det här kapitlet har vi pratat om hur en Bitcoin är en viktig del av ett mycket större ekosystem av kryptovalutor och altcoins. De tävlar, samarbetar och interagerar på olika sätt, vissa samarbetsvilliga, andra skadliga. Det är också möjligt att det i framtiden kommer att finnas tekniska sätt för transaktioner i en blockkedja för att uttryckligen hänvisa till transaktioner i en annan blockkedja.

Det återstår flera öppna frågor. Kommer altcoin-ekosystemet att konsolideras så att ett litet antal dominera, eller kommer det att förbli diversifierat? Kommer applikationsspecifika altcoins att spridas eller kommer Ethereum
modellen för en allmän plattform kommer att dominera? Kommer Bitcoin själv så småningom att bli omkörd av någon annan altcoin? Är det en bra idé att uppmuntra interaktion mellan Bitcoin och altcoins? Eller bör varje kryptovaluta vara ett separat system, till exempel genom att använda inkompatibla

gruvpussel snarare än slå samman gruvdrift? Vi kan inte svara på dessa frågor just nu, men det har vi gjort pratade om alla begrepp du behöver för att förstå och uppskatta deras betydelse.

Vidare läsning

Sidokedjorna vitt papper:

Back, Adam, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón och Pieter Wuille. Aktivera blockchain innovationer med spikad sidokedjor . 2014.

En artikel om Namecoin och alternativa sätt att designa namn/värdebutiker med kryptovalutor:

Kalodner, Harry, Miles Carlsten, Paul Ellenbogen, Joseph Bonneau, och Arvind Narayanan. En empirisk studie av Namecoin och lärdomar för decentraliserad namespace designen . I Workshop om Economics of Information Security, 2015.

Ethereums vitbok:

Olika författare. Nästa generations Smart Contract och decentraliserade Application Platform .

Ett papper som analyserar incitamentsavvikelsen i Ethereum:

Luu, Loi, Jason Teutsch, Raghav Kulkarni och Prateek Saxena. Avmystifiera incitament i konsensus dator . ACM SIGSAC-konferens om dator- och kommunikationssäkerhet, 2015.

Kapitel 11: Decentraliserade institutioner: Bitcoins framtid?

Hittills i den här boken har vi utforskat tillståndet för Bitcoin och blockkedjeteknologier från 2015. I detta kapitel kommer vi att överväga vilka framtida möjligheter som kan realiseras av Bitcoin. Vi kommer inte att hävda att vi vet
vad som kan utvecklas, efter ordspråket "Gör aldrig förutsägelser, särskilt om framtiden." Därmed frågetecknet i rubriken.

Istället kommer vi att hålla fast vid det akademiska förhållningssätt som vi har använt hittills i den här boken, även när vi studerar
potentiella framtida tekniker. Bitcoins framtid är ett ämne som verkar uppbåda entusiastiska och andfådd vision av en verklig teknisk revolution. Det här kapitlet kan vara ett manifest. Det är det inte. Vi identifiera anmärkningsvärda förslag och ta ett kliniskt tillvägagångssätt för att kategorisera dem och kritiskt utvärdera
deras relativa för- och nackdelar.

Bitcoin är ett brett ämne som omfattar själva protokollet såväl som dess potential som plattform för nya applikationer. Fokus i detta kapitel är inte framtiden för Bitcoin-protokollet, även om vi inse att det finns många frågor som kommer att forma framtiden för protokollet som är viktiga för studie, inklusive Bitcoins styrning, effektivitet, skalbarhet och funktionsuppsättning.

Istället kommer vi att fokusera på hur Bitcoins uppenbara framgång med att decentralisera valuta kan få oss att göra det
Tänk om andra centraliserade institutioner – de som hanterar aktier, obligationer, egendomsrätter och mer. Vi kommer att fråga om blockkedjeteknologi kan användas för att decentralisera dem också. Inte bara borde vi frågar om decentralisering är tekniskt möjlig, men också om det är ekonomiskt förnuftigt och fördelaktigt att samhälle.

11.1 Blockkedjan som ett fordon för decentralisering

Det fanns många misslyckade försök med digitala eller elektroniska kontanter före Bitcoin (förordet till detta boken berörde många av dem). Bitcoins viktigaste skillnad jämfört med de flesta av dessa försök är decentralisering. Kärnan i Bitcoin som möjliggör decentralisering är blockkedjan.

I det här avsnittet kommer vi att överväga hur blockkedjeteknologi kan möjliggöra decentralisering i områden annat än valuta. I det här kapitlet kommer vi att använda ett löpande exempel på en bil vars ägande styrs genom en blockkedja. Detta är ett specifikt exempel på en mer allmän idé om smart egendom som vi introducerade i kapitel 9. Smart egendom, och digitala kontrakt som styr dem, var pionjär av Nick Szabo och andra i början av 1990-talet, långt innan Bitcoin föreslogs. dock med en blockkedja kan idén konkretiseras.

Motiverande exempel. Moderna bilar använder två primära låsmekanismer: fysiska lås på dörrar och en fordonsstartspärr som elektroniskt hindrar motorn från att starta. Ägaren är försedd med en nyckelbricka som kommunicerar trådlöst med bilen för att tillåta att dörrarna låses upp

---

och motorn att starta baserat på närheten av fob till bilen och potentiellt en användaråtgärd som som att trycka på en knapp.

För att förhindra en motståndare från att förfalska bilnyckeln bör sådana upplåsningsmekanismer användas kryptografi. Medan säkerhetsforskare har hittat problem med många nyligen utplacerade låsningar protokoll är det möjligt att få det rätt. Dessa algoritmer använder vanligtvis symmetrisk nyckelkryptografi, men för vårt exempel, överväg en som använder ett digitalt signatursystem, som t.ex ECDSA, baserat på asymmetrisk kryptografi.

I det här exemplet kan bilen lagra en kopia av den/de offentliga nyckeln/nycklarna till fob/erna som är behöriga att öppna
dörrar och starta motorn. När en fob begär åtkomst skickar bilen en slumpmässig utmaning och frågar fob för att signera den med den privata nyckel som den lagrar. Om och bara om fobben kan svara med en ordentlig signatur på denna utmaning, auktoriserar bilen åtkomst. Hittills är detta inte mycket av en avvikelse från hur

låsmekanismer fungerar faktiskt, förutom att den använder tyngre krypto som skulle vara något dyrare att installera.

Bli smart. Nästa iteration av att designa en smart bil är att anta att den offentliga nyckeln som verifierar nyckelbrickan är inte hårdkodad av tillverkaren direkt. Istället har bilen den tekniska förmågan att ständigt, trådlöst ta emot nya block från en blockkedja som Bitcoins. När bilen är tillverkad, är den publika nyckeln i nyckelbrickan för sin första användare (säg en chef på monteringsfabriken) läggs till blockkedjan i en speciell transaktion, och bilen programmeras med dess transaktions-ID.

Kärnidén är att när bilen byter innehav - kan den gå från ett löpande band till kvalitet kontroll till en leveransperson till en bilhandlare till dess första ägare — uppdateringar av blockkedjan kommer godkänna varje överföring. Det är viktigt att notera att den auktoriserade nyckelbrickan inte gör det i denna modell resa med bilen. Varje person eller enhet har en redan existerande nyckelbricka (eller bär/bär teknik lämplig för att implementera funktionerna hos en nyckelbricka) med en unik signeringsnyckel som är aktiverad eller avaktiveras baserat på transaktioner som sker i blockkedjan. En sådan transaktion skulle ta bilens senaste transaktions-ID som indata och utse en ny publik nyckel som utgång. Det skulle det vara signerad med den privata nyckeln som motsvarar den nuvarande ägaren.

Detta liknar idén med smart egendom som vi diskuterade i kapitel 9, men med en viktig skillnad. Blocket kedjan transaktionen inte bara *representerar* en förändring i *ägandet* av bilen: det dessutom överför faktiska fysiska kontroll eller *innehav* av bilen. När en bil flyttas på detta sätt desto tidigare ägarens nyckelbricka slutar fungera och den nya ägarens nyckelbricka får möjlighet att öppna låsen och starta motorn. Att på detta sätt likställa ägande och besittning har djupgående konsekvenser. Det möjliggör en kraftfull sorts decentralisering, men det är inte självklart om detta är en bra idé. Vi återkommer till detta fråga i det sista avsnittet av detta kapitel.

Säkert byte. Låt oss överväga situationen där Alice äger en smart bil och vill sälja den till Bob. Möjligheten att överföra styrning digitalt öppnar för intressanta möjligheter. Till exempel kan Alice vara det resa utomlands och för att finansiera ytterligare resekostnader kanske vill sälja sin bil, vilket är fysiskt parkerade på hennes uppfart hemma. Med en internetuppkoppling kunde Bob betala Alice för bilen med

---

Bitcoin, Alice kan på distans överföra äganderätten till Bob med blockkedjan som används av bilen, och Bob kan köra iväg med sin nya bil.

Sådana transaktioner medför dock en viss risk. Om Bob skickar betalningen först, kan Alice behålla pengar och inte överföra äganderätten. Om Alice överlåter ägandet först, kanske Bob kör iväg utan betala för bilen. Även om Alice är fysiskt närvarande kan den ena parten avbryta och det kan vara svårt för en tredje part som inte var närvarande för att medla i tvisten.

Vi har stött på det här problemet flera gånger tidigare, inklusive i Coinjoin (kapitel 6) och i Namnmynt (kapitel 10). Lösningen i alla dessa fall använder samma princip. Så länge som valuta som används för betalning och bilägandet samexisterar i samma blockkedja, kan Alice och Bob bilda en enda atomär transaktion som samtidigt överför äganderätten till bilen och betalningen för bilen. Närmare bestämt skulle transaktionen specificera två indata: Alices ägande och Bobs betalning; och ange två utgångar: äganderätten till Bob och betalningen till Alice. Transaktionen

kräver att båda parter skriver under eftersom båda tillhandahåller indata. Om den ena skriver under och den andra inte gör det,
transaktionen är inte giltig. När en part väl har skrivit under kan transaktionsdetaljerna inte ändras utan ogiltigförklarande av signaturen. När den undertecknade transaktionen sänds till blockkedjan kommer bilen vänta på ett förinställt antal bekräftelser (säg 6) och tillåt sedan Bob åtkomst. Samtidigt, Bobs betalning till Alice kommer att bekräftas. Det ena kan inte hända utan det andra.

Den flitiga läsaren kanske lägger märke till ett subtilt problem. Bob kunde acceptera en transaktion undertecknad av Alice, signera
det, men faktiskt inte sänt det (ännu). Om priset på det Alice säljer förändras kan Bob då sända den gamla transaktionen till det ursprungliga priset. Mer komplicerade atomära transaktioner har varit föreslagna som inkluderar en time-out. Alice kan också helt enkelt spendera inmatningen till en ny adress som hon kontrollerar
för att ogiltigförklara den undertecknade transaktionen som hon gav till Bob som ett sätt att återkalla den.

Detta är det första av många exempel som vi kommer att se i den här föreläsningen som låter oss använda blockchain
teknik för att decentralisera en mängd olika typer av verkliga protokoll, och vi kommer att uppnå olika typer av decentralisering. Men denna idé om **atomicitet** är gemensamma för de flesta av dem, det vill säga koppla samman leveranserna från varje sida av en transaktion så att de alla sker samtidigt (eller inte alls). Atomicitet är ett viktigt säkerhetskoncept med applikationer utanför blockkedjan teknologi.

11.2 Vägar till blockkedjeintegration

Eftersom Bitcoins blockkedja har skräddarsytts för valuta kan det vara utmanande att återanvända den till representerar semantiken för andra applikationer. I Bitcoin-communityt hittar du många människor som är ganska partiska till antingen Bitcoin eller alternativa blockkedjor som en plattform för decentralisering. Vi kommer att försöka neutralisera de två alternativen i detta avsnitt.

Rutt 1: Direkt på Bitcoin

Den naturliga utgångspunkten för blockkedjeintegration är Bitcoins blockkedja. Det här är tillvägagångssättet vi användes i det föregående exemplet på en smart bil. Den största fördelen med att använda Bitcoin direkt är utplacerbarhet: koden körs, nätverket har fått betydande gruvkraft och konsensus
processen verkar ljud. Men vi kunde bara använda Bitcoin i exempelapplikationen med vissa hacks, till exempel en likvärdighet mellan krypton som används för att auktorisera Bitcoin-transaktioner och krypton som används för att öppna bildörrar. Det kommer inte alltid att vara så att sådana hack är möjlig. Mer fundamentalt, om du har något godtyckligt komplext kontrakt mellan olika
parter är det inte nödvändigtvis så att det kan representeras på ett adekvat sätt på Bitcoins blockkedja och utförs atomärt. För att illustrera farorna med att använda Bitcoins blockkedja, låt oss överväga hur vi kan implementera några naturliga tillämpningar av disintermediation.

Först överväga **Gräsrotsfinansiering tjänster** . Från och med 2015 är det största exemplet Kickstarter som matchar
entreprenörer med finansiärer genom en central webbplats. Om vi gillade idén med Kickstarter men ville
för att bygga ett helt decentraliserat alternativ skulle vi behöva förverkliga ett system där
företagare kan begära bidrag, men kan inte spendera pengarna förrän de samlar in en
förutbestämt belopp, allt utan att det finns någon mellanhand.

Figur 11.1: crowdfunding via Bitcoin. Varje bidragsgivare signerar sin egen input och output.
Transaktionen kommer att vara ogiltig tills den ackumulerade summan av indata matchar eller överstiger utdata.

Ett tillvägagångssätt för att tekniskt uppnå detta, med Bitcoin, är att instruera entreprenörer att skapa en singel
transaktion med ett godtyckligt antal ingångar (som kan variera allteftersom processen pågår) och en enda
produktion till sig själva för ett specificerat belopp, säg 1000. Sådana transaktioner kommer att cirkulera bland dem
potentiella sponsorer, där vem som helst kan bidra genom att lägga till en input till transaktionen för
mängden av sitt bidrag och digitalt signera sin egen input, såväl som den totala produktionen. En sådan
transaktionen kan inte spenderas av företagaren förrän insatserna är större än eller lika med
produktion. Detta använder några föga kända funktioner hos Bitcoin för att spendera den slutgiltiga transaktionen

---

endast dessa signaturer av begränsad form. Även om detta är möjligt på Bitcoin idag, måste vi redan göra det
gräva in i några lite kända hörn av Bitcoin. Det är inte en vanlig Bitcoin-transaktion.

Nu överväga ett andra exempel: **betala för ett bevis** . Det här exemplet kan från början verka konstigt men har
några viktiga applikationer. För att illustrera det, säg att det finns en hashfunktion H och ett allmänt känt värde
y som är skenbart ett utgångsvärde på H på någon ingångsvärdet, eller pre-bild, x . Alice hävdar att hon vet
detta värde x och Bob skulle vilja betala Alice att lära sig det också. I allmänhet, H kan istället vara vilken som helst
beräkningsbart program, och Bob skulle vilja lära sig ingångsvärden som producerar vissa utdata han är
intresserad av. I en variant av detta problem kan Bob betala för att indatavärdena publiceras offentligt
på blockkedjan.

För att på ett säkert sätt förverkliga denna transaktion måste vi säkerställa atomicitet: Alice ska bara få betalt om hon
producerar en korrekt insats och Bob måste förbinda sig att betala vid produktion av en sådan input.
Kom ihåg att vi i protokollet för atomära tvärkedjebyten i kapitel 10 visade hur man knyter en
betalning med avslöjandet av ingångsvärdet till en given hash-utgång. Ett liknande tillvägagångssätt kan användas
här.

Dessa exempel illustrerar en viktig begränsning av det direkta tillvägagångssättet att använda Bitcoins blockkedja.
I varje fall var vi tvungna att koda en komplex transaktion från den verkliga världen till Bitcoins abstraktioner.
Detta kanske inte alltid är möjligt. I exemplet med den smarta bilen antog vi bekvämt att
bil använder ECDSA-signaturer för att autentisera bilägaren. Det gjorde att vi kunde använda samma sak
offentligt/privat nyckelpar på blockkedjan och i en nyckelbricka för att låsa upp och starta bilen. I den
exempel på crowdfunding, som vi har beskrivit det, kan entreprenören bara samla in

exakt belopp de begärde, inte mer. Om bidragen överstiger det beloppet blir det överskottet
en transaktionsavgift. Slutligen, i exemplet med att betala för bevis, koppla betalningen till avslöjandet av en
värdet blir svårt om funktionen H är inte en av de hashfunktioner att Bitcoin s script stöder.

Om du inte kan – eller inte vill – få in din applikation i Bitcoins transaktionssemantik,
det finns alltid möjlighet att använda en överlagringsvaluta, vilket vi såg i kapitel 9. Detta behandlar Bitcoin
som en ren datalagring, så uttrycksfullheten i Bitcoins skript blir irrelevant. Utöver
förmåga att implementera många fler typer av applikationer, kan detta tillvägagångssätt också möjliggöra
transparens.
Tänk på bilförsäljningsexemplet igen. Om färgen på verkliga objekt (i betydelsen av färgade mynt) är
känd kan vem som helst undersöka blockkedjan för att se när en bilförsäljning har skett och hur mycket det var
betalat för det utan att nödvändigtvis känna till köparens och säljarens identitet. Detta kan vara användbart i
vissa omständigheter, och färgen kan hållas privat i situationer där den är skadlig.

Å andra sidan finns det viktiga nackdelar. Användare av en överlagringsvaluta kan inte lita på Bitcoin
gruvarbetare för att validera sina transaktioner (eftersom gruvarbetare inte förstår transaktionssemantiken i
täcka över). Detta innebär att alla användare av överlägget måste köra sina egna fullständiga noder, och SPV är inte
möjligt.
Överlagringsvalutor är också spröda om det finns buggar i implementeringar som orsakar konsensusprotokoll
att misslyckas. Om två implementeringar av en överlagringsvaluta inte är överens om huruvida en viss
transaktionen är giltig, kan den dela valutan i två, med potentiellt katastrofala konsekvenser. Förbi

---

däremot, när gruvarbetare validerar transaktioner, är det mycket mindre sannolikt att detta händer, och om det gör det, det
kommer att märkas snabbt och kommer sannolikt att lösa sig utan att det resulterar i en gaffel.

En ytterligare övervägande, oavsett om vi använder en överlagring eller inte, är frågan om belastning eller
"förorenar" Bitcoin-blockkedjan med transaktioner som ligger utanför dess ursprungliga omfattning. Det här är en
splittrande fråga i Bitcoin-gemenskapen. Vi kommer inte att välja sida, men vi kommer att påpeka att det finns ett sätt att göra det
mildra detta problem: att använda Bitcoin som en ren tidsstämplingstjänst, som vi såg i kapitel 9.1, och
inte ens som datalager. Från och med 2015 finns det begynnande tjänster som erbjuder en separat blockkedja eller
datalager, men en som är tidsstämplad via Bitcoin-blockkedjan. Det här är precis som GuardTime
tjänst från kapitel 9, men med hash som begås var tionde minut till Bitcoin-blockkedjan
istället för varje vecka i tidningen. Att använda Bitcoin för tidsstämpling kräver bara en transaktion
per block (för varje sådan tjänst eller protokoll). En nackdel är att sådana externa datalager är
Det är osannolikt att det kommer att vara lika brett replikerat och tillgängligt som Bitcoins blockkedja. Dessutom introducerar den en
grad av centralisering.

För att sammanfatta, oavsett om man använder en inbäddningsteknik eller inte, möjliggör Bitcoins blockkedja många
nya applikationer. Det kommer med fördelen av bredskalig användning, från både användare och gruvarbetare,
vilket gör det till ett säkert och lättanvändbart alternativ.

Väg 2: Alternativa blockkedjor

Den andra vägen till decentralisering är att använda en alternativ blockkedja. Även här är det några alternativ. Det mest uppenbara är att ha en separat blockkedja med sina egna regler, funktionalitet och valuta, dvs en altcoin. Ett annat alternativ är sidokedjor, som vi tittade på i kapitel 10. De viktigaste Skillnaden är att valutan som representeras av sidokedjan skulle vara kopplad på ett 1:1-sätt till Bitcoin. Sidokedjor med förbättrade skriptfunktioner skulle kunna göra det möjligt för oss att uppnå komplexa kontrakt
och möjliggöra disintermediation. Men att stödja sidokedjor kräver modifieringar av Bitcoin, och från och med 2015 har det ännu inte hänt.

Det tredje alternativet är att använda en redan existerande alternativ blockkedja som stödjer förmågan att skapa nya applikationer ovanpå det. Från och med 2015 är det mest framstående projektet som strävar efter att vara en
plattform för decentraliserade kryptovalutabaserade applikationer är Ethereum, som vi diskuterade i Kapitel 10. Begreppsmässigt är det en drömplattform för att decentralisera godtyckliga komplexa kontrakt. Men det har också några praktiska utmaningar: åtminstone från och med 2015 har det inte mognad, adoption eller gruvdrift av Bitcoin, och den har inte heller fått en jämförbar granskning. Ändå, det är ett fascinerande tankeexperiment för att decentralisera kraftfulla kontrakt, och antingen Ethereum eller en liknande system kan bli praktiskt genomförbart i framtiden.

11.3 Mall för decentralisering

Vi har sett över ett antal vägar för att uppnå decentralisering i en blockkedja. Nästa, det skulle vara användbart att upprätta en mall för hur decentralisering ser ut i termer av vad som pågår decentraliserad, vilken typ av blockkedja som är lämplig och vad exakt decentralisering innebär i villkor för enheter och säkerhet.

Nivåer av decentralisering

Decentralisering genom disintermediation. Låt oss återgå till exemplet med den smarta bilen. Till förstå det bättre, låt oss fråga, vad är den verkliga processen för denna digitala typ av ägande överföring syftar till att ersätta?

Håller med bilar som exempel på egendom, i USA, ägande bestäms av titeldokument. Detta är en centraliserad form av ägande. Titeldokumentet har bara betydelse för omfattning som Department of Motor Vehicles (DMV) erkänner det. När en bil säljs räcker det inte att fysiskt överföra detta dokument från säljaren till köparen. Överlåtelsen måste registreras i person med DMV, som kommer att uppdatera sin centrala databas. Med blockchain-överföringar flyttar vi från en statligt kontrollerad centraliserad process till en utan några mellanhänder. Det uppnår

decentralisering genom **disintermediering** .

Tvistmedling: decentralisering genom konkurrens. Antag nu att det finns en tvist om försäljning av en bil. Kanske säljaren sålde en citronbil till köparen, och köparen är missnöjd och vill för att vända transaktionen. I kapitel 3 diskuterade vi 2-av-3 multisignaturtransaktioner som kan tillåta deposition om det utöver köparen och säljaren finns en domare eller en medlare. I denna scenario kan köparen överföra bitcoins i en separat transaktion från bilen, inte direkt till säljaren, utan istället till en två-av-tre-adress som kontrolleras gemensamt av köparen, säljaren och medlaren. Medlaren kan antingen godkänna överlåtelsen eller återkalla den med hjälp av en eller den andra parten, men kan inte stjäla pengarna.

Detta är en bra början på att bygga en mekanism för tvistlösning, men det finns fortfarande många detaljer att sortera
ut. För det första förlorar vi atomiciteten i bilförsäljningen som vi förlitade oss på tidigare. För det andra är det inte klart om
bilens ägande kan återställas med pengarna. För det tredje, om bilen konverteras till en 2-av-3-adress samt, vems nyckelbricka ska vara auktoriserad att låsa upp den i detta tillstånd? Vårt syfte här är inte att reda ut dessa frågor men att använda exemplet för att noggrant överväga medlarens roll.
Låt oss specifikt jämföra denna medlingsmodell med en mer traditionell modell.

Hur skulle tvistmedling ske i den fysiska världen? Det skulle troligen gå igenom domstolen system, en centraliserad, statskontrollerad medlingsprocess som bäst navigeras med hjälp av anställd advokater. Å andra sidan, med ett digitalt avtal är parterna fria att välja vilken medlare som helst vilja. Inte längre mandat att arbeta med rättssystemet, en privat marknad för medling skulle kunna uppstå där potentiella mellanhänder kan konkurrera om upplevd rättvisa, effektivitet och kostnad. där

---

är också ett antal utmaningar. Den första är incitament: medlare kan mutas av någon av de parter i en transaktion. Den andra är att medel är låsta under tvistanmälan.
Slutligen kan deltagarna vara anonyma, vilket gör det svårt att i slutändan involvera domstolarna om intern tvistlösning misslyckas. Även om parterna är identifierade är det för närvarande inte digitala kontrakt erkänns av domstolar.

Vår poäng här är dock att detta inte är decentralisering genom disintermediation – det är vi inte helt ta bort mellanhanden. Det gör det snarare möjligt för enheter att välja vem de litar på. I andra ord är det decentralisering genom **konkurrens** . Det finns alltså ett spektrum där du på ena sidan har en enda obligatorisk mellanhand och på andra sidan tar man bort behovet av ev mellanhand överhuvudtaget — fullständig disintermediation. I mitten kan du ha flera som tävlar mellanhänder som vi nyss såg. Faktum är att vi såg detta tidigare i kapitel 9 när vi diskuterade decentraliserade förutsägelsemarknader. Istället för att en enda enhet, som InTrade, driver marknaden, deltagare är fria att välja vem de litar på bland flera konkurrerande skiljemän som utför känslig verksamhet på marknaden.

Hur säkerhet uppnås

Det finns en annan observation vi kan göra om detta exempel. Tvistmedlingens säkerhet processen är inte beroende av atomicitet. Istället kräver det att man litar på medlaren. Hur gör medlare bli pålitlig? Det kan finnas en mängd olika sätt, men ett uppenbart är rykte. Till skillnad från atomicitet, som är en teknisk säkerhetshöjande mekanism, byggs rykten upp över tiden genom i sig sociala mekanismer.

Rykte har en roll att spela i avsaknad av tekniska lösningar eller som ett komplement till dem. Det är dock inte utan nackdelar. Rykte är knutet till identiteter, och om identiteter inte är statiska eller bindande, rykte fungerar inte bra. Till exempel om en restaurang får hemska recensioner online och bestämmer sig för att stänga och öppna igen under samma ledning men ett nytt namn, är dess dåliga rykte återställa. I en anonym miljö kan rykten inte fungera alls, och i en pseudonym miljö där identiteter kan bytas enkelt, ryktebaserade system står inför betydande utmaningar. Ryktesystem kämpar också för att validera "han sa / hon sa" påståenden som påverkar ens rykte. I traditionella system som Yelp fungerar företag under sina riktiga namn, och det gör det också användare till viss del. Men i en pseudonym miljö kan det vara omöjligt att förnuftigt sortera bort falska anklagelser från fakta.

Det finns andra säkerhetsmekanismer inklusive säker hårdvara som vi inte kommer att gå närmare in på. Oavsett vilken mekanism som används, finns det i slutändan en stor säkerhetsutmaning eftersom det inte finns någon verklig verkställighet. Det finns inga straffåtgärder för dåligt uppträdande och tvister kan inte hamna i domstol, särskilt om ingen använder verkliga identiteter. Att erbjuda skulder är omöjligt, eftersom det inte finns verkställighet för att säkerställa att de kommer att återbetalas, och därför kräver transaktioner ofta insättningar, som låser upp medel för tvistperioden.

Sidofält: förtroende. Vissa människor i Bitcoin-gemenskapen använder termer som "minimering av förtroende" eller "tillitlöshet" som mål. Detta kan låta bakvänt --- vill vi inte ha system som vi kan lita på att fungera korrekt?

Ordet tillit har olika betydelser som kan orsaka denna förvirring. När Alice lånar Bob tio dollar och säger att hon litar på honom, hon menar att hon tycker att han är en pålitlig person, och att hon har förtroende för att han kommer att betala tillbaka henne. I säkerhetssammanhang är en betrodd komponent en som du är tvungen att vara beroende av. När människor använder ordet betrodd för att beskriva certifiering Myndigheter, menar de att onlinesäkerhetsgarantier skulle vara ogiltiga om sådana myndigheter misskött sig.

"Tillitsminimering" är ett värdefullt mål i den meningen att allt annat lika vill vi bygga system med färre komponenter som vi är beroende av för säkerhet. Men när du har en hammare, allt ser ut som en spik, och Bitcoin-entusiaster rycker ofta med att ta bort pålitliga komponenter från system. En pålitlig komponent är inte alltid dålig, och förekomsten av en verkliga förtroenderelationer är verkligen inte ett problem i sig. Ta bort betrodda komponenter kan också ha andra icke-uppenbara nackdelar.

Vi kommer att utveckla dessa punkter i det sista avsnittet, men för nu, efter att ha noterat komplexiteten i ordförtroende kommer vi att försöka undvika det och istället prata om säkerhet, ett mindre tvetydigt ord.

Ramverket

För att sammanfatta detta kapitel upp till denna punkt, kan vi karakterisera förslag för decentralisering på bred front
olika saker genom att ställa fyra frågor:

1. Vad är det som decentraliseras?
2. Vilken är graden av decentralisering?
3. Vilken blockkedja används?
4. Vilken säkerhetsmekanism använder den?

Med svar på dessa fyra frågor kan vi kortfattat representera nästan vilket som helst av förslagen som vi ser i Bitcoin-gemenskapen för blockkedjebaserad decentralisering. Låt oss överväga några exempel.

Exempel: Smart Property

1. Vad är det som decentraliseras?
Fastighetsägande och handel
2. Vilken är graden av decentralisering? Disintermediation
3. Vilken blockkedja används?
Bitcoins blockkedja
4. Vilken säkerhetsmekanism använder den? Atomicitet

Smart egendom, som vi har sett, decentraliserar begreppet fastighetsägande och överföringar av äganderätt. Det uppnår fullständig disintermediation - det eliminerar behovet av enheter som DMV eller staten. Vi såg hur man realiserar det med Bitcoins blockkedja, men du kan säkert använda en alternativ blockkedja. Och slutligen, den viktigaste säkerhetsprincipen som vi använde var atomicitet vid bindning tillsammans med betalningen med överlåtelsen av bilägandet.

Exempel: Decentraliserade förutsägelsemarknader

1. Vad är det som decentraliseras?
Förutsägelsemarknad
2. Vilken är graden av decentralisering? Konkurrens, disintermediation
3. Vilken blockkedja används?
Altcoin
4. Vilken säkerhetsmekanism använder den? Rykte, atomicitet

På en centraliserad förutsägelsemarknad utför den centraliserade plattformen eller börsen minst två avgörande tjänster: skilje utfallet av varje evenemang som satsas på, och sälja aktier till deltagare (eller underlätta för deltagare att säkert handla med varandra). Den decentraliserade förutsägelsemarknaden som vi såg i kapitel 9 undanröjer behovet av en central myndighet för båda dessa funktioner. Det låter vem som helst skapa en marknad för ett evenemang och vara dess domare genom att skicka en enkel transaktion,
sänkning av infartsbarriären för att utföra denna funktion. Det finns alltså fortfarande mellanhänder, men användare a
fritt att välja bland en uppsättning konkurrerande mellanhänder, och om användaren fortfarande är missnöjd kan de alltid
utföra denna funktion själva. Å andra sidan handlar användare direkt med aktier med varandra atomärt, så denna centrala myndighets funktion har disintermediated. Decentraliserat förutsägelsemarknader kräver ny funktionalitet som inte finns i själva Bitcoin, och är därför naturligt implementeras genom en anpassad altcoin med egen blockkedja.

Exempel: StorJ

1. Vad är det som decentraliseras?
Fillagring och hämtning
2. Vilken är graden av decentralisering? Konkurrens
3. Vilken blockkedja används?
Bitcoin
4. Vilken säkerhetsmekanism använder den? Rykte

StorJ är ett förslag av Greg Maxwell för fillagring och hämtning. Det har utvecklats med tiden, men vi kommer att göra det
diskutera en enkel version av det. På hög nivå distribuerar StorJ en ”agent” som bor i molnet och är programmerad att fatta vissa beslut på egen hand. Till exempel kan den hyra molnberäkning och lagring för att ge sig själv beräkningsresurser. En annan funktion som det ger användarna är möjligheten att lagra en fil under en viss tid, säg 24 timmar, i utbyte mot betalning i Bitcoin. Det kommer att hålla värd för filen så länge den fortsätter att ta emot betalning. Utöver enkel lagring kan den göra ett antal intressanta saker som vi inte kommer att överväga här. Inom vårt ramverk decentraliserar StorJ fillagring och hämtning, som är kärnan i centraliserade tjänster som Dropbox. Agenten är en mellanhand; det spelar ingen roll för våra syften att det är automatiserat. Det kan dock finnas

konkurrens mellan mellanhänder. Betalning sker med Bitcoin, men det finns ingen atomlänk mellan dem agenten som utför sina tjänster och de betalningar den får, så säkerhet är en fråga om agentens rykte.

Exempel: Zerocoin

1. Vad är det som decentraliseras?
Blandning av mynt
2. Vilken är graden av decentralisering? Disintermediation

3. Vilken blockkedja används?
Altcoin
4. Vilken säkerhetsmekanism använder den? Atomicitet

Zerocoin, som vi diskuterade i kapitel 6, är faktiskt en metod för att decentralisera blandningsmynten för att uppnå anonymitet. Istället för att använda en centraliserad blandningstjänst realiserar Zerocoin en kryptografisk
protokoll som är funktionellt likvärdigt med att använda en mix men inte använder några mellanhänder alls - bara matematik
och konsensus. Den relativt tunga kryptografi som behövs i Zerocoin (och dess efterträdare, Zerocash) innebär att en separat blockkedja är den mycket mer genomförbara vägen. När det gäller säkerhetsmekanismen, minns
att tanken att bränna ett basmynt och få ett nollmynt i utbyte mot det är atomärt kopplade
genom samma transaktion; och på liknande sätt för att senare lösa in ett nollmynt. Detta är ett exempel på atomicitet.

11.4 När är decentralisering en bra idé?

I detta kapitel har vi hittills fokuserat på de tekniska utmaningarna för att uppnå decentralisering. Nu vi ska fördjupa oss i frågor om motivation. Dessa frågor är icke-tekniska men ofta är de det lika svårt att svara på: Är decentralisering en bra idé? Är det ekonomiskt genomförbart? Vad är sociala konsekvenser av decentralisering?

Hittills har vi använt begreppet decentralisering som ett tekniskt begrepp utan att vara explicit om det faktum att den är politiskt laddad. När vi pratar om att ersätta traditionella system helt eller delvis med tekniska alternativ talar vi egentligen om att omfördela makt från väletablerade juridiska, sociala och finansiella institutioner. Tanken på decentralisering härrör alltså från Bitcoins rötter i cypherpunk-rörelsen — en rörelse som påbörjades av nonkonformister som drömmer om kryptografi förmåga att ge individuell autonomi. Med blockkedjan verkar detta ideal närmare än någonsin. Men är detta ideal genomförbart eller önskvärt?

För att återgå till vårt löpande exempel, det finns två problem som de traditionella institutionerna försöker lösa för bilägare. Den första är att upprätthålla ägande, eller i huvudsak förhindra stöld. Den andra är säkerställa säkra byten, eller förhindra att någon blir lurad under en försäljning. Så att analysera hur smart fastigheter står sig jämfört med det befintliga systemet måste vi titta på inte bara hur effektivt saker är när allt går rätt, men också, avgörande, hur illa saker kan bli när något
går fel.

Utmaningen med säkerhet i den verkliga världen

Att försvara sig mot någon form av stöld – bilar, konst, pengar etc. – är en övning för att förebygga, upptäcka och korrigering. Förebyggande säkerhetsmekanismer försöker stoppa stöld innan det händer, samtidigt som de upptäcks
mekanismer säkerställer att stöld uppfattas så att potentiella korrigerande åtgärder kan vidtas för att återställa

skadestånd för stölden och för att straffa gärningsmannen (vilket också kan fungera som ett avskräckande begår stöld). Billås och larm är förebyggande mekanismer, medan GPS-spårningsenheter (t.ex LoJack) kan hjälpa till att upptäcka stölden och göra det möjligt för polisen att återställa den stulna bilen. De nyckelinsikten är att billåset bara är en liten del av avskräckning från bilstöld – en del av en stor, invecklat system som involverar polis, försäkringsbolag, domstolar, etc. Om du bodde i en laglös miljö, skulle ett billås i sig inte vara mycket avskräckande mot stöld. Lämna din bil låst gatan skulle se till att den snabbt skulle bli stulen.

Modellen vi har sett för smart egendom är starkt beroende av förebyggande mekanismer. Vi kunde att uppnå decentralisering endast för att vi likställde innehav med ägande — att äga en bil är i huvudsak likvärdigt med att känna till den privata nyckeln som motsvarar en angiven transaktion på en blockkedja. Men denna kontrollmekanism är en dålig ersättning för vår nuvarande mosaik av institutionella stöd, som vi ska förklara.

Om vi minskar ägandet till problemet med att säkra privata nycklar, ökar det insatserna för digital säkerhet, vilket är ett svårt problem med att människan är en svag länk. Programmerare har försökt skriva buggfri kod i årtionden, men utmaningen förblir svårfångad. Designers av kryptosystem har försökt i decennier för att få icke-tekniska användare att använda och hantera privata nycklar på ett sätt som motstår både stöld och oavsiktlig förlust av nycklar, även med små framsteg. Om modellen för decentralisering förlitar sig överdrivet på privata nycklar kan bilar bli stulna av skadlig programvara eller i nätfiskeattacker, och förlust av en nyckel kan förvandla din bil till en gigantisk tegelsten. Även om det kan finnas reservmekanismer för att täcka dessa
typer av händelser, oundvikligen tenderar sådana mekanismer att leda oss tillbaka mot mellanhänder och centraliserade system, spånade fördelarna med den decentraliserade modellen som vi strävade efter för.

Ett annat område av fastighetsöverlåtelser som i grunden är mänskligt orienterat är att hantera tvister som kan uppstå om försäljningsvillkoren eller andra aspekter av överföringen. Om den verkliga världen, om deltagarna inte kan nå en lösning kommer frågan att hamna i domstol där en domare metodiskt granskar var och en lite bevis, vittnesmål och skrivna ord för att nå ett nyanserat beslut om giltigheten av försäljningen. Det är frestande, särskilt för tekniker, att tänka på lagen som en uppsättning logiska regler eller algoritmer som kan ge ett tydligt beslut. Men verkligheten i rättssystemet är att inte bara är lagar och kontrakt mångsidiga, är de i slutändan föremål för mänsklig tolkning och gottfinnande, vilket är längre bort från begreppet tydliga logiska regler. Detta är ingen svaghet. Det tillåter att lösa situationer som är mycket mer komplexa än vad individen förutsåg lagen.

För att driva hem obalansen mellan säkerhetsegenskaperna får vi från den decentraliserade modellen och de säkerhetsegenskaper som vi faktiskt vill ha, låt oss återgå till det tidigare exemplet med decentraliserad folkmassafinansiering. Vi såg en teknisk mekanism för att säkerställa att en entreprenör inte kan ta ut pengar på investeringar tills bidragen uppgår till något förutbestämt belopp. Detta är dock inte på något sätt förhindrar en entreprenör som framgångsrikt har samlat in pengarna från att fly med pengarna! I Faktum är att även med den nuvarande centraliserade modellen har det förekommit många påstådda bedrägerier webbplatser för crowdfunding, vilket resulterade i flera stämningar. I en modell där entreprenörer är potentiellt anonym och det finns ingen avskräckande effekt från hotet att bli stämd, kommer detta problem sannolikt att vara

mycket värre. Det är svårt att föreställa sig att det skulle kunna finnas en teknisk lösning på detta problem. Det här är en annan
fall där tekniken bara löser en liten del av problemet, och ärligt talat, inte ens
intressant del av problemet.

För att sammanfatta, de intressanta problemen med smart egendom verkar vara sociala problem, frågor som uppstår
när något går fel. Teknik kan säkerställa en mycket effektiv transaktion när alla parter är det
nöjd, men den är inte väl positionerad för att lösa svåra tvister.

För- och nackdelar med smart egendom

Som hävdats har smart egendom svårt att decentralisera aspekterna av ett system som traditionellt
kräver mänskligt ingripande. Faktum är att automatisering till och med gör det svårare genom att inte komponera
väl med medling och andra processer om de läggs på i lager i efterhand. Slutligen kan det skapa
nya kategorier av problem, som att kräva mjukvarusäkerhet utöver fysisk säkerhet i
fallet med en bil.

Dessa exempel är till viss del tecknade versioner av vad ett genomarbetat förslag på smart
egendom kan se ut. Många förslag i Bitcoin-gemenskapen är mer nyanserade, men även i vår
enkel inställning, vi kan urskilja fördelarna och nackdelarna med smart egendom.

Den största fördelen med smart egendom är effektiviteten av ägandeöverföring, som kan göras från
var som helst när som helst. För försäljning av föremål som är mindre värda än en bil, kanske en smartphone eller dator,
det är osannolikt att tvister hamnar i domstol, så inget går förlorat i det avseendet. För sådana föremål, atomic
transaktioner är en användbar säkerhetsfunktion.

Smart egendom genom blockkedjor ger också större integritet och till och med anonymitet. Medan vi har
hävdade att det komplicerar tvistlösning, integritet är också fördelaktigt i ett samhälle där konsumenten
data används av företag på sätt som är osynliga och sannolikt oavsiktliga genom att göra köpen. I
vissa fall kan det vara viktigt för parterna i en transaktion att inte avslöja sin identitet,
vilket inte är genomförbart i en centraliserad mellanmodell.

Slutligen tillåter den decentraliserade modellen att välja medlare. Även om vi nöjer oss med det juridiska
system, ofta förmedlas tvister av privata företag som Visa eller PayPal bakom stängda dörrar
med en metod som är svår att granska. Genom att använda en alternativ modell där sådan medling är

öppnat upp för konkurrens kan vi potentiellt ge mer transparens och offentlig tillsyn till
bearbeta.

Krypto, staten och den stora möjligheten

Det finns en slående parallell mellan framväxten av den moderna staten och målen för
teknik som vi har diskuterat i detta kapitel. I att skala upp samhället från stammar och små grupper,

regeringar har varit tvungna att konfrontera just problemet med att möjliggöra säker handel och annat interaktioner mellan främlingar. Metoderna kan vara väldigt olika men målet är ett gemensamt.

Även om en maximalistisk vision för decentralisering kan innebära att avveckla staten, så är det inte riktigt livskraftig vision, särskilt när andra som delar vår demokrati vill ha en. Däremot decentralisering genom teknik är inte nödvändigt i opposition till staten alls. Faktum är att de kan vara ömsesidigt välgörande. Till exempel, om man antar välidentifierade parter, kan överföringar av smart egendom använda blockkedja för effektiva överföringar och fortfarande använda domstolssystemet om det uppstår en tvist. Vi tänker det stora
möjligheten för blockchain-teknologi är att implementera decentralisering på ett sätt som kompletterar statens funktioner, snarare än att försöka ersätta dem.

Det är frestande att tro att saker och ting kommer att bli decentraliserade bara för att tekniken finns. Men i praxis måste det finnas ett övertygande ekonomiskt skäl, såsom statlig reglering det vill säga särskilt betungande eller ineffektivt, eller en maktobalans som kan leda till missbruk. Som en illustration av detta har människor i olika afrikanska länder antagit mobiltelefonminuter som en ad-hoc-valuta som ligger utanför statens kontroll och mindre föremål för maktmissbruk.

För att sammanfatta, har vi visat den tekniska planen för decentralisering i detta kapitel, och även kritiskt granskat motiven bakom decentralisering. Vi uppmuntrar dig att leta efter övertygande använda fall av decentralisering, särskilt sådana som integreras i befintliga lagar och regler praxis.

Avslutning på boken

Vissa människor är entusiastiska över Bitcoin på grund av den underliggande tekniken. Andra är exalterade om dess kommersiella möjligheter, och ytterligare andra om dess sociala och politiska implikationer. Förnuftiga människor kan vara oense om de två sistnämnda, men vi hoppas att den här boken har övertygat dig om det
tekniskt sett är Bitcoin djup, ny, intressant och baserad på sunda principer. Bortom Bitcoin det finns en fascinerande värld av alternativa kryptovalutadesigner som vi precis har börjat utforska, varav några en dag kan bli viktigare än själva Bitcoin.

Vi kom in på Bitcoin för att vi tror på kraften i dess teknik, och vi tror att det är djupt kopplat till resten av datavetenskap. Medan vi har lyft fram hur till synes fantastiskt nytt teknik kan kämpa för att tränga undan etablerade institutioner, vi tror att i det långa loppet, människor kommer att fortsätta hitta nya kommersiellt och socialt användbara saker att göra med kryptovaluta teknologi. Även om ditt intresse främst är kommersiellt, gör du klokt i att behärska det underliggande teknik — att förstå dess kraft och begränsningar hjälper dig att bättre klara marknadens hype cykler.

Vi avslutar med några ord om vart vi ska gå härifrån. En av de bästa sakerna med decentralisering är att det är en bra plattform för experiment och lärande. Vem som helst kan ladda ner och analysera Bitcoins blockkedja, eller bygga sina egna applikationer ovanpå den; vi hoppas att du tar dra nytta av dessa möjligheter.

Vi har skapat ett antal onlinematerial som kompletterar denna text. Vår Coursera kurs innehåller videoföreläsningar som speglar innehållet i den här boken. Den har också frågesporter och en serie programmering
uppgifter. Genom att gå kursen får du också tillgång till forumen där du hittar en community av likasinnade elever.

Även om det första utkastet av denna bok är färdigt, är det ett arbete som pågår. Vi följer utvecklingen områden som Ethereum, och närhelst en mängd vetenskaplig kunskap utvecklas kring ett nytt område, vi kommer att släppa ytterligare kapitel. Kolla vår kurs hemsida !

Om Författarna

Arvind Narayanan (Ph.D. 2009) är en biträdande professor i datavetenskap vid Princeton. Narayanan leder Princeton Web Transparency and Accountability-projektet som syftar till att avslöja hur företag samlar in och använder vår personliga information. Han leder också en forskargrupp som studerar säkerheten, anonymiteten och stabiliteten för Bitcoin och kryptovalutor. Det visade hans doktorsforskning att anonymisering av data bryts på grundläggande sätt, vilket han gemensamt fick 2008 års Privacy Enhancing Technologies Award. Du kan följa honom på Twitter på @random_walker .

Joseph Bonneau är en teknik Fellow vid Electronic Frontier Foundation och forskar Forskare vid Stanford. Förutom att forska om Bitcoin och kryptovalutor har han arbetat med lösenord och webbautentisering, säkra meddelandeverktyg och HTTPS för säker webbsurfning. han fick en doktorsexamen från University of Cambridge under ledning av Ross Anderson och en MS från Stanford under överinseende av Dan Boneh. Tidigare var han postdoktor vid CITP, Princeton och han har tidigare arbetat på Google, Yahoo och Cryptography Research Inc.

Edward W. Felten är professor i datavetenskap och Public Affairs vid Princeton och grundande direktör för Centrum för informationsteknologipolitik. 2011–12 fungerade han som den första Chefsteknolog vid US Federal Trade Commission. Hans forskningsintressen inkluderar dator säkerhet och integritet, och tekniklagstiftning och policy. Han har publicerat mer än 100 artiklar i forskningslitteratur och två böcker. Hans forskning om ämnen som Internetsäkerhet, integritet, upphovsrätt och kopieringsskydd, och elektronisk röstning har behandlats flitigt i populärpressen.

Andrew Miller är en datavetenskap doktorand vid University of Maryland, och tidigare fick sin MS-examen från University of Central Florida. Han har studerat kryptovalutor sedan dess 2011, och har skrivit vetenskapliga artiklar om ett brett utbud av originalforskning, inklusive ny proof-of-work pusselkonstruktioner, programmeringsspråk för blockkedjedatastrukturer och peer-to-peer nätverksmätnings- och simuleringstekniker. Han är biträdande direktör för Initiativ för kryptovalutor och kontrakt (IC3) på Cornell och en rådgivare till zcash-projektet.

Steven Goldfeder är doktorand vid Institutionen för datavetenskap vid Princeton University, rådgiven av Arvind Narayanan. Han är medlem i Security & Privacy Research Group, en CITP Graduate Student Fellow och en National Science Foundation Graduate Research Fellow. Hans forskning intressen inkluderar kryptografi, säkerhet och integritet, särskilt decentraliserade digitala valutor. Hans nuvarande arbete innebär att öka säkerheten för Bitcoin-plånböcker.

Jeremy Clark är biträdande professor vid Concordia Institute for Information Systems Engineering i Montreal. Han tog sin doktorsexamen från University of Waterloo 2011, dit han sökte kryptografi för att designa och distribuera verifierbara röstsystem, inklusive Scantegrity - det första användning av ett verifierbart system i ett offentligt val. Han blev intresserad av Bitcoin i 2010 och publicerade en av de första akademiska artiklarna i området. Utöver forskning har han arbetat med flera kommuner om röstningsteknik och vittnade för den kanadensiska senaten om Bitcoin.