Iiro Henrik Iivanainen

Bitcoin mallina lohkoketjun toimintaperiaatteisiin

Tietotekniikan kandidaatintutkielma

8. kesäkuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Iiro Henrik Iivanainen

Yhteystiedot: iiheiiva@studnet.jyu.fi

Ohjaaja: Tytti Saksa

Työn nimi: Bitcoin mallina lohkoketjun toimintaperiaatteisiin

Title in English: The principles of blockchain with Bitcoin as an example

Työ: Kandidaatintutkielma

Opintosuunta: Information technology

Sivumäärä: 20+1

Tiivistelmä:

Lohkoketjuteknologiden periaatteita selvittäessä kryptovaluutta Bitcoin on asianmukainen tarkastelukohde, sillä se on ensimmäinen, menestynein sekä implementaatioltaan verrattain yksinkertainen lohkoketjujärjestelmä. Lohkoketjuteknologiat mahdollistavat hajautetusti ylläpidettyjä järjestelmiä, joille on ominaista mm. tiedon pysyvyys, neutraalisuus sekä anonyymisyys. Tässä kirjallisuuskatsauksessa selvitetään Bitcoinin toimintaperiaatetta sekä sitä, miten eri tekniikat johtavat lohkoketjuiteknologioille ominaisiin piirteisiin.

Avainsanat: lohkoketjuteknologia, Bitcoin, kryptovaluutta, toimintaperiaate

Abstract:

When studying blockchain technologies, the cryptocurrency Bitcoin makes for an adequate case study. Bitcoin is the first, the most succesful and relatively straight forwardly implemented blockchain system. Blockchain technologies enable the existance of decentralized systems that have attributes such as data persistance, neutrality as well as anonymity. This literature review explains the principles of Bitcoin while going over how the implementation leads to the attributes generally associated with blockchain tecnhologies.

Keywords: Blockchain technology, Bitcoin, cryptocurrenccy, principle

Termiluettelo

Bitcoin	Hajautetusti ylläpidettävä kryptovaluutta, jonka katsotaan olevan ensimmäinen modernin lohkoketjuteknologian implementointi.
Ethereum	Lohkoketjuteknologia, joka mahdollistaa sovellusten ajamisen Turing-täydellisten älysopimusten avulla. Nähdään nk. toisen sukupolven lohkoketjuteknologiana.
Konsensusmekanismi	Tapa, jolla järjestelmässä tullaan yhtämielisyyteen tehdyistä muutoksista.
Lohkoketju	Linkitettyä listaa muistuttava tietorakenne, jonka alkioita kutsutaan lohkoiksi. Jokaisen lohkon sisältöön kuuluu sitä edeltävän lohkon sisällöstä johdettu tiiviste.
Tiiviste	Vakiomittainen deterministinen arvo joka on johdettu mielivaltaisen kokoisesta syötteestä. Ei muunnettavissa alkuperäiseksi dataksi.
Vertaisverkko	Verkko, jossa ei ole keskitettyä palvelinta ja johon voi liittyä tai josta voi poistua vapaaehtoisesti.

Kuviot

Sisältö

1 Johdanto

Viime vuosikymmenen aikana Bitcoin ja muut kryptovaluutat ovat olleet kasvavan markkina-asemansa ansioista usein uutisotsikoissa ja niiden pohjana olevan lohkoktejuteknologian uskotaan olevat yksi viime aikojen tärkeimmistä innovaatioista (Chatterjee ja Chatterjee 2017). Lohkoketjuteknologiat mahdollistavat hajautetun järjestelmän ylläpidon, jonka ominaisiin etuihin lasketaan mm. avoimuus, anonyymisyys, puolueettomuus sekä tiedon muuttumattomuus ja pysyvyys.

Lohkoketjuteknologioihin kohdistuu suuren kasvun johdosta jatkuvaa tutkimusta uusien sovellusalojen löytämiseksi ja tekniikan parantamiseksi (Dabbagh, Sookhak ja Safa 2019). Kryptovaluutat ovat kehittyneet nopeammiksi ja ne pystyvät esittämään Bitcoinia monimuotoisempia sekä toiminnallisesti rikkaampia resursseja. Nykyaikaiset lohkoketjujärjestelmät, kuten Ethereum, mahdollistavat myös Turing-täydelliset älysopimukset. Ne ovat ohjelmia joiden back-end suoritetaan lohkoketjussa hajautetusti ja joita hallinnoidaan arvoa omaavan kryptovaluutan avulla (Vujičić, Jagodić ja Ranđić 2018). Bitcoinin verrattaisesta yksinkertaisuudesta ja rajoitteista huolimatta se on edelleen rahalliselta arvoltaan ylivoimaisesti merkittävin kryptovaluutta.

Tässä kirjallisuuskatsauksessa tutustutaan nk. ensimmäisen sukupolven lohkoketjujärjestelmien (*Blockchain 1.0*) periaatteisiin ja toteutukseen käyttämällä esimerkkinä Bitcoinia, jonka katsotaan olevan ensimmäinen modernin lohkoketjuteknologian implementointi (Chatterjee ja Chatterjee 2017). Aluksi käsitellään lohkoketjuteknologian taustalla olevia avainteknologioita, jotka mahdollistavan Bitcoinin toteutuksen. Seuraavaksi käydään läpi Bitcoinin toimintalogiikka tarkistelemalla sen protokollaa. Tavoitteena tutkimuksessa on löytää perustelut sille, mihin lohkoketjuteknologialle ominaiset piirteet perustuvat. Lopuksi tarkastellaan tarkemmin Bitcoinin konsensusmekanismia ja siihen liittyviä haavoittuvuuksia, sekä läpi käydyistä teknologioista tehdään yhteenveto kuvaamalla niitä kerrosmallin avulla.

2 Bitcoinin avainteknologiat ja protokolla

Ensimmäinen tekninen kuvaus modernin lohkoketjuteknologian toiminnasta annettiin Satoshi Nakamoton Bitcoinin valkoisessa kirjassa. Tuolloin tekniikasta ei vielä käytetty ilmaisua lohkoketju, vaan siihen viitattiin vertaisverkon ylläpitämänä aikaleimapalvelimena (*timestamp server*). Tarkkaa määritelmää sille mikä lasketaan lohkokoetjuteknologiaksi on vaikea antaa koska teknologia on vielä varhaisessa ja nopeasti kehittyvässä vaiheessa (Zıle ja Strazdiņa 2018). Bitcoin, kuten moni muu kryptovaluutta, edustaa nk. avointa lohkoketjujärjestelmää eli sen ylläpitoon ja käyttöön voi kuka tahansa ottaa osaa.

Avoimet lohkoketjuteknologiat käyttävät hyödykseen vertaisverkon tiedostojakoa ja ovat samaan tapaan hajautettuja järjestelmiä, mutta ne tarjoavat myös aidosti enemmän kuin tavanomainen tiedostonsiirto. Lohkoketjuun tallennetaan jaettu, järjestyksessä laajeneva tietokanta jonka ylläpito onnistuu konsensusalgoritmin ansiosta tuntemattomien osapuolten välillä (Nakamoto 2008). Bitcoinin tietokantaan tallennetaan rahansiirtoihin eli transaktioihin liittyvät tiedot sitä mukaa, kun niitä tapahtuu. Jokaisella käyttäjällä, eli verkon solmulla, on kopio tästä tietokannasta jotta ne voivat varmistaa tehtyjen lisäysten oikeellisuuden. Tämä tarkoittaa sitä, että kaikki transaktiot ovat asianmukaisilla luvilla tehtyjä, eikä varoja voi noin vain luoda tyhjästä.

Seuraavissa alaluvuissa tutustutaan lohkoketjujärjestelmän toteutusta yleisellä tasolla käyttämällä Bitcoinia esimerkkinä. Ensiksi käsitellään lohkoketjun toteutukselle keskeisiä tekniikoita. Sen jälkeen sekä lohkojen että niihin tallennettujen transaktioiden rakenteet käydään läpi. Kommunikoinnista vastaavan vertaisverkon rajoitteet huomioidaan Bitcoinin protokollassa, mutta sen toteutusta ei tässä tutkimuksessa tarkastella sitä tarkemmin. Lisäksi käydään läpi, miten uusia transaktioita lisätään tietokantaan, ja kuinka konsensusmekanismi mahdollistaa yhtämieliset päätökset solmujen kesken.

2.1 Tiivisteet

Tiivisteitä hyödynnetään erityisesti tiedon muuttumattomuuden sekä eheyden tarkistamisessa. Sitä voi ajatella datan sormenjälkenä, joka todistaa riittävän varmasti että se on johdettu

tietystä datasta paljastamatta kuitenkaan kyseessä olevaa sisältöä. Tiivistefunktion h määritelmä (Haber ja Stornetta 1991) on muotoa:

$$h : \{0,1\}^* \rightarrow \{0,1\}^l$$

Se siis ottaa mielivaltaisen määrän mitä tahansa binääridataa ja palauttaa sille määritetyn arvon jolla on l:n bitin vakiopituus syötteestä riippumatta. Bitcoinissa tämä toteutetaan SHA256-algoritmilla (Okupski 2014), jonka ulostulona on aina 256-bitin arvo. Itseasiassa Bitcoinin toteutuksessa informaatio tiivistetään kahdesti, sillä mikään ei estä syöttämästä itse tiivistettä funktiolle uudelleen.

Luottamus lohkoketjuteknoligaan vaatii pohjaksi toimivan tiiviste-algoritmin. Käytännössä sen on oltava nopeasti laskettavissa ja samaan tiivisteeseen törmääminen (*collision*) eri arvoilla on oltava erittäin epätodennäköstä. Mikäli funktio jakaa eri tiivisteitä tasaisesti mahdollisten syötteiden välillä, niin 256-bitin arvoavaruudessa samaan tiivisteen törmääminen kahdella eri arvolla on laskennallisesti käytännössä mahdotonta. Tiivistealgoritmin on oltava yksisuuntainen, eli tiivisteestä ei voi päätellä etukäteen eri sisääntuloja, tämä on erityisen tärkeää Bitcoinin louhinta (*mining*) prosessin toimivuudelle, joka takaa koko tietokannan integriteetin.

2.2 Avainparit ja digitaalinen allekirjoitus

Valuuttajärjestelmä, missä kuka tahansa voi kuluttaa kenen tahansa varoja ei ole toimiva. Jaetussa tietokannassa on oltava säännöt sille, miten eri varojen omistaminen varmennetaan pitäen siihen vaaditut tunnustiedot silti salaisina. Haastavalta kuulostavassa ongelmassa hyödynnetään epäsymmetriseen kryptografiaan perustuvaa avainparia ja sen mahdollistamaa digitaalista allekirjoitusta.

Jokaista transaktiota varten Bitcoinin käyttäjien lompakkosovellus luo ja säilyttää satunnaisen 256-bittisen arvon, jota kutsutaan yksityiseksi avaimeksi. Jokaista yksityistä avainta vastaa julkinen avain, joka voidaan johtaa siitä elliptisten käyrien laskutoimituksilla (Zheng ym. 2017). Toteutuksesta seuraa se, ettei yksityistä avainta voi johtaa julkisesta avaimesta. Julkista avainta voidaan ajatella osoitteena, jonka tiedetään kuuluvan tietylle käyttäjälle verkossa. Yksityinen avain vastaavasti toimii kuin salasana ja sen tulisi pysyä vain omistajan

tiedossa.

Digitaalinen allekirjoitus on todistus siitä, että osapuoli omistaa tietyn yksityisen avaimen ja on itse käyttänyt sitä jonkin digitaaliseen dokumentin salaukseen. Allekirjoitus voidaan suorittaa minkälaiselle datalle tahansa, ja se usein tehdään alkuperäisen tiedoston sijaan sen tiivisteelle sisällön suojaamiseksi (Haber ja Stornetta 1991). Bitcoinin protokollassa allekirjoitus tehdään lähettäjän omistamille varoille. Avaimet ovat toisiinsa kytköksissä siten, että toisella avaimella salattu tiedosto voidaan avata vain sille kuuluvalla arvoparilla. Kun käyttäjä salaa tiedoston yksityisellä avaimellaan, hän todistaa allekirjoituksensa koska muut solmut voivat varmistaa sen avaamalla tiedoston yleisesti tiedossa olevalla julkisella avaimella. Salaaminen suoritetaan lokaalisti joten yksityistä avainta ei tarvitse koskaan lähettää verkon kautta muualle.

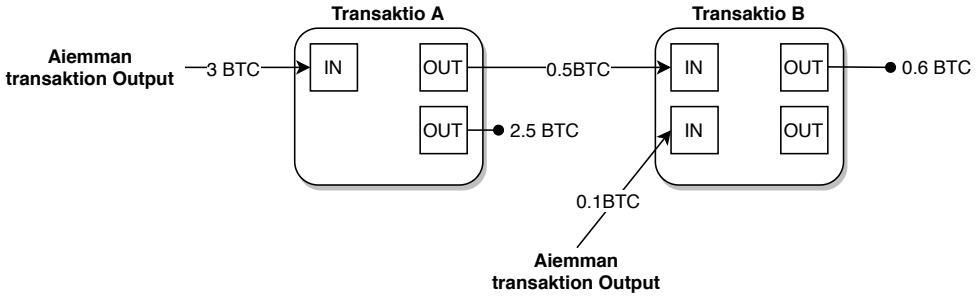
2.3 Transaktioiden rakenne ja allekirjoitus

Bitcoin-sovellukset muistuttavat käyttäjän näkökulmasta tavanomaisia pankkisovelluksia. Tietokannan rakenteen näkökulmasta tilejä ja niiden saldoja ei kuitenkaan ole olemassa sellaisenaan kuin ne käyttäjälle näkyvät, vaan ne ovat johdettuja lohkoihin kronologisesti talletetuista rahansiirroista eli transaktioista (Nakamoto 2008). Transaktiot ovat tilasiirtymiä joilla on sisääntulot, eli maksujen lähteet, sekä kaksi ulostuloa, maksettu määrä ja maksajalle palautettu vaihtoraha. Käytetyt ulostulot johtavat järjestyksessä seuraaviin transaktioihin (ks. kuvio 1). Tilasiirtymiä seuraamalla päädytään aina käyttämättömään ulostuloon, joka vastaa käytettävissä olevia bitcoineja ja on siihen liitetyn yksityisen avaimen omistajan lunastettavissa.

Lohko sisältää joukon transaktioita, jotka Bitcoinin käyttäjät ovat ilmoittaneet verkossa. Transaktioita on kahdenlaisia (Okupski 2014): tavallisia rahansiirtoja käyttäjältä toiselle ja jokaisen lohkon alkuun kirjatut nk. coinbase-transaktiot. Coinbase-transakitot ovat Bitcoinin tapa palkita lohkon muodostaneet louhijat sekä luoda uutta valuuttaa kiertoon. Siksi ne eivät viittaa sisääntuloihin tai vaadi aikaisemman omistajan allekirjoitusta. Tämän toteutuksen ansiosta varoja ei voi noin vain luoda tietokantaan tyhjästä, vaan niiden lähteet ovat aina kenen tahansa varmennettavissa. Lohkojen lisäämistä ketjuun, eli louhimista, käsitellään tarkem-

min kappaleessa 2.6.

Jokaiseen transaktioon merkitään sen hetkisen protokollan versio, joka määrittelee lohkon validointisäännöt, sekä haluttaessa lukitusaika jota ennen maksua ei haluta suoritettavan. Sisään- ja ulostulot ovat tallennettu transaktioon taulukkomuodossa. Bitcoinin omalla pinopohjaisella Script-kielellä on määritelty ulostuloihin ehdot niiden lunastamiselle, ja sisääntuloihin vastaavasti maksun autorisoivat allekirjoitukset. Allekirjoituksen ansiosta kukaan muu ei voi jälkikäteen muuttaa transaktion asetuksia, koska tämä vaatisi yksityisen avaimen tuntemisen (Nakamoto 2008). Transaktion viemä tallennustila lohkossa kasvaa kun siinä käytetään useampaa sisääntuloa ja monimutkaisempia skriptejä. Palvelumaksu (*transaction fee*) on tapa nopeuttaa rahansiirron lisäystä lohkoketjuun. Se on käyttäjän maksama ylimääräinen palkkio lohkon muodostajalle, jonka on tarkoitus saada tämä priorisoimaan transaktion käsittely.

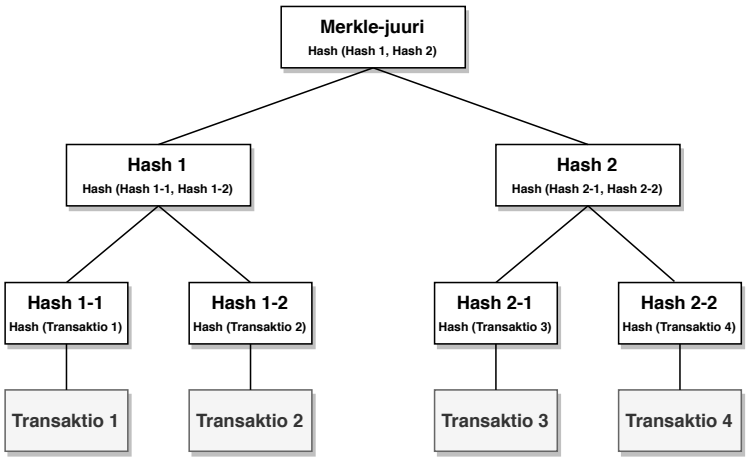


Kuvio 1. Kaksi transaktiota jossa ulostuloista 2,5 BTC:n vaihtoraha ja 0,6 BTC:n saatu maksu vastaavat nykyisen tilan käytettäviä bitcoineja. Kuvio ei huomioi palvelumaksua.

Jos käyttäjä kadottaisi yksityisen avaimensa, siihen liitetyt varat käytännössä katoaisivat kokonaan kierrosta, koska kukaan ei pystyisi todistamaan niitä omikseen. Toisaalta kryptovaluuttojen etuna on anonyymisyyttä varjeleva minimaalinen tunnistautuminen verrattuna esimerkiksi täsmällisiä henkilötietoja vaativaan pankkiin. Vaikka käyttäjä voi luoda useita avainpareja anonyymisyytensä takaamiseksi, niin tulee huomioida, että koska Bitcoinin tietokanta on avoin, niin erityisesti monen sisääntulon transaktioita analysoimalla on mahdollista yhdistää niiden maksujen tekijät toisiinsa (Nakamoto 2008).

2.4 Merkle-puu

Lohkoihin valitut transaktiot ovat esitettävissä Merkle-puuna (kuvio 2). Se on binääripuu, jota voidaan käyttää datan nopeaan verifioimiseen ja se auttaa lohkoketjun skaalattavuudessa vähentäen peruskäyttäjältä vaadittua levytilaa tietokannan kasvaessa (Buterin ym. 2014). Puun alimmat lehtisolmut edustavat kaikkia lohkoon tallennettuja transaktioita niiden keskeisessä järjestyksessä. Ylempien tasojen solmut koostuvat sisällöltään tiivisteistä, jotka ovat johdettu lapsisolmuista. Samat lehtisolmut muodostavat aina ylöspäin saman puun. Tämän takia välisolmuja ei tarvitse tallentaa lohkoihin (Nakamoto 2008). Lohkoketjun otsikkotietoihin tallennetaan Merkle-juuri eli puun juurisolmussa oleva tiiviste.



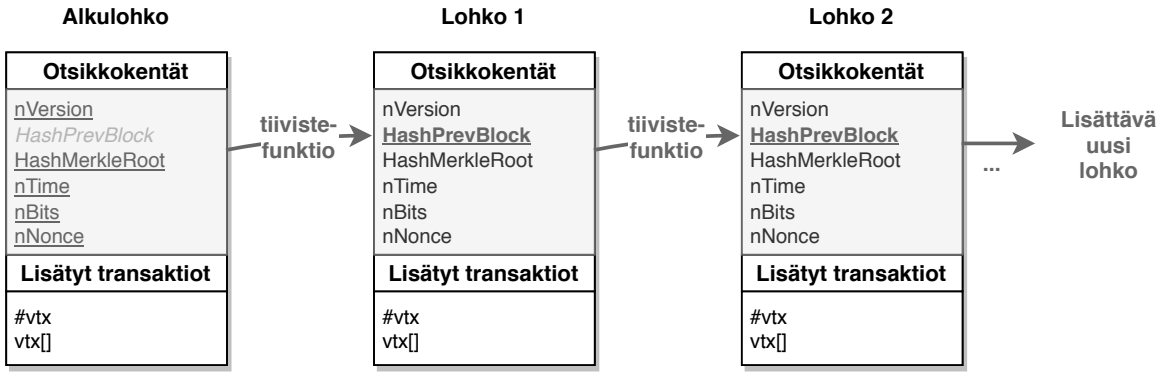
Kuvio 2. Esimerkki pienestä Merkle-puusta, jossa neljän lehtisolmun tiivisteistä johdetaan sisällöt puun muille solmuille

Merkle-puun avulla datan eheyden näkee suoraan puun juuresta. Mikäli jossain lehtisolmussa on virhe, se tulee näkyviin kaikissa ylemmissä solmuissa mukaanlukien johdetussa juurisolmussa. Datan eheyden takaavan tiivisteen voisi periaatteessa laskea myös ilman puurakennetta. Tämän toteutuksen etuna on se, että transaktioon tehty muutos voidaan paikantaa (Okupski 2014). Tällöin juurisolmun päivittämiseksi tulee laskea tiivisteet siitä haarasta, johon muutos tehtiin. Tasapainoisessa binääripuussa tämä tarkoittaa, että uusien tiivisteiden laskemista suoritetaan logaritminen määrä transaktioiden määrään nähden.

2.5 Lohkoketjun rakenne

Lohkoketju on linkitettyä listaa muistuttava tietorakenne, joka toimii kaikille käyttäjille jaettuna tietokantana. Bitcoinissa tietokantaa kutsutaan toiselta nimeltään jaetuksi pääkirjaksi (*shared ledger*). Kuten fyysisen kirjan sivuja, myös ketjun lohkoja on erittäin hankala muokata jälkikäteen. Samalla tavalla kuin pääkirjan sisältöä arvioi määräajoin luotettu taho, vertaisverkon käyttäjät tarkistavat kirjattavat muutokset pitäviksi.

Lohkot koostuvat otsikkokentistä ja dataosuudesta, joka sisältää Bitcoinin tapauksessa siihen lohkoon mahtuvien uusien transaktioiden tiedot. Rakenteen kannalta tärkein otsikkokenttä lohkossa on aiemmasta lohkosta muodostettu tiiviste (*hashPrevBlock*-kenttä kuviossa 3). Koska yksi pala tiivisteen muodostavasta datasta on sitä aiemmasta lohkosta muodostettu tiiviste, jokainen lohko on linkitetty sitä aikaisempiin lohkoihin. Poikkeuksena tälle on kovakoodattu alkulohko (*genesis block*), joka aloittaa ketjun.



Kuvio 3. Bitcoinin otsikkokentistä muodostetut tiivisteet linkittävät lohkot ketjuksi. Huomaa että transaktioita muuttaessa myös otsikkona tallennettu Merkle-juuri muuttuisi, vaatien tiivisteiden uudelleen laskemisen

Jos lohkoketjussa aiemman lohkon sisällöstä muuttaisi mitään tietoa, niin kaikki sitä seuraavien lohkojen tiivistekentät tulisi luoda uudelleen. Mitä enemmän lohkoja muodostetaan, sitä varmemmassa asemassa ketjun jälkimmäiset lohkot ovat. Lohkoketju itsessään ei kuitenkaan riitä tiedon pysyvyyden ja muuttumattomuuden takaamiseksi, koska uusien tiivisteiden laskeminen olisi pitkässäkin ketjussa triviaalisen helppoa. Tämän takia lohkojen muodostamiselle tulee asettaa laskennallisesti aikaa vaativa louhinta-prosessi, jota käydään läpi kappaleessa 3.1.

2.6 Uusien lohkojen muodostus ja ketjun haarautuminen

Bitcoinin valkoisessa kirjassa transaktioiden käsittely kuvattiin seuraavanlaisesti:

1. Uudet transaktiot kuulutetaan verkossa muille solmuille, jolloin niiden käsittely siirtyy jonotukseen.
2. Lohkoa muodostava solmu valitsee siihen tallennettavat transaktiot.
3. Muodostettuaan lohkon, solmu yrittää ratkaista konsensusmekanismin vaatiman tehtävän. Tätä kutsutaan Bitcoinissa louhimiseksi.
4. Kun ensimmäinen solmu verkossa löytää hyväksyttävän vastauksen tehtävään, se lähettää sen muille solmuille tarkistettavaksi.
5. Lohko lisätään ketjuun, kun muut solmut ovat varmentaneet siihen tallennettujen transaktioiden ja ratkaisun oikeellisuuden.
6. Solmut osoittavat uuden lohkon hyväksymisen rakentamalla sen perään seuraavaa lohkoa.

Lohkoa luova solmu haluaa priorisoida erityisesti ne transaktiot, joissa palvelumaksu on suurin sen viemään kokoon nähden, sillä yhden lohkon tallennustila on rajattu noin yhteen megatavuun. Uusi lohko muodostuu noin 10 minuutin välein ja lohkoketjuun hyväksytään keskimäärin 7 transaktiota sekunnissa (Zheng ym. 2017). Lohkon sisältämien transaktioiden hyväksyminen tietokantaan varmistuu sitä mukaa kun sen perään muodostuu uusia lohkoja, koska louhimishaaste tekee syvällä olevien lohkojen muuttamisesta käytännössä mahdotonta. Vertaisverkko kestää virheitä, eikä uusien tapahtumien tarvitse välttämättä saapua kaikille käyttäjille, vaan ne voivat pyytää niiden välistä hukkaamia lohkoja. (Nakamoto 2008)

Ketjun haarautuminen (*fork*) on poikkeustilanne, joka tapahtuu kun solmu onnistuu luomaan lohkon lähes samaan aikaan jonkin toisen solmun kanssa. Tämä aiheuttaa kilpailutilanteen, jossa solmut alkavat työstämään uutta lohkoa yleensä siihen haaraan, josta ne kuulivat verkossa ensimmäisenä. Solmut siis äänestävät laskentatehollaan siitä, mikä haara tulee jatkamaan virallista ketjua. Bitcoinin konsensusmekanismin mukaan pisimmän lohkoketjun haara on oikea, koska sen takana on todennäköisesti eniten laskennallista työtä (Nakamoto 2008). Solmut siirtyvät siihen ketjuun, johon uusi lohko muodostuu ensimmäisenä ja pois pudonneet transaktiot jotka olivat toisen haaran ketjussa siirtyvät takaisin jonotukseen. On mahdol-

lista että haarat kasvavat samaan aikaan, jolloin kilpailutilanne jatkuu mutta samanaikainen lohkojen muodostuminen yhä uudelleen muuttuu koko ajan epätodennäköisemmäksi.

3 Konsensusongelma ja teknologioiden yhteenveto

3.1 Proof of work -konsensusalgoritmi

Vertaisverkon päälle rakennetun lohkoketjun toiminnalle on välttämätöntä ratkaista nk. bysanttilaisten kenraalien ongelma (Zheng ym. 2017). Siinä erimielisten kenraalien on tultava eneimmistönä yhtämielisyyteen strategiasta, mutta sekä viestintäkanavissa että osapuolten välillä on luottamuksen puutetta. Vertauskuvassa lohkoketjujärjestelmän käyttäjät vastaavat kenraaleita ja strategia tarkoittaa tapaa valita transaktiot kronologisesti uuteen lohkoon eli konsensusta. Järjestys ei ole yksiselitteinen, koska vertaisverkossa tieto saapuu eri aikoina. Tämän lisäksi käyttäjä voi tahallaan tai kärsimättömästi ilmoittaa useita transaktioita jotka käyttävät samoja lähteitä maksussa. Tällöin tulee valita keskenään ristiriitaisista maksulähteiden käytöistä yksi viralliseksi tulkittu transaktio, jotta vältetään tämä nk. double-spending ongelma. Keskitetty toimielin joka hallitsee tietokantaa voisi tehdä nämä päätökset itse, mutta hajautetussa verkossa vaaditaan yhtämielisyyteen johtava konsensusalgoritmi.

Bitcoinin tapaa käyttää nk. Proof of work -konsensusalgoritmia pidetään Nakamoton merkittävimpänä innovaationa (Zıle ja Strazdiņa 2018; Buterin ym. 2014). Konsensuksen lisäksi se varmistaa transaktioiden johdonmukaisuuden ratkaisemalla double-spending ongelman, satunnaistaa lohkoa muodostavan solmun laskentatehon mukaan ja hallinnoi uuden valuutan luomista kiertoon palkiten samalla lohkoja prosessoivat solmut. Aiemmassa kappaleessa selvisi, kuinka pisin haara lohkoketjusta nähdään sen oikeana versiona, seuraavaksi konsensusmekanismia ja louhintaa käydään läpi tarkemmin.

Louhimisen ideana on muuttaa solmun muodostaman lohkoehdokkaan nonce-otsikkokenttää niin kauan, kunnes lohkosta johdettu tiivistearvo täyttää työtodistuksen haasteen asettaman ehdon, jolloin louhija voittaa oikeuden lisätä muodostamansa lohkon ketjuun. Haaste on siis aina uniikki muodostettavissa olevalle lohkolle ja sen onnistuminen perustuu ehdon täyttymisen todennäköisyydelle. Bitcoin käyttää samaa tekniikkaa tiivisteen hyväksymiselle kuin Hashcash-algoritmi, joka oli alunperin tarkoitettu roskapostaamisen vähentämiseen vaatimalla muutaman sekunnin edestä laskentatehoa jokaista viestiä kohden (Back 2002).

Louhimistehtävässä vaaditaan vähintään tietty määrä alkavia nollia tiivisteen binääriesityksestä. Esimerkiksi todennäköisyys sille, että ensimmäiset 10 bittiä ovat nollia on $0,5^{10} \approx$ 0,001 sillä oletuksella, että tiivistefunktiossa nollien ja ykkösten esiintyminen on yhtä todennäköistä. Vaatimalla yhden nollan lisää hyväksytyn tiivisteen löytämisen todennäköisyys puolittuu. Bitcoinissa haasteen vaikeus määritetään niin, että kaikkien louhijoiden yhteisen laskentatehon (*total hash rate*) huomioiden, noin joka kymmenes minuutti jokin solmu onnistuu muodostamaan uuden lohkon. Haastavuus päivittyy noin kahden viikon välein perustuen siihen miten nopeasti viimeiset 2015 lohkoa muodostettiin (Okupski 2014).

3.2 Bitcoinin manipulointi ja virhetoleranssi

Bysanttilainen virhetoleranssi tarkoittaa ehtoa, jolla hajautettua järjestelmää pystytään manipuloimaan. Lohkoketjujärjestelmissä sen mittaaminen on olennaisesti yhteydessä konsensusmekanismiin toteutukseen (Wang ym. 2018). Keinotekoisten käyttäjien luonti ei auta Bitcoinin lohkoketjun hallitsemisessa, koska sen konsensus perustuu laskentatehoon ja sääntöjä kunnioittavat solmut hyväksyvät vain sen mukaiset muutokset. Louhintaryhmittymä (*mining pool*) on joukko louhijoita, jotka ovat yhdistäneet laskentatehonsa ja jakavat lohkonmuodostuspalkkiot sen jäsenten kesken. Ne voivat olla yksittäisiä käyttäjiä tai laajoja palvelinfarmeja. Bitcoinin valtavan koon vuoksi yksikään louhintaryhmittymä ei hallitse lohkojen luontia (Buterin ym. 2014), mutta ne keskittävät valtaa järjestelmässä, jonka ideaaliksi katsotaan sen puolueettomuus.

Bitcoinin protokollassa on alusta asti tunnistettu potentiaalinen haavoittuvuus transaktioiden järjestyksen manipuloinnin muodossa (Nakamoto 2008). Siinä huijari pyrkii hyödyntämään double-spend-ongelmaa muuttamalla oikeaksi katsotun transaktion hänen omalle tililleen tehdyksi. Tämän tapahtuessa uhri havaitsee että on menettänyt oikeutensa maksettuihin bitcoineihin. Hyökkääjä haluaa tämän tapahtuvan tarpeeksi myöhään, kun hän on jo vastaanottanut kaupassa ostetut hyödykkeet. Onnistuakseen hyökkääjän tulisi onnistua pisimmän ketjuhaaran tekemisessä. Tämä vaatisi epärealistisesti laskentatehon eneimmistön (nk. 51% hyökkäys). Manipuloinnin vaativuus kasvaa eksponentiaalisesti, mitä syvemmissä lohkoissa olevia transaktioita halutaan uudelleenjärjestää. Bitcoinin käytössä on tapana odottaa uusien lohkojen muodostumista transaktion sisältävän lohkon päälle ennen kuin sen lopullisesta hy-

väksymisestä voi olla varma.

Voitollinen louhinta vaatii merkittäviä investointeja tiivisteiden laskemiseen erikoistuvien ASIC-piirien hankinnassa sekä jatkuvia energiakustannuksia (Buterin ym. 2014). Bitcoin kannustinjärjestelmän avulla suuren laskentatehon omaavat osapuolet pyritään pitämään rehellisinä louhijoina; ts. laskentatehon käyttäminen lohkoketjujen muodostamiseen tulisi olla kannattavampaa kuin double-spend-hyökkäyksen suorittaminen. Voittoa tavoittelevien osapuolten käytöstä ennustettaessa peliteorian näkökulma on erityisen tärkeässä osassa. Esimerkiksi itsekäs louhinta (*selfish mining*) on eräs strategia, jolla louhintavoitot pyritään maksimoimaan. Sen ideana on, että louhintaryhmittymä työstää omaa ketjuhaaraansa ilman kilpailua ja lopulta julkaisee sen, jos siitä saadaan pidempi kuin sen hetkinen virallinen ketju. On laskettu, että jos louhijaryhmittymän taakse saataisiin n. 25% louhijoiden kokonaislaskentatehosta, strategiasta tulisi kannattavampaa tavanomaiseen louhintaan verrattuna (Zheng ym. 2017).

3.3 Lohkoketjujärjestelmän eri tasot

Lopuksi tarkastellaan Bitcoinin tekniikoita ryhmittämällä ne toiminnan mukaan kuudeksi tasoksi (Wang ym. 2018). Tasot ovat yhteisiä kaikille lohkoketjuteknologioille, mutta niiden toteutuksessa ja tarjoamissa ominaisuuksissa voi olla merkittäviä eroja.

1. Datataso määrittelee lohkojen ja transaktioiden rakenteen, sekä viestien salauksen.
2. Verkkotaso mahdollistaa käyttäjien välisen kommunikoinnin ja tietokannan jaon. Avoimet lohkoketjuteknologiat kuten Bitcoin käyttävät vertaisverkkoa.
3. Konsensustaso määrittelee lohkoketjun tavan, jolla oikea versio päivitetystä tietokannasta hyväksytään yhteisesti. Sillä on tietty virhetoleranssi käyttäjien manipuloinnille.
4. Kannustintaso antaa syyn lohkoketjun rehelliseen ylläpitoon ja on tapa lisätä uusia bitcoineja kiertoon.
5. Sopimustaso määrittelee transaktioiden ehdot ja osapuolet. Bitcoinin Script-kieli on vastuussa sopimusten muotoilusta.
6. Sovellustaso tarjoaa käyttöliittymän sekä mahdollisia lisäpalveluja Bitcoin-järjestelmän päälle (*second layer protocol*).

4 Yhteenveto

Bitcoin-kryptovaluutta oli ensimmäinen tunnettu implementaatio lohkoketjuteknologiasta, jota on vuodesta 2009 seurannut moni muu kehittyneempi lohkoketjujärjestelmä. Lohkoketjuteknologioiden suurimmaksi vahvuudeksi katsotaan se, että ne perustuvat luottamuksen siasta todistukseen. Lohkoketjua ylläpitävät tuntemattomat osapuolet voivat tulla yhtämielisyyteen jaettujen sääntöjen ansiosta. Bitcoinin suurimmaksi edistysaskeleeksi katsotaan sen tapa muodostaa konsensus Proof of work -protokollan avulla. Jossa tietokannan viralliseksi versioksi katsotaan se, jonka takana on eniten käytettyä laskentatehoa.

Lohkoketju viittaa rakenteeseen, missä jokainen lohko viittaa sitä aikaisemmasta lohkosta muodostettuun tiivisteeseen. Ketju korostaa lohkojen keskeistä järjestystä ja tekee siten sen muokkaamisesta konensusmekanismin kanssa erittäin hankalaa. Lohkoihin tallennetut transaktiot ovat digitaalisen allekirjoituksen avulla tunnistuneiden käyttäjien tekemiä muutoksia tietokantaan. Koska tietokanta on avoin, kuka tahansa voi tarkistaa tehtyjen muutosten oikeellisuuden. Bitcoinin tunnistautumistiedot ovat minimaaliset, mutta niiden säilytys on käyttäjän omalla vastuulla, koska Bitcoinissa ei ole ylläpitävää keskitettyä elintä.

Käyttäjäkunta itsessään on osa järjestelmää, joten Käyttäjien motivaatiot huomioivan kannustinrakenteen analysointi on tärkeä osa toimivan lohkoketjujärjestelmän suunnittelussa. Lohkoketjun säännöt perustuvat todistukseen, mutta sen ylläpito vaatii järjestelmältä itseisarvoa, jota käyttäjät haluavat ylläpitää. Transaktioiden uudelleenjärjestely tunnistettiin jo Bitcoinin valkoisessa kirjassa mahdollisena hyökkäysrajapintana; mutta siihen vaaditut resurssit ovat lohkoketjun toteutuksen ansiosta epäkäytännöllisiä ja rahallisesti kannattamattomia. Itsekäs louhinta on uudempi strategia manipuloida lohkoketjua louhintapalkkion maksimoinniksi johon vaadittu louhijoiden osuus on tekee siitä paljon realistisemman uhan.

Lähteet

Back, Adam. 2002. “Hashcash-a denial of service counter-measure” (elokuu). http://www.hashcash.org/papers/hashcash.pdf.

Buterin, Vitalik, ym. 2014. “A next-generation smart contract and decentralized application platform”. *white paper* 3 (37). https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf.

Chatterjee, R., ja R. Chatterjee. 2017. “An Overview of the Emerging Technology: Blockchain”. Teoksessa *2017 3rd International Conference on Computational Intelligence and Networks (CINE),* 126–127. Lokakuu. doi:10.1109/CINE.2017.33.

Dabbagh, Mohammad, Mehdi Sookhak ja Nader Sohrabi Safa. 2019. “The evolution of blockchain: A bibliometric study”. *Ieee Access* 7:19212–19221. doi:10.1109/access.2019.2895646.

Haber, Stuart, ja W. Scott Stornetta. 1991. “How to Time-Stamp a Digital Document”. Teoksessa *Advances in Cryptology-CRYPTO’ 90,* toimittanut Alfred J. Menezes ja Scott A. Vanstone, 437–455. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-38424-3.

Nakamoto, Satoshi. 2008. *Bitcoin: A peer-to-peer electronic cash system.* Tekninen raportti. Satoshi Nakamoto Institute. https://git.dhimmel.com/bitcoin-whitepaper/.

Okupski, Krzysztof. 2014. “Bitcoin Developer Reference”. Teoksessa *Eindhoven.* https://www.lopp.net/pdf/Bitcoin_Developer_Reference.pdf.

Wang, S., Y. Yuan, X. Wang, J. Li, R. Qin ja F. Wang. 2018. “An Overview of Smart Contract: Architecture, Applications, and Future Trends”. Teoksessa *2018 IEEE Intelligent Vehicles Symposium (IV),* 108–113. Kesäkuu. doi:10.1109/IVS.2018.8500488.

Vujičić, D., D. Jagodić ja S. Ranđić. 2018. “Blockchain technology, bitcoin, and Ethereum: A brief overview”. Teoksessa *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH),* 1–6. Maaliskuu. doi:10.1109/INFOTEH.2018.8345547.

Zheng, Z., S. Xie, H. Dai, X. Chen ja H. Wang. 2017. “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends”. Teoksessa *2017 IEEE International Congress on Big Data (BigData Congress),* 557–564. Kesäkuu. doi:10.1109/BigDataCongress.2017.85.

Zıle, Kaspars, ja Renāte Strazdiņa. 2018. “Blockchain use cases and their feasibility”. *Applied Computer Systems* 23 (1): 12–20. https://content.sciendo.com/view/journals/acss/23/1/article-p12.xml.

Liitteet