Bitcoin: Sähköinen käteisjärjestelmä vertaisverkossa

Satoshi Nakamoto
https://bitcoin.org/en/bitcoin-paper

Suomentanut:
@Biocycle, Biocycle@protonmail.com
Yhteistyössä: @LohkoKettu
Aleksi Suomalainen, @locusf
Antti Majakivi, @Anduck
Niko Laamanen, @n1kofi

Tiivistelmä

Täysin vertaisverkossa toimiva sähköinen käteinen mahdollistaa verkossa tehtävät maksut suoraan osapuolelta toiselle niiden liikkumatta rahoituslaitoksen kautta. Digitaaliset allekirjoitukset ovat osa ratkaisua, mutta tärkeimmät niiden suomat edut menetetään, jos luotettua kolmatta osapuolta yhä tarvitaan estämään kaksinkertainen kulutus. Esitämme kaksinkertaisen kulutuksen ongelmaan ratkaisuksi vertaisverkkoa. Verkko aikaleimaa tapahtumat tiivistelaskemalla ne osaksi jatkuvaa tiivisteisiin perustuvaa työntodisteketjua. Tätä kirjanpitoa ei voi muuttaa laskematta työntodisteita uudelleen. Pisin ketju toimii paitsi todisteena tapahtumien järjestyksestä, myös todisteena siitä, että ketjuun on kulutettu yhteensä eniten laskentatehoa. Niin kauan kuin suurinta osaa laskentatehosta säätelevät solmut, jotka eivät tee yhteistyötä keskenään hyökätäkseen verkkoa vastaan, ne pystyvät luomaan pisimmän ketjun jättäen hyökkääjät jälkeensä. Itse verkko ei vaadi kuin minimaalisen rakenteen. Verkko lähettää viestejä parhaansa mukaan ("best effort"), ja solmut voivat vapaasti poistua verkosta ja liittyä siihen uudelleen. Uudelleen liittyessään ne hyväksyvät pisimmän työntodisteketjun todisteeksi siitä, mitä tapahtui niiden poissa ollessa.

1 Johdanto

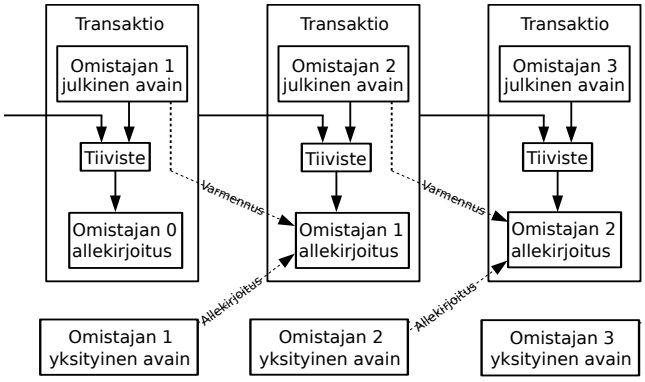
Internetissä tapahtuva kaupankäynti on sähköisten maksujen käsittelyn osalta päätynyt nojaamaan lähes yksinomaan rahoituslaitoksiin, jotka toimivat luotettuina kolmansina osapuolina. Vaikka järjestelmä toimii riittävän hyvin useimmissa transaktioissa, se kärsii silti luottamuspohjaisen mallin luontaisista heikkouksista. Täysin peruuttamattomat transaktiot eivät ole todellisuudessa mahdollisia, sillä rahoituslaitokset eivät voi välttyä joutumasta sovittelemaan riitatilanteita. Sovittelun kustannukset nostavat transaktioiden kustannuksia, mikä

rajoittaa pienimmän mahdollisen käytännöllisen siirtotapahtuman kokoa ja poistaa pienten satunnaisten siirtojen mahdollisuuden, ja ne kustannukset, jotka aiheutuvat siitä, että peruuttamattomat maksut peruuttamattomista palveluista eivät ole mahdollisia, ulottuvat vieläkin laajemmalle. Kaupan perumisen mahdollisuuden myötä luottamuksen tarve lisääntyy. Kauppiaiden on oltava varovaisia asiakkaidensa suhteen ja hankittava heiltä enemmän tietoja kuin olisi muuten tarvetta. Petoksista tietty prosenttiosuus hyväksytään väistämättöminä. Kahdenkeskisessä kaupankäynnissä nämä kustannukset ja maksuihin liittyvät epävarmuudet voidaan välttää käyttämällä fyysistä valuuttaa, mutta maksujen suorittamiseen jonkin viestintäkanavan välityksellä ei ole olemassa mekanismia ilman luotettua kolmatta osapuolta.

Tarvitaan siis sähköinen maksujärjestelmä, joka perustuu luottamuksen sijaan kryptografiseen todisteeseen, jolloin kahden halukkaan osapuolen on mahdollista tehdä kauppaa suoraan keskenään ilman, että tarvitaan luotettua kolmatta osapuolta. Transaktiot, joiden peruuttaminen on laskennallisesti epäkäytännöllistä, suojaisivat myyjiä petoksilta, ja helposti toteutettavat rutiininomaiset escrow-mekanismit puolestaan suojaisivat ostajia. Tässä artikkelissa ehdotamme ratkaisuksi kaksinkertaisen kulutuksen ongelmaan vertaisverkossa toimivaa hajautettua aikaleimapalvelinta, joka tuottaa laskennallisia todisteita siirtotapahtumien kronologisesta järjestyksestä. Järjestelmä on turvallinen niin kauan kuin vilpittömillä solmuilla on hallussaan yhteensä enemmän laskentatehoa kuin yhdelläkään yhteistyössä toimivien hyökkääjäsolmujen ryhmällä.

2 Transaktiot

Määrittelemme sähköisen kolikon digitaalisten allekirjoitusten ketjuksi. Kukin omistaja siirtää kolikon aina seuraavalle omistajalle allekirjoittamalla digitaalisesti sekä edellisen siirron tiivisteen että seuraavan omistajan julkisen avaimen ja lisäämällä nämä kolikon loppuun. Maksunsaaja voi varmentaa allekirjoitukset varmentaakseen omistusketjun.

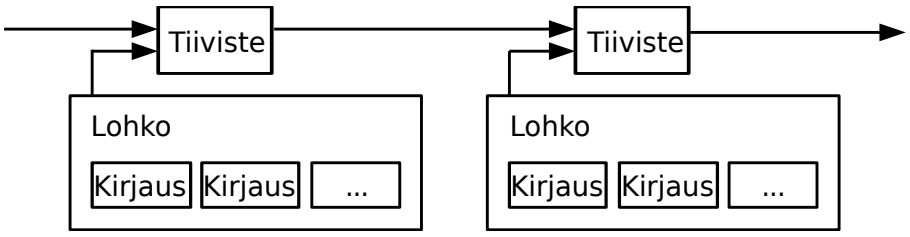


Ongelmaksi muodostuu tietenkin se, ettei maksunsaaja voi mitenkään tarkistaa, onko joku omistajista kuluttanut kolikon kahteen kertaan. Tavallinen ratkaisu on ottaa mukaan luotettu keskitetty auktoriteetti, toisin sanoen rahapaja, joka tarkistaa jokaisen transaktion kaksinkertaisen kulutuksen varalta. Jokaisen tapahtuman jälkeen kolikko on palautettava rahapajaan, joka laskee liikkeelle uuden kolikon, ja vain suoraan rahapajasta liikkeelle laskettujen kolikoiden voi luottaa olevan sellaisia, joita ei ole käytetty kahteen kertaan. Kuten pankkitoiminnassa, tämän ratkaisun ongelmana on koko rahajärjestelmän riippuvaisuus sitä ylläpitävästä yrityksestä, jonka kautta jokaisen siirron on kuljettava.

Tarvitsemme keinon, jolla maksunsaaja voi selvittää, ovatko edelliset omistajat allekirjoittaneet aiempia transaktioita. Tavoitteemme huomioon ottaen vain varhaisimmalla siirrolla on merkitystä, joten emme välitä myöhemmistä kaksinkertaisen kulutuksen yrityksistä. Vain olemalla tietoinen kaikista siirtotapahtumista voi vahvistaa tietyn siirron puuttumisen. Rahapajapohjaisessa mallissa rahapaja on tietoinen kaikista siirroista ja päättää, mikä niistä on saapunut ensimmäisenä perille. Jotta tämä voidaan suorittaa ilman luotettua kolmatta osapuolta, transaktioiden täytyy olla julkisia [1] ja tarvitaan menetelmä, jonka avulla osallistujat pääsevät sopuun yhdestä ja samasta siirtojen vastaanottamisen järjestyksen historiasta. Maksunsaaja tarvitsee todisteen siitä, että kunkin transaktion tapahtumahetkellä kyseinen transaktio oli solmujen enemmistön mielestä vastaanotettu ensimmäisenä.

3 Aikaleimapalvelin

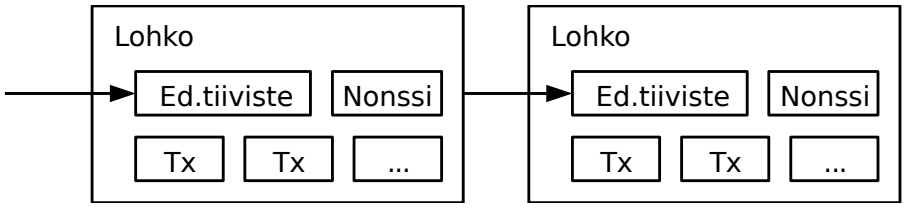
Ehdottamamme ratkaisu alkaa aikaleimapalvelimesta. Aikaleimapalvelin toimii siten, että palvelimelle syötetään erinäisiä kirjauksia sisältävien lohkojen tiivisteitä, jotka palvelin aikaleimaa ja julkaisee laajasti – niin kuin sanomalehti tai Usenet-viesti [2–5]. Aikaleima todistaa, että päätyäkseen mukaan tiivisteeseen tietojen on luonnollisesti täytynyt kyseisellä hetkellä olla olemassa. Jokaisen aikaleiman tiivisteeseen sisältyy aina edellinen aikaleima, mistä muodostuu ketju, jossa jokainen uusi aikaleima vahvistaa sitä edeltäneitä aikaleimoja.



4 Työntodiste

Hajautetun aikaleimapalvelimen toteuttamiseksi vertaisverkossa vaaditaan työntodistejärjestelmä, joka muistuttaa Adam Backin Hashcash-järjestelmää [6] sanomalehden tai Usenet-viestien sijaan. Työntodistelaskennassa etsitään sellaista arvoa, jonka tiiviste esimerkiksi SHA-256-algoritmilla tiivistelaskettuna alkaa joukolla nollabittejä. Keskimääräisesti tämän arvon etsiminen vaatii vaadittujen nollabittien määrään nähden eksponentiaalisesti työtä, ja tehty työ voidaan varmentaa yhdellä tiivistelaskelmalla.

Tässä aikaleimaverkossa työntodistejärjestelmä toteutetaan lisäämällä lohkoon nonsseja eli kertakäyttöisiä arvoja niin kauan, kunnes löydetään sellainen arvo, joka tuottaa lohkon tiivisteeseen vaaditun määrän nollabittejä. Kun työntodisteen vaatimukset täyttävä määrä laskentatehoa on kerran kulutettu, ei lohkoa voi muuttaa laskematta työntodistetta uudelleen. Koska lohkot on ketjutettu toisiinsa, yhtäkään lohkoa ei voi muuttaa laskematta myös kaikkia sen jälkeen tulevia lohkoja uudelleen.



Työntodisteella ratkaistaan myös edustusoikeuden määräytymiseen liittyvät ongelmat enemmistöpäätöksiä tehtäessä. Jos IP-osoite määrittäisi äänioikeuden, äänestystuloksia voisi vääristää varaamalla itselleen useita IP-osoitteita. Työntodistejärjestelmässä käytännössä jokaisella prosessorilla on yksi ääni. Tällöin pisin ketju, jonka työntodisteisiin on sijoitettu eniten vaivaa, edustaa enemmistöpäätöstä. Jos vilpittömät solmut hallitsevat suurinta osaa laskentatehosta, vilpitön ketju kasvaa nopeimmin ja päihittää kilpailevat ketjut. Muokatakseen jotakin aiemmista lohkoista hyökkääjän täytyisi laskea paitsi kyseisen lohkon myös kaikkien sen jälkeen tulleiden lohkojen työntodisteet uudelleen ja sen jälkeen vielä saada vilpittömien solmujen tekemä työmäärä kiinni ja ylittää se. Osoitamme myöhemmin, miten todennäköisyys sille, että hitaampi hyökkääjä saisi vilpittömän ketjun kiinni, pienenee eksponentiaalisesti jokaisen lisätyn lohkon myötä.

Aikojen saatossa laitteistojen nopeudet kasvavat ja kiinnostus solmujen ajamista kohtaan vaihtelee. Tätä kompensoidaan säätämällä työntodisteen laskemisen keskimääräistä vaikeutta aina sen mukaan, kuinka monta lohkoa tunnissa keskimäärin syntyy. Jos niitä syntyy liian nopeasti, vaikeus kasvaa.

5 Verkko

Verkon toiminta voidaan jakaa seuraaviin vaiheisiin:

1. Uudet tapahtumat lähetetään kaikille solmuille.
2. Jokainen solmu kerää uusia tapahtumia lohkoon.
3. Jokainen solmu pyrkii löytämään vaikean työntodisteen lohkolleen.
4. Kun solmu löytää työntodisteen, se lähettää lohkonsa kaikille solmuille.
5. Solmut hyväksyvät lohkon vain, jos kaikki sen sisältämät siirrot ovat kelvollisia eikä niitä ole jo kertaalleen kulutettu.
6. Solmut ilmaisevat hyväksyvänsä lohkon alkamalla työstää ketjun seuraavaa lohkoa hyväksytyn lohkon tiivisteen päälle.

Solmut pitävät pisintä ketjua aina oikeana ja jatkavat sen työstämistä pidemmäksi. Jos kaksi solmua lähettää samaan aikaan eri version seuraavasta lohkosta, osa solmuista saattaa vastaanottaa ne eri järjestyksessä. Siinä tapauksessa ne työstävät sitä haaraa, jonka vastaanottivat ensimmäisenä, mutta tallentavat toisen haaran siltä varalta, että siitä tulee pidempi. Tasapeli ratkeaa, kun seuraava työntodiste löytyy ja toisesta haarasta tulee pidempi. Toista haaraa työstäneet solmut vaihtavat tällöin pidemmän haaran pariin.

Uusien transaktioiden lähetysten ei tarvitse välttämättä tavoittaa kaikkia solmuja. Kunhan ne tavoittavat monia solmuja, ne ennen pitkää päätyvät johonkin lohkoon. Lohkojen lähetyksessä siedetään myös pudonneita viestejä. Jos jokin solmuista ei vastaanota tiettyä lohkoa, solmu pyytää sitä seuraavan lohkon vastaanottamisen yhteydessä havaitessaan jääneensä yhtä lohkoa vaille.

6 Kannustin

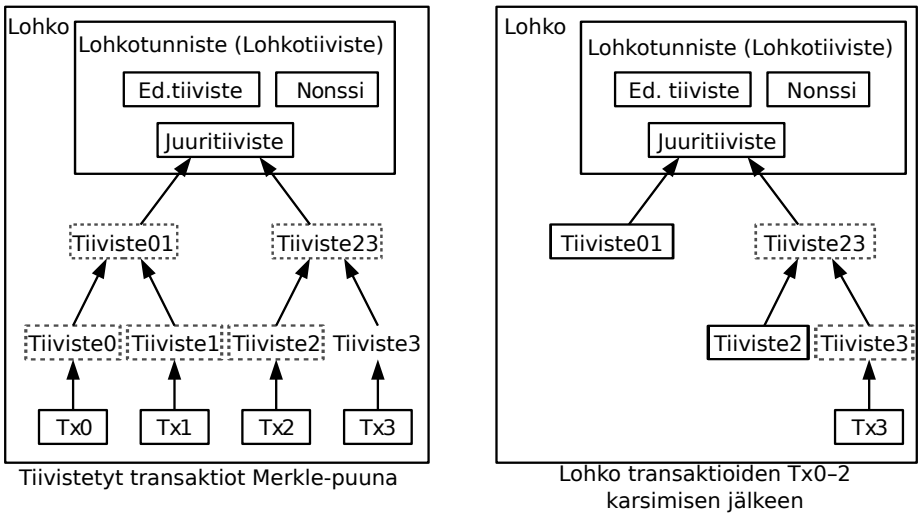
Verkossa on käytäntönä pitää aina lohkon ensimmäistä siirtotapahtumaa erityisenä: se luo uuden kolikon, jonka omistaa lohkon luoja. Tämä antaa solmuille lisäkannustimen tukea verkkoa ja toimii keinona laskea kolikoita liikkeelle ilman keskitettyä liikkeelle laskevaa viranomaista. Uusia kolikoita tulee tasaisesti lisää vakiomäärä samaan tapaan kuin kultaa tulee kiertoon lisää kullankaivajien kuluttaessa resurssejaan. Meidän tapauksessamme kulutetaan laskenta-aikaa ja sähköä.

Kannustin voidaan rahoittaa myös siirtomaksuilla. Mikäli siirtotapahtuman tulosarvo on pienempi kuin sen syötearvo, niiden välinen erotus on siirtomaksu, joka lisätään sen lohkon kannustinarvoon, johon kyseinen siirtotapahtuma sisältyy. Heti kun ennalta määritelty määrä kolikoita on tullut kiertoon, kannustin voi kokonaisuudessaan muodostua siirtomaksuista ja olla täysin inflaatiovapaa.

Kannustin voi osaltaan auttaa pitämään solmut vilpittöminä. Jos ahne hyökkääjä kykenee kokoamaan enemmän laskentatehoa kuin kaikki vilpittömät solmut yhteensä, hänen täytyisi valita sen välillä, huijaako ihmisiä anastaakseen tekemänsä maksut takaisin itselleen vai tuottaako uusia kolikoita. Hänen luulisi pitävän kannattavampana vaihtoehtona noudattaa sellaisia sääntöjä, jotka suosivat häntä ja suovat hänelle enemmän uusia kolikoita kuin muille yhteensä, kuin sabotoida koko järjestelmää ja heikentää samalla varallisuutensa arvoa.

7 Levytilan uusiokäyttö

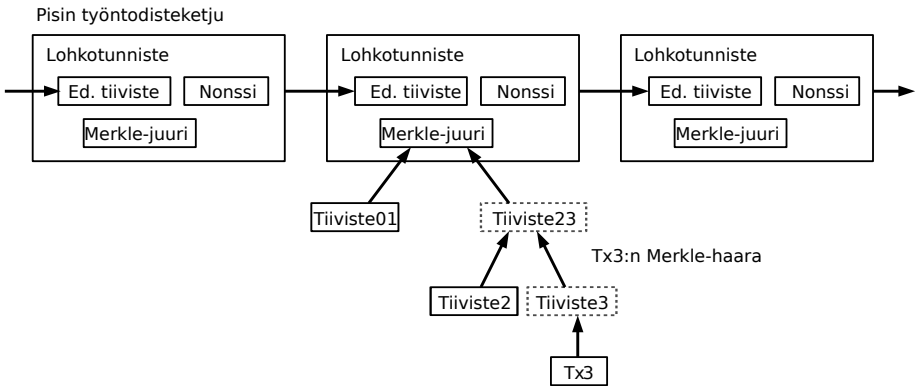
Kun kolikon viimeisin transaktio on hautautunut riittävän monen lohkon alle, ennen sitä tehdyt, kertaalleen käytetyt transaktiot voidaan levytilan säästämiseksi poistaa. Jotta tämä voisi tapahtua rikkomatta lohkon tiivistettä, transaktiot on hajautettu Merkle-puuhun [7][2][5], josta pelkkä juuri on sisällytetty lohkon tiivisteeseen. Vanhempia lohkoja voidaan sitten pakata tiiviimpään muotoon puun haaroja katkaisemalla. Lohkon sisällä olevia tiivisteitä ei tarvitse säilyttää.



Lohkon tunniste ilman transaktioita on kooltaan noin 80 tavua. Jos oletamme, että lohkoja syntyy 10 minuutin välein, 80 tavua * 6 * 24 * 365 = 4,2 Mt vuodessa. Koska vuonna 2008 myytyihin tietokonejärjestelmiin sisältyi tyypillisesti 2 Gt RAM-muistia ja Mooren laki ennustaa tietokoneiden muistikapasiteetin kasvavan näillä näkymin 1,2 Gt vuodessa, tallennustilan ei pitäisi muodostua ongelmaksi, vaikka lohkon tunnisteet olisikin säilytettävä muistissa.

8 Yksinkertaistettu maksunvarmennus

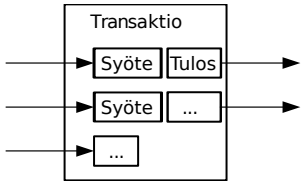
Maksut on mahdollista varmentaa ajamatta täyttä verkkosolmua. Käyttäjän tarvitsee ainoastaan säilyttää kopio lohkon tunnisteista siitä työntodisteketjusta, jonka on vakuuttunut olevan pisin muille verkkosolmuille tekemiensä kyselyjen perusteella, sekä löytää se Merkle-puun haara, joka linkittää kyseisen transaktion siihen lohkoon, johon se on aikaleimattu. Käyttäjä ei pysty varmistamaan siirtotapahtumaansa itse, mutta linkittämällä sen tiettyyn kohtaan ketjussa hän havaitsee, kun jokin verkon solmuista on kelpuuttanut sen, ja kyseisen siirron jälkeen lisätyt lohkot toimivat vahvistuksena siitä, että koko verkkokin on kelpuuttanut sen.



Varmennus on sinänsä luotettava, mutta vain niin kauan kuin vilpittömät solmut hallitsevat verkkoa; hyökkääjän onnistuessa valtaamaan verkon varmennuksesta tulee haavoittuvaisempi. Verkkosolmut pystyvät varmentamaan toistensa siirtotapahtumia, mutta hyökkääjä kykenee huijaamaan yksinkertaistettua menetelmää väärennetyillä siirroillaan niin kauan kuin pitää verkkoa vallassaan. Yksi tapa suojautua tältä olisi ottaa käyttöön hälytykset, joita verkkosolmut antaisivat epäkelpoja lohkoja havaitessaan. Hälytykset toimisivat kehotteena käyttäjän ohjelmistolle ladata lohko kokonaisuudessaan ja hälytyksiä aiheuttaneet siirrot ristiriitaisuuksien vahvistamiseksi. Yritykset, joiden maksuliikenne on vilkasta, haluavat luultavasti silti ajaa omia solmujaan, jotta niiden turvallisuus olisi riippumattomampaa ja varmennukset nopeampia.

9 Arvon jakaminen ja yhdistäminen

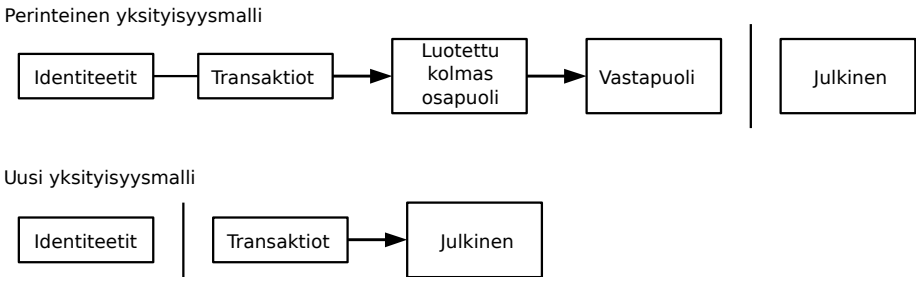
Vaikka jokaisesta siirretystä sentistä olisi mahdollista tehdä oma tapahtumansa, olisi niiden käsittely erillisinä kolikkoina kömpelöä. Jotta arvon jakaminen ja yhdistäminen olisi mahdollista, transaktiot koostuvat useista syöte- ja tulosarvoista. Normaalisti ne koostuvat joko yhdestä edellisen, suuremman tapahtuman syötteestä tai useista pienempiä summia yhdistävistä syötteistä sisältäen enintään kaksi tulosarvoa: toinen on itse maksu ja toinen lähettäjälle palautuvat mahdolliset vaihtorahat.



Huomautettakoon, että tässä ei synny ongelmaa siitä, että siirrot levittäytyvät laajalle (“fan-out”) eli ovat riippuvaisia useista siirroista, jotka puolestaan ovat riippuvaisia useista lisää. Koskaan ei tule tarvetta ottaa erillistä, täydellistä kopiota tietyn transaktion koko historiasta.

10 Yksityisyys

Pankkitoiminnan perinteisessä mallissa tietoihin pääsy on rajattu siten, että ainoastaan siirtotapahtuman osapuolet sekä luotettu kolmas osapuoli pääsevät niihin käsiksi, mikä luo jonkin verran yksityisyyttä. Kun kaikkien siirtojen on välttämätöntä olla julkisia, tällainen menetelmä on poissuljettu. Yksityisyyttä pystytään kuitenkin ylläpitämään katkaisemalla tiedonkulku toisesta kohtaa pitämällä julkiset avaimet nimettöminä. Kun joku lähettää jonkin summan jollekulle toiselle, siirto on kaikkien nähtävissä mutta ilman tietoja, jotka yhdistäisivät tapahtuman johonkin tiettyyn tahoon. Tämä on verrattavissa siihen, millaisia tietoja pörssit julkaisevat: yksittäisestä kaupasta kirjataan julkiselle ”nauhalle”aika ja suuruus ilmoittamatta, keitä sen osapuolet olivat.



Jokaisessa transaktiossa tulisi käyttää lisäpalomuurina uutta avainparia, jotta tapahtumia ei voi yhdistää yhteiseen omistajaan. Tältä ei kuitenkaan voi täysin välttyä useamman syötteen siirroissa, joista pakostakin ilmenee syötteiden olleen samassa omistuksessa. Tähän liittyy se riski, että jos tietyn avaimen omistaja paljastuu, tietoja yhdistelemällä voisi paljastua muitakin samalle omistajalle kuuluvia transaktioita.

11 Laskelmat

Tarkastellaanpa skenaariota, jossa hyökkääjä yrittää luoda vaihtoehtoisen ketjun vilpitöntä ketjua nopeammin. Vaikka hyökkääjän onnistuisi niin tehdä, koko järjestelmä ei yhtäkkiä olisi alttiina mielivaltaisille muutoksille, kuten että arvoa luotaisiin tyhjästä tai että hyökkääjä saisi anastettua rahaa, joka ei koskaan hänelle kuulunut. Solmut eivät kuitenkaan hyväksyisi epäkelpoja siirtoja maksuiksi, eivätkä vilpittömät solmut koskaan hyväksy lohkoja, joihin sisältyy epäkelpoja siirtoja. Hyökkääjä voi ainoastaan yrittää muuttaa yhtä omista transaktioistaan saadakseen takaisin rahat, jotka itse äskettäin käytti.

Vilpittömän ketjun ja hyökkääjäketjun välistä kilpajuoksua voidaan luonnehtia binominaaliseksi satunnaiskuluksi. Onnistuneeksi tapahtumaksi kutsutaan sitä, kun vilpitön ketju pitenee yhdellä lohkolla ja sen etumatka hyökkääjän ketjuun kasvaa (+1); epäonnistunut tapahtuma puolestaan on se, kun hyökkääjän ketju pitenee yhdellä lohkolla ja ketjujen välinen ero kaventuu (-1).

Hyökkääjän todennäköisyys saada kurottua välimatka umpeen altavastaajaasemasta on vastaava kuin uhkapelurilla Gambler's Ruin -probleemassa. Oletetaan että uhkapelaaja, jolla on rajaton määrä pelirahaa, aloittaa tappiolla ja pelaa äärettömän monta peliä yrittäessään päästä omilleen. Todennäköisyys sille, että uhkapelaaja pääsee omilleen tai hyökkääjä kuroo umpeen välimatkan vilpittömään ketjuun, voidaan laskea seuraavasti [8]:

p = todennäköisyys sille, että vilpitön solmu löytää seuraavan lohkon

q = todennäköisyys sille, että hyökkääjä löytää seuraavan lohkon

q_z = todennäköisyys sille, että hyökkääjä koskaan kuroo umpeen z lohkon etumatkan

$$q_z = \begin{cases} 1 & jos \quad p \leq q \\ (q/p)^z & jos \quad p > q \end{cases}$$

Koska oletamme, että $p > q$, todennäköisyys pienenee eksponentiaalisesti sitä mukaa kun hyökkääjän kiinni kurottavien lohkojen määrä kasvaa. Todennäköisyydet eivät ole hänen puolellaan, joten jos hän ei onnistu harppaamaan ajoissa reilusti eteenpäin, hänen mahdollisuutensa kutistuvat häviävän pieniksi hänen jäädessään yhä kauemmas taakse.

Tarkastellaan seuraavaksi, kuinka kauan uuden siirron vastaanottajan on odotettava ennen kuin voi olla riittävän varma siitä, ettei lähettäjä pysty tekemään siirtoon muutoksia. Oletetaan, että lähettäjä on hyökkääjä, joka haluaa jonkin aikaa uskotella vastaanottajalle maksaneensa tälle mutta kääntääkin siirron jonkin ajan kuluttua takaisin itselleen. Vastaanottajalle ilmoitetaan, jos näin tapahtuu, mutta lähettäjä toivoo sen olevan liian myöhäistä.

Vastaanottaja luo uuden avainparin ja antaa julkisen avaimen lähettäjälle juuri ennen allekirjoitusta. Tällä estetään se, että lähettäjä voisi etukäteen valmistella lohkojen ketjua ja jatkuvasti työstää sitä siihen asti, kunnes hänen onnistuisi ottaa riittävästi etumatkaa, jolloin hän sitten suorittaisi siirtonsa. Kun siirto on lähetetty, epärehellinen lähettäjä alkaa salassa työstää rinnakkaisketjua, joka sisältää vaihtoehtoisen version hänen siirrostaan.

Vastaanottaja odottaa, kunnes tapahtuma on lisätty johonkin lohkoon ja z määrä lohkoja on ketjutettu sen jälkeen. Hän ei tiedä tarkalleen, kuinka pitkälle hyökkääjä on edennyt, mutta kun oletetaan, että vilpittömiä lohkoja syntyy keskimäärin odotuksenmukaisella tahdilla, hyökkääjän edistyminen on Poissonin jakauma odotusarvolla:

$$\lambda = z\frac{q}{p}$$

Jotta saadaan selville se todennäköisyys, jolla hyökkääjä vielä voisi kyseisellä hetkellä kuroa välimatkan umpeen, kerrotaan jokaisen mahdollisen siihenastisen edistysaskeleen Poissonin todennäköisyystiheys sillä todennäköisyydellä, jolla hän tuosta hetkestä käsin pystyisi ottamaan etumatkan kiinni:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{Bmatrix} (q/p)^{(z-k)} & jos & k \leq z \\ 1 & jos & k > z \end{Bmatrix}$$

Järjestellään tämä uudelleen niin, ettei jakauman ääretön häntä tule laskettuksi mukaan...

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Muutetaan C-koodiksi...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Laskemalla muutamia tuloksia voimme nähdä todennäköisyyden pienenevän z:lla eksponentiaalisesti:

```
q=0.1
z=0    P=1.0000000
z=1    P=0.2045873
z=2    P=0.0509779
z=3    P=0.0131722
z=4    P=0.0034552
z=5    P=0.0009137
z=6    P=0.0002428
z=7    P=0.0000647
z=8    P=0.0000173
z=9    P=0.0000046
z=10   P=0.0000012

q=0.3
z=0    P=1.0000000
z=5    P=0.1773523
z=10   P=0.0416605
z=15   P=0.0101008
z=20   P=0.0024804
z=25   P=0.0006132
z=30   P=0.0001522
z=35   P=0.0000379
z=40   P=0.0000095
z=45   P=0.0000024
z=50   P=0.0000006
```

Ratkaistaan P-arvo alle 0,1 %...

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12 Päätelmät

Olemme esittäneet sähköisiin siirtoihin luottamuksesta riippumatonta järjestelmää. Lähdimme liikkeelle digitaalisista allekirjoituksista koostuvien kolikoiden tavanomaisesta viitekehyksestä: niiden omistusoikeuden hallinta on vahvaa, mutta ilman keinoja estää kaksinkertainen kulutus ne ovat vaillinaisia. Ratkaisuksi tähän esitimme vertaisverkkoa, jossa työntodistejärjestelmän avulla siirtotapahtumat kirjataan julkiseen historiaan, jonka muuttamisesta tulee äkkiä hyökkääjälle laskennallisesti epäkäytännöllistä, mikäli laskentatehosta suurin osa on vilpittömien solmujen hallussa. Verkko on kestävä kaikessa ohjaamattomassa yksinkertaisuudessaan. Solmut työskentelevät vähäisellä koordinaatiolla kaikki samaan aikaan. Solmujen ei tarvitse tunnistautua, sillä viestejä ei reititetä mihinkään tiettyyn paikkaan ja riittää, että verkko toimittaa viestit perille parhaansa mukaan. Solmut voivat poistua verkosta ja liittyä siihen milloin vain uudelleen, jolloin ne hyväksyvät työntodisteketjun todisteeksi siitä, mitä tapahtui niiden poissa ollessa. Solmut äänestävät laskentatehollaan ilmaisten hyväksyvänsä kelvolliset lohkot jatkamalla laskentaa niiden pohjalta ja hylkäävänsä epäkelvot lohkot kieltäytymällä rakentamasta niiden päälle. Kaikki tarvittavat säännöt ja kannustimet voidaan vahvistaa tällä konsensusmekanismilla.

Viitteet

[1] W. Dai. *b-money*. `http://www.weidai.com/bmoney.txt`. 1998.

[2] H. Massias, X. Serret Avila ja J.-J. Quisquater. "Design Of A Secure Timestamping Service With Minimal Trust Requirement". Teoksessa: *the 20th Symposium on Information Theory in the Benelux*. 1999.

[3] Stuart Haber ja W. Scott Stornetta. "How to Time-stamp a Digital Document". *Journal of Cryptology* 3 (1991), s. 99–111.

[4] Dave Bayer, Stuart Haber ja W. Scott Stornetta. "Improving the Efficiency and Reliability of Digital Time-Stamping". Teoksessa: *Sequences II: Methods in Communication, Security and Computer Science*. Springer-Verlag, 1993, s. 329–334.

[5] Stuart Haber ja W. Scott Stornetta. "Secure Names for Bit-Strings". Teoksessa: *in ACM Conference on Computer and Communications Security*. ACM Press, 1997, s. 28–35.

[6] Adam Back. *Hashcash - A Denial of Service Counter-Measure*. `http://www.hashcash.org/papers/hashcash.pdf`. 2002.

[7] R.C. Merkle. "Protocols for public key cryptosystems. Proc. 1980 Symposium on Security and Privacy". Teoksessa: IEEE Computer Society, 1980, s. 122–133.

[8] W. Feller. *Introduction to Probability Theory and its Applications*. 1957.

Suomentajilta

Satoshi Nakamoto yhdistelee monien eri tieteenalojen käsitteitä. Tähän sanastoon on kerätty niistä keskeisimmät. Linkkien takaa löydät aiheesta lisätietoa suomeksi tai englanniksi, mikäli suomenkielistä materiaalia ei käännöshetkellä löytynyt. Anna palautetta suomennoksesta: `Biocycle@protonmail.com`, tai ota yhteyttä kääntäjiin Telegramin kautta. Tämä suomennos on julkaisuversio 1.1.0.

Sanasto

Aikaleimapalvelin Timestamp server
`https://medium.com/brandin-kirjasto/21-oppituntia-2300d2f797c0`
1–4, 7

Binominaalinen satunnaiskulku Binomial random walk
`https://fi.wikipedia.org/wiki/Satunnaiskulku`
9

Erityistili tiettyä tarkoitusta varten (esim. sulkutili) Escrow
`https://pankkiasiat.fi/escrow-tili`
2

Internetin protokollaosoite IP
`https://fi.wikipedia.org/wiki/IP-osoite`
4

Julkinen ja yksityinen avain Public & private key
`https://fi.wikipedia.org/wiki/Julkisen_avaimen_salaus`
2, 8–10

Kaksinkertainen kulutus Double-spending
`https://en.wikipedia.org/wiki/Double-spending`
1–3, 12

Konsensusmekanismi Consensus mechanism
`https://fi.wikipedia.org/wiki/Konsensus-ptksenteko`
12

Levittäytyminen (laajalle) Fan-out
`https://en.wikipedia.org/wiki/Fan-out_(software)`
8

Nonssi - kertakäyttöinen arvo Nonce
`https://en.wikipedia.org/wiki/Cryptographic_nonce`
4

Parhaansa mukaan Best effort basis
`https://en.wikipedia.org/wiki/Best-effort_delivery`
1, 12

Poissonin jakauma Poisson distribution
https://fi.wikipedia.org/wiki/Poissonin_jakauma
10

Tiiviste Hash
https://fi.wikipedia.org/wiki/Tiiviste_(tietotekniikka)
1–6

Työntodiste Proof-of-work
https://medium.com/brandin-kirjasto/tyntodisteen-anatomia-475f371696f4
1, 4, 5, 7, 12

Uhkapelaajan turmio Gambler's Ruin
https://en.wikipedia.org/wiki/Gambler's_ruin
9

Vertaisverkko Peer-to-Peer network
https://fi.wikipedia.org/wiki/Vertaisverkko
1, 2, 4, 12