Henri Järvi

BITCOIN TÄYSSOLMUN TOTEUTTAMINEN RASPBERRY PI -LAITTEELLA

Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintätekniikan koulutusohjelma
Toukokuu 2022

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	**Aika** Toukokuu 2022	**Tekijä/tekijät** Henri Järvi
Koulutus Tieto- ja viestintätekniikan koulutusohjelma		☒ AMK ☐ YAMK
Työn nimi Bitcoin täyssolmun toteuttaminen Raspberry Pi -laitteella		
Työn ohjaaja Kauko Kolehmainen		**Sivumäärä** 26

Opinnäytetyöni tarkoituksena on oppia enemmän Bitcoin-rahajärjestelmästä ja sen käytöstä. Opinnäytetyön tarkoituksena on myös tukea bitcointietoverkkoa, sillä bitcointietoverkko tarvitsee vapaaehtoisesti toimivia täyssolmuja, jotta se olisi täysin hajautettu ja ilman keskusvaltaa toimivana maksupalvelujärjestelmänä. Bitcoin-täyssolmu vahvistaa ja siirtää rahansiirtoja eteenpäin bitcoinvertaisverkossa.

Tämä opinnäytetyö ja aiheen opiskelu tulee edistämään osaamistani ja työllistymismahdollisuuksia kryptovaluuttojen tietämyksellä. Kryptovaluutat, lohkoketjut ja hajautetut rahoituspalvelusovellukset ovat kasvava ala, ja yritykset tarvitsevat näiden tietojen osaajia kasvavissa määrin. Toteutin täyssolmun Raspberry Pi 3:lla a opin sen takia myös paljon Linux-käyttöjärjestelmästä ja sen konfiguroimisesta.

Opinnäytetyöni toimii myös oppaana ja esimerkkinä kaikille, jotka haluavat tukea bitcointietoverkkoa ja rakentaa oman turvallisen täyssolmun.

Asiasanat
Bitcoin, Käyttöjärjestelmä, Lohkoketju Raspberry Pi, Täyssolmu

ABSTRACT

Centria University of Applied Sciences	**Date** May 2022	**Author** Henri Järvi
Degree programme Information and Communication Technologies		
Name of thesis Implementing a full Bitcoin node with Raspberry Pi		
Centria supervisor Kauko Kolehmainen		**Pages** 26

The purpose of my thesis is to learn more about the Bitcoin money system and its use. The purpose of the thesis is also to support the bitcoin data network, as the bitcoin data network needs voluntarily working full nodes to be fully decentralized and fully operating without a central government as a payment service system. The full Bitcoin node validates and forwards money transfers in the Bitcoin peer-to-peer network.

This thesis and the study of the topic will promote my skills and employment opportunities with my knowledge of cryptocurrencies. Cryptocurrencies, blockchains, and decentralized financial service applications are a growing industry and companies are increasingly in need of this knowledge. I implemented a full-node on a Raspberry Pi 3, and because of that, I also learned a lot about the Linux operating system and how to configure it.

My thesis also serves as a guide and example for anyone who also wants to support the Bitcoin data network and build their own fully secured full node.

Key words
Bitcoin, Operating system, Blockchain, Raspberry Pi, Full node

KÄSITTEIDEN MÄÄRITTELY

Altcoin

Kaikki muut kryptovaluutat paitsi bitcoin

ASIC

Mikropiiri, joka soveltuu vain yhteen tarkoitukseen

Bitcoin Cash

Bitcoinista eri versio, jossa on suuremmat lohkot

Hard Fork

Kun kryptovaluutta jakautuu kahteen tai useampaan protokollaan

Linux

Linux on avoimen lähdekoodin käyttöjärjestelmä

Lohkoketju

Tekniikka, jolla voidaan ylläpitää tietokantoja hajautetusti

NOOBS

Ohjelma, joka helpottaa käyttöjärjestelmän asentamista Raspberry Pi:lle

RAM

Hajasaantimuisti, tätä muistia voidaan lukea mielivaltaisessa järjestyksessä

Satoshi Client

Alkuperäinen bitcoin core -ohjelma

SHA-256

Kryptografinen tiivistefunktio, jossa on 256 bittiä

Swap file

Lisätila, joka voidaan valita hajasaantimuistille muilta muisteilta

Tor-ohjelma

Ohjelma, joka mahdollistaa anonyymin internetliikenteen

Uncomplicated Firewall

Helppo tapa konfiguroida palomuuri Linux-pohjaisilla käyttöjärjestelmillä

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on kehittää omaa ymmärrystä bitcoinin toiminnasta ja tukemisesta. Opinnäytetyön tavoitteena on myös olla opas bitcointäyssolmun toteuttamisesta Raspberry Pi:n avulla. Bitcoin-järjestelmä tarvitsee vapaaehtoisesti toimivia täyssolmuja pysyäkseen mahdollisimman hajautettuna ja ilman keskusvaltaa tarvitsevana rahajärjestelmänä. Täyssolmun toteuttamiseen valittiin Raspberry Pi, koska se on edullinen yhden piirilevyn tietokone ja näin ollen erityisen hyvä vaihtoehto erilaisiin projekteihin, joissa tietokoneen tehoja ei tarvita paljon. Raspberry Pi:n käyttöjärjestelmä Raspberry Pi OS (64bit) on myös hyvä valinta täyssolmun käyttöjärjestelmäksi turvallisuutensa takia.

Opinnäytetyössäni opastetaan vaiheittain, kuinka Raspberry Pi otetaan käyttöön, tehdään alustavat päivitykset ja asetukset, ja asennetaan bitcoin core -ohjelma. Opinnäytetyössäni käytetään myös kuvia opastuksessa ja opetellaan käyttämään komentoriviä Linux-pohjaisella Raspberry Pi OS-käyttöjärjestelmällä. Opinnäytetyössä oletetaan, että lukijalla on perustietämys tietokoneista ja internetin tietoturvasta.

2 BITCOININ HISTORIA

Bitcoin on avoimeen lähdekoodiin pohjautuva kryptovaluutta. bitcoinmaksujärjestelmää kuvaili ensimmäisen kerran Satoshi Nakamoto julkaisemassaan artikkelissa ” Bitcoin: A Peer-to-Peer Electronic Cash System” vuonna 2008 marraskuussa, ja julkinen lähdekoodi julkaistiin vuonna 2009 tammikuussa. Tämän jälkeen syntyi ensimmäinen bitcoinasiakasohjelma (eng. bitcoin-client), josta syntyivät ensimmäiset bitcointransaktiot. Satoshi Nakamoto louhi ensimmäisen bitcoinlohkon ja sai palkinnoksi 50 bitcoinia. Tätä ensimmäistä lohkoa bitcoinlohkoketjussa kutsutaan nimellä ”genesis block”. Hal Finney teki Satoshin kanssa ensimmäisen bitcointransaktion, kun Satoshi lähetti Finneylle 10 bitcoinia sen jälkeen, kun Finney latasi bitcoinasiakasohjelman tietokoneellensa. Hal Finney on amerikkalainen ohjelmistokehittäjä, joka oli myös ensimmäisiä ihmisiä edistämässä bitcoinjärjestelmää. Nakamoto oli aktiivinen bitcoinin kehittäjä vuoden 2010 joulukuuhun asti. (Chohan 2022, 8–10.)

Satoshi Nakamoto halusi ratkaista ongelman, joka oli tarve kolmannen osapuolen luottamiseen, käyttäessään nykyisiä maksupalvelujärjestelmiä. Nakamoton mukaan digitaaliset allekirjoitukset maksuihin olisivat osa ratkaisua, mutta eivät riittäviä, sillä tarvittaisiin vielä luottamista kolmanteen osapuoleen. Digitaalisilla allekirjoituksilla ei estettäisi kaksinkertaista kulutusta ja näin ollen tarvittaisiin silti luottoa kolmanteen osapuoleen. Kaksinkertainen kulutus on prosessi, jossa samaa bitcoinia voidaan käyttää useasti. Nakamoto kertoo, että ratkaisu digitaalisen rahan käyttämisen kahteen kertaan on käyttämällä vertaisverkkoa. Verkolla maksut leimataan tiivistämällä ne jatkuvaan ketjuun hash-pohjaisella työntodisteella, muodostaen ketjun, jota ei voi muuttaa ilman työtodisteen uudelleen tekemistä. (Nakamoto 2009.) Nakamoto halusi luoda maksupalvelujärjestelmän, jossa ei olisi tarpeellista luottaa kuin kryptografiaan.

Bitcoinin avoin lähdekoodin julkaisemisen jälkeen alettiin kehittelemään uusia kryptovaluuttoja, ja näitä kutsuttiin nimellä ”altcoin”. Ensimmäinen altcoin oli namecoin, joka julkaistiin vuonna 2011 keväällä. Maailmanlaajuinen maksupalvelu BitPay ilmoitti, että vuoteen 2012 mennessä yli 1000 kauppiasta oli hyväksynyt bitcoinin käyttämisen BitPayssä. Ensimmäiset maailmanlaajuiset sääntelyt koskivat bitcoinia vuonna 2013, ja ne koskivat bitcoinlohkoketjun jakautumista kahdeksi eri ketjuksi. (Chohan 2022, 10–15.)

3 BITCOIN-JÄRJESTELMÄN TOIMINTAPERIAATE

3.1 Bitcoinin louhinta

Bitcoin-louhinta on prosessi, jossa päivitetään hajautettua kirjanpitoteknologiaa. Louhinta tehdään käyttämällä tietokoneen prosessorin tehoja matemaattisen ongelman ratkaisemiseen, jolla saadaan uusi lohko lisättyä lohkoketjuun. Kun tietokone arvaa numeron oikein, louhintaohjelma määrittää, mitkä nykyhetken vireillä olevista rahansiirroista siirretään seuraavaan lohkoon lohkoketjussa. Tämä tietokone muuttaa bitcoin rahansiirtohistoriaa pysyvästi. Jokainen täyssolmu bitcoinverkossa päivittää transaktiohistorian ja lisää sinne tämän uuden transaktion. Uusi lohko ja numeroarvaus lähetetään muille tietokoneille, jotka vahvistavat sen. (Ghimire 2019, 26–27.)

Jokainen tietokone, joka vahvistaa vastauksen, päivittää uuden lohkon lohkoketjuun niillä rahansiirroilla, jotka valittiin mukaan lohkoon. Bitcoinrahajärjestelmään tulee uusia bitcoineja noin 10 minuutin välein. Bitcoin-järjestelmä palkitsee näillä uusilla bitcoineilla niitä tietokoneita, jotka ratkaisivat matemaattisen ongelman. Tietokone, joka ratkaisee ongelman, saa myös rahansiirtokulut palkinnokseen niistä rahansiirroista, jotka louhintaohjelma valitsi uuteen lohkoon. Muut tietokoneet bitcoinrahajärjestelmässä ovat nyt vahvistaneet uuden lohkon ja näin ollen uuden lohkon rahansiirtoja on käytännössä mahdotonta peruuttaa. (Ghimire 2019, 26–27.)

Bitcoinin kehittäjä Satoshi Nakamoto asetti sääntöjä bitcoinille tavalla, jonka mukaan mitä enemmän tehoja bitcointietoverkolla on, sitä vaikeampi matemaattinen ongelma on tietokoneille ratkaistavaksi. Näin ollen mitä enemmän tehoja on ongelmanratkaisuun, sitä vaikeampi se tulee olemaa. Sama toimii myös toisinpäin eli jos bitcoinverkon tehot laskevat, niin ongelman vaikeusaste laskee myös. Tämän tarkoituksena on se, että uusia bitcoineja tulisi koko ajan samaa tahtia. Tällä hetkellä uusia bitcoineja tulee maailmaan 6,25 kappaletta noin joka 10 minuutin välein. (Ghimire 2019, 29.)

Puoliintuminen (eng. halving) on tapahtuma bitcoinverkossa, jossa tietokoneet, jotka louhivat bitcoinia, saavat puolet vähemmän bitcoinia palkinnoksi kuin aikaisemmin. Tämä tapahtuu, joka 210 000 lohkon välein, joka on noin 4 vuoden välein. Seuraava puoliintuminen tapahtuu noin maaliskuussa vuonna 2024. Tämä puoliintumisprosessi jatkuu aina niin kauan, että kaikki 21 miljoonaa bitcoinia on louhittu. Se tapahtuu arviolta vuonna 2140. (Ghimire 2019, 27–28.) Ei ole

varmuutta, mitä tapahtuu sen jälkeen, kun kaikki bitcoinit on louhittu. On arvioitu, että viimeisen puoliintumisen jälkeen bitcointietoverkon tietokoneita palkitaan vain transaktiokustannuksilla.

Bitcoinin alkuaikoina sitä louhittiin keskussuorittimilla. Vuonna 2011 siirryttiin näytönohjaimiin ja vuonna 2013 markkinoille tuli tietokone, joka oli tarkoitettu vain bitcoinin louhintaan. Sen nimi on ASIC (application specific integrated circuit). Näillä tietokoneilla ei voinut tehdä mitään muuta kuin louhia SHA-256-algoritmia. Bitcoin-louhintapooliksi kutsutaan tietokoneita, jotka louhivat bitcoinia yhdessä ja jos jokin tietokone arvaa oikein, niin kaikki louhintapoolin tietokoneet saavat bitcoinia ja se jaetaan tietokoneitten tehojen mukaan. Näin ollen myös yksityiset louhijat voivat aloittaa louhimisen ja saada edes vähän bitcoinia. (Suman. 2019.). Tällä hetkellä suurin louhintapool on nimeltään Foundry USA 24,4 %, toiseksi suurin on AntPool 14,4 % ja kolmanneksi suurin on F2Pool 13,5 %. (Pool Distribution 2022.)

3.2 Bitcoin-täyssolmu

Täyssolmut ovat solmuja, jotka ylläpitävät koko bitcoinrahajärjestelmän maksuhistoriaa. Andreas Antonopoulos (2017) kertoo, että vielä parempi nimitys olisi täyslohkoketjusolmu. Bitcoinin alkuaikana kaikki solmut olivat täyssolmuja ja nykyään bitcoin core -ohjelma on myös täyssolmu. Muutamassa vuodessa on tullut uudenlaisia bitcoinasiakasohjelmia, jossa ei ylläpidetä koko lohkoketjun historiaa. Näitä solmuja kutsutaan ”kevyiksi asiakasohjelmiksi”. (Antonopoulos 2017, 180–182.)

Täyssolmut itsenäisesti ylläpitävät, rakentavat ja varmistavat kaikki lohkot bitcointietoverkossa ensimmäisestä lohkosta asti (genesis block) aina uusimpaan tiedettyyn lohkoon lohkoketjussa. Täyssolmu voi itsenäisesti ja virallisesti varmistaa mitä tahansa rahansiirtoja turvautumatta muihin solmuihin lohkoketjussa. (Ghimire 2019, 15–16.)

Bitcoin-täyssolmu saa päivityksiä tietoverkolta uusista lohkoista, jotka täyssolmu sitten varmistaa ja sisällyttää omaan kopioonsa lohkoketjusta. Bitcoin perustuu täysin täyssolmuihin. Täyssolmut varmistavat transaktiot itsenäisesti ja ilman luottoa kolmanteen osapuoleen. Bitcoin-täyssolmu vaatii noin 250 gigatavua pysyvää tilaa varastoidakseen koko lohkoketjun ja useamman päivän synkronoidakseen lohkoketjun tietoverkon kanssa. Tämä on hintana itsenäisestä ja keskushallinnosta vapaasta järjestelmästä.

On olemassa muutamia erilaisia täyssolmuasiakasohjelmia, jotka hieman eroavat toisistaan. Bitcoin-täyssolmuasiakasohjelmat voivat erota toisistaan käyttämällä eri koodauskieltä tai ohjelmistoarkkitehtuuria. Selvästi käytetyin sovellus on Satoshi client, jota käytetään yli 75 %:ssa kaikista bitcointäyssolmuasiakasohjelmista. (Antonopolous 2017, 180.)

3.3 Bitcoin skaalattavuusongelma

Bitcoinrahajärjestelmän rahansiirtojen kasvu on herättänyt huolta sen skaalautumiskyvystä. Bitcoin-rahajärjestelmään lisätään lohkoja vain tietyin väliajoin, sen maksimi transaktioiden määrä per lohko, on rajattu lohkon koon suuruudella. Bitcoin-lohkokokojen kasvaessa on alettu etsimään uusia tekniikoita, joilla voitaisiin parantaa bitcoinrahajärjestelmän skaalautumista. Bitcoinlohkoketjulta menee 10 minuuttia tai enemmän transaktioiden vahvistamiseen, ja se pystyy tekemään vain 7 transaktiota sekunnissa. Tämä on hyvin hidas vaihtoehto verrattuna esim. Visa-maksujärjestelmään, joka kykenee tekemään jopa 56 000 transaktiota sekunnissa. (Croman, Decker, Eyal, Gencer, Juels, Kosba, Miller, Saxena, Shi, G¨un, Sirer, Song, Wattenhofer 2016, 3–12.)

3.3.1 Lohkon koon suurentaminen

Bitcoinlohkon kokorajoitus muodostaa ”pullonkaulan” bitcoinmaksukapasiteetin kanssa. Tämä johtaa transaktiomaksujen suurenemiseen ja transaktioiden hidastumiseen. Lohkon kokorajoitus on herättänyt paljon keskustelua siitä, kuinka bitcoinista saataisiin skaalattua nopeampi ja halvempi maksupalvelujärjestelmä. Alun perin Satoshi Nakamoto asetti bitcoinlohkon kokorajoitukseksi 1 MB, mutta ei kertonut syytä tähän. Ainut mahdollisuus suurentaa bitcoinlohkon kokoa on tehdö ”hard fork” eli jakaa lohkoketju kahteen osaan, jossa toinen on uudempi versio. Toisessa olisi nän ollen suurempi lohkon koko ja toinen olisi sama kuin aikaisemmin. Näin toimittiin, kun osa bitcoinin kehittäjistä, louhijoista ja aktivisteista halusi muokata bitcoinia niin, että se sopisi paremmin maksuihin niin kuin bitcoin oli alun perin tarkoitettu heidän mielestänsä. Tästä syntyi bitcoin cash, jossa oli 8 MB:in lohkon kokorajoitus. (Croman 2016, 1–4.)

3.3.2 Salamaverkko

Lighting network eli salamaverkko kehitettiin vuonna 2015. Salamaverkon kehittäjät ovat Joseph Poon ja Thaddeus Dryja. Salamaverkko on riippuvainen lohkoketjuteknologiasta. Salamaverkolla on mahdollista tehdä turvallinen verkosto, jossa pystyy lähettämään bitcoinia suurella määrällä ja suurella nopeudella. Salamaverkkoa voidaan käyttää, jos käytetään oikeita bitcoinmaksuja, sen alkuperäistä älysopimuksien (eng. smart contracts) skriptauskieltä. Salamaverkko toimii niin, että verkkoa kuormittavia mikromaksuja ei viedä lohkoketjuun. Tätä toimintaperiaatetta kutsutaan nimellä Layer 2.

Mikromaksujen pois siirtämien lohkoketjusta nopeuttaa ja halventaa bitcointransaktioita eli tällä toiminta periaatteella pyritään pääsemään yhtä nopeaksi ja helpoksi kuin luottokorttia käytettäessä. Salamaverkon käyttämiseen tarvitaan salamasolmu (eng. lightning node). Salamasolmu on luoteeltaan avoin eli samanlainen kuin bitcoin. Salamasolmu eroaa bitcoinsolmusta siten, että salamasolmun ylläpitäjän tulee valita kuinka monta kanavaa hän avaa. Tämän jälkeen ylläpitäjä avaa salamalompakon, tallettaa sinne bitcoinia ja valitsee tietyille kanaville bitcoin kapasiteetin eli kuinka paljon bitcoinia voi tietyssä kanavassa vaihtaa. Salamaverkon maksujen päätteeksi eli esimerkiksi päivän lopussa kaikki siinä tehdyt maksut viedään lopuksi bitcoinlohkoketjuun pysyviksi maksuiksi ja lopullinen lompakkojen tase päivitetään jokaiseen täyssolmuun. (Poon & Dryja 2016, 1–4.)

4 TÄYSSOLMUN TOTEUTTAMINEN

4.1 Laitevaatimukset

Työssä tarvitaan Micro SD-muistikortti 8 GB, ulkoinen kovalevy 512 GB tai suurempi, Raspberry Pi 3 tai uudempi malli, HDMI-kaapeli, hiiri, näppäimistö ja USB-adapteri 2,5 A.

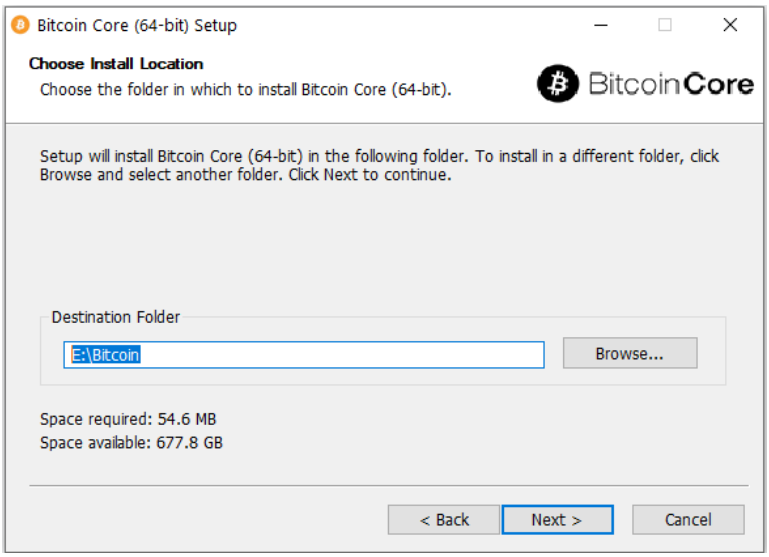
Micro SD-muistikorttiin tallennetaan vain NOOBS-ohjelma ja käyttöjärjestelmä, joten 8 GB on riittävä määrä. Ulkoiseen kovalevyyn ladataan koko lohkoketju, joka on yli 300 GB ja kasvaa koko ajan. Tällä hetkellä 512 GB on sopiva määrä bitcoinlohkoketjulle. Työssä tarvitaan myös internetyhteys. Raspberry Pi tarvitsee sitä enemmän virtaa mitä enemmän komponentteja siihen liitetään. Nämä liitännät tarvitsevat yli 2 ampeerin virtalähteen toimiakseen. Siihen riittää USB-adapteri 2,5 A.

Raspberry Pi on yhden piirilevyn tietokone ja se sopii hyvin projekteihin, joissa ei tarvita paljon tietokoneen tehoja. Raspberry Pi on edullinen, noin luottokortin kokoinen tietokone, joka voidaan liittää tietokoneen näyttöön tai TV:hen. Raspberry Pi:llä toimivat hiiri ja näppäimistö. Raspberry Pi toimii kuin mikä tahansa tietokone ja se on suunniteltu käytettäväksi erilaisiin projekteihin ja tehtäviin. Raspberry Pi:tä käytetään paljon myös erilaisen ohjelmointikielien ja tietokoneiden opetuksissa. Työssä voitaisiin käyttää muitakin yhden piirilevyn tietokoneita, mutta Raspberry Pi on kaikista suosituin, ja näin ollen sen käytöstä löytyy eniten oppaita internetistä (Raspberry Pi Foundation).

Työssä valittiin täyssolmun toteuttaminen Raspberry Pi:llä ja Raspberry Pi OS-käyttöjärjestelmä, koska Linux-tyyliset käyttöjärjestelmät on toteutettu avoimella lähdekoodilla. Tämä tarkoittaa sitä, että kuka tahansa pystyy tarkastamaan ja varmistamaan, että niiden koodissa ei ole virheitä. Näin ollen Raspberry Pi OS-käyttöjärjestelmä on paras vaihtoehto toteuttaa täyssolmu. (Taylor 2018.)

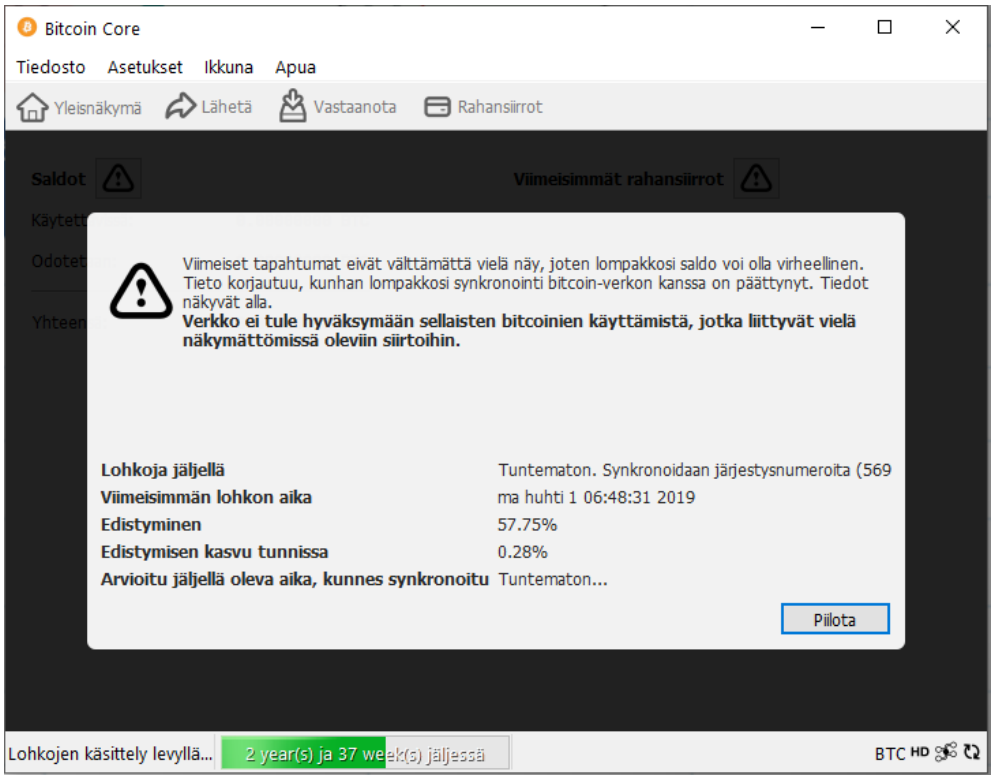
4.2 Lohkoketjun lataaminen ja tarkistus

Bitcoin lohkoketjun lataamisessa ja synkronoimisessa kuluu aikaa useita viikkoja, joten se kannattaa aloittaa hyvissä ajoin. Bitcoinin-lohkoketjun koko on yli 300 gigatavua ja sen koko nousee koko ajan uusien lohkojen louhimisessa. Raspberry Pi 3:n suoritustehot ja muisti eivät riitä koko lohkoketjun tarkistamiseen, joten ladataan lohkoketju pöytätietokoneella. Bitcoin core -asennusohjelma löytyy osoitteesta bitcoincore.org/en/download. Ladataan lohkoketju ulkoiselle kovalevylle (KUVA 1).



KUVA 1. valitsen ulkoisen kovalevyn bitcoin core -ohjelman datalle

Bitcoin core -ohjelma alkaa lataamaan lohkoketjua vuodelta 2009 lähtien. Tässä kuluu aikaa useita viikkoja riippuen tietokoneen tehoista (KUVA 2).



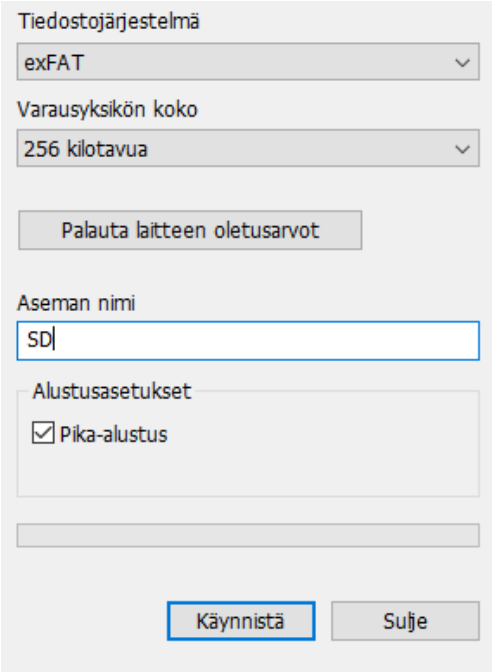
KUVA 2. bitcoinsolmujen lataaminen bitcoin core -ohjelmalla

On kaksi erilaista tapaa käyttää bitcoin core -ohjelmaa, toisessa on graafinen käyttöliittymä ja toisessa on vain komentorivi, jolla voidaan konfiguroida bitcoin core -ohjelmaa. Graafisen käyttöliittymän nimi

on bitcoin-qt ja vain komentorivillisen bitcoin core -ohjelman nimi on bitcoind. Työhön valitaan bitcoin-qt, koska bitcointäyssolmua käytetään myös bitcoinlompakkona. Graafinen käyttöliittymä sopii paremmin bitcoinlompakon käyttämiseen. Bitcointäyssolmu voidaan käynnistää myöhemmin ilman graafista käyttöliittymää, koska nämä kaksi tapaa käynnistää bitcoin core -ohjelma, ovat yhteensopivia toistensa kanssa, käyttävät samoja komentoja, lukevat ja kirjoittavat samoille tiedostoille. Bitcoin core -ohjelma alkaa välittömästi synkronoida lohkoketjua sen käynnistämisen jälkeen. Jos tietokoneella, jolla lohkoketjua synkronoidaan, on paljon muistia, voidaan sitä käyttää hyväkseen synkronoidakseen nopeammin. Suurempi välimuistin käyttö voidaan hyödyntää menemällä bitcoin core -ohjelmassa settings/options/open configuration file ja kirjoittamalla tähän tiedostoon **dbcache=6000**. Oletus välimuistin käyttö on 300 megatavua. Vaihdetaan välimuistin käyttö 6 gigatavuun, tallennetaan tiedosto ja käynnistetään uudelleen bitcoin core -ohjelman.

4.3 Raspberry Pi:n alustaminen

Raspberry Pi:n alustaminen aloitetaan micro SD-muistikortin alustamisesta. Micro SD-muistikortti alustetaan exFat-tiedostojärjestelmään. Windows tukee kahta eri tiedostojärjestelmävaihtoehtoa NTFS ja exFat. Raspberry Pi OS-käyttöliittymä voi vain lukea NTFS-tiedostojärjestelmän dataa, mutta ei kirjoittaa siihen uutta dataa. Tästä syystä micro SD-muistikortti tulee alustaa exFat-tiedostojärjestelmäksi. Valitaan Windows-tietokoneelta micro SD-muistikortti, klikataan oikealla hiirenpainikkeella SD-muistikorttia, valitaan ”alusta”. Tämän jälkeen vaihdetaan NTFS-tiedostojärjestelmä exFat-tiedostojärjestelmään, valitaan varausyksikön kooksi 256 kilotavua ja painetaan käynnistä (KUVA 3).

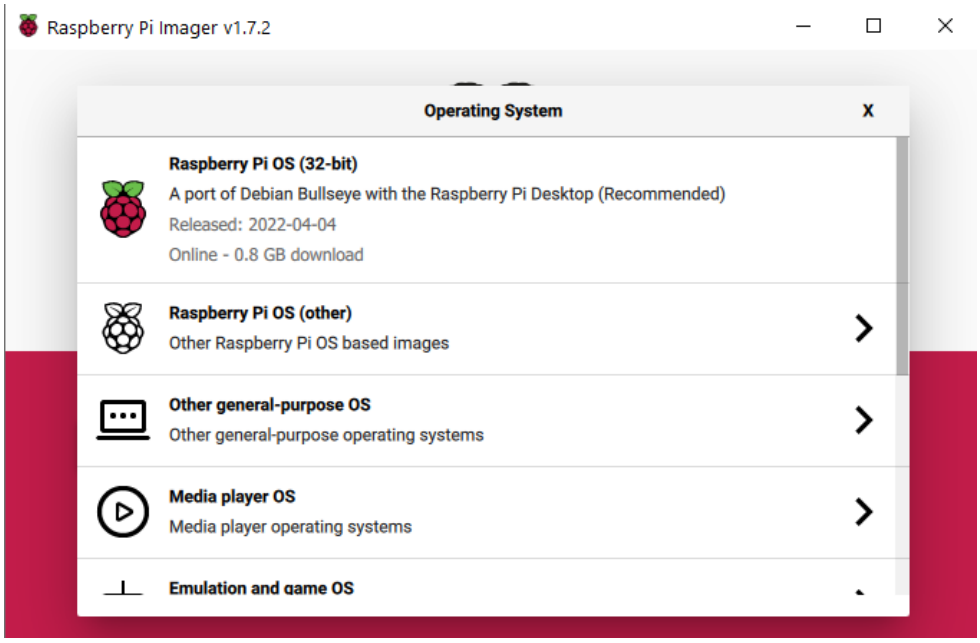


KUVA 3. tiedostojärjestelmän muuttaminen Windowsilla

Seuraavaksi ladataan NOOBS-tiedosto SD-muistikortille. NOOBS-tiedosto löytyy Raspberry Pi:n Github sivuilta https://github.com/raspberrypi/noobs. Tiedoston purkamisen jälkeen kopioidaan kaikki tiedostojen sisältö micro SD-muistikortille, josta Raspberry Pi:llä voi käynnistää NOOBS-ohjelman. Pelkästään puretun tiedoston kopioiminen ei SD-muistikortille ei riitä, vaan täytyy kopioida puretun tiedoston sisältö. NOOBS-ohjelma helpottaa Raspberry Pi:n ensimmäistä konfiguraatiota huomattavasti. NOOBS-tiedoston avulla asennetaan Raspberry Pi OS-käyttöjärjestelmä Raspberry Pi:lle. Raspberry Pi OS on debian-pohjainen käyttöjärjestelmä, jota suositellaan käytettäväksi Raspberry Pi:llä. Debian on linux-jakelupaketti. Raspberry Pi OS-käyttöjärjestelmän asennuksen jälkeen NOOBS-ohjelma jää SD-muistikortille. NOOBS-ohjelmaa käyttämällä voidaan vaihtaa tai uudelleen asentaa käyttöjärjestelmä Raspberry Pi:ssä. NOOBS-ohjelmalla voidaan myös helposti editoida asennetun käyttöjärjestelmän config.txt konfiguraatiotiedostoa. Config.text tiedosto toimii konfigurointi tiedostona Raspberry Pi:ssä. Asentamalla NOOBS-ohjelman config.text tiedostoon Raspberry Pi käynnistää NOOBS-ohjelman aina ennen kuin käynnistää käyttöjärjestelmän.

Ensimmäisellä Raspberry Pi:n käynnistyksellä tarvitaan asettaa siihen muutamia perusasetuksia. Rapsberry Pi 3:sen käyttöjärjestelmäksi valitaan Raspberry Pi OS (64bit) (KUVA 4). Tähän tarvitaan internetyhteys, jotta voidaan ladata ja sitten asentaa Raspberry Pi OS-käyttöjärjestelmä. Seuraavaksi asetetaan maa suomi, kieli suomi ja aikavyöhyke Helsinki. Valitaan myös kohdasta, jossa lukee ”use english language”. Valitaan kieleksi englanti, koska internetistä saatava materiaali on lähes aina

englanniksi, joten se helpottaa työtä hieman. Sitten valitaan ”next”. Seuraavaksi asetetaan Raspberry Pi:lle käyttäjä ja käyttäjälle salasana. Käyttäjän nimeksi laitetaan ”pi”. Tämän jälkeen Raspberry Pi alkaa lataamaan ja päivittämään käyttöjärjestelmää. Uudelleen käynnistyksen jälkeen Raspberry Pi ja Raspberry Pi OS-käyttöjärjestelmä ovat valmiita käytettäviksi erilaisiin projekteihin.



KUVA 4. Raspberry Pi:n ensimmäinen käynnistys

4.4 Raspberry Pi:n konfigurointi

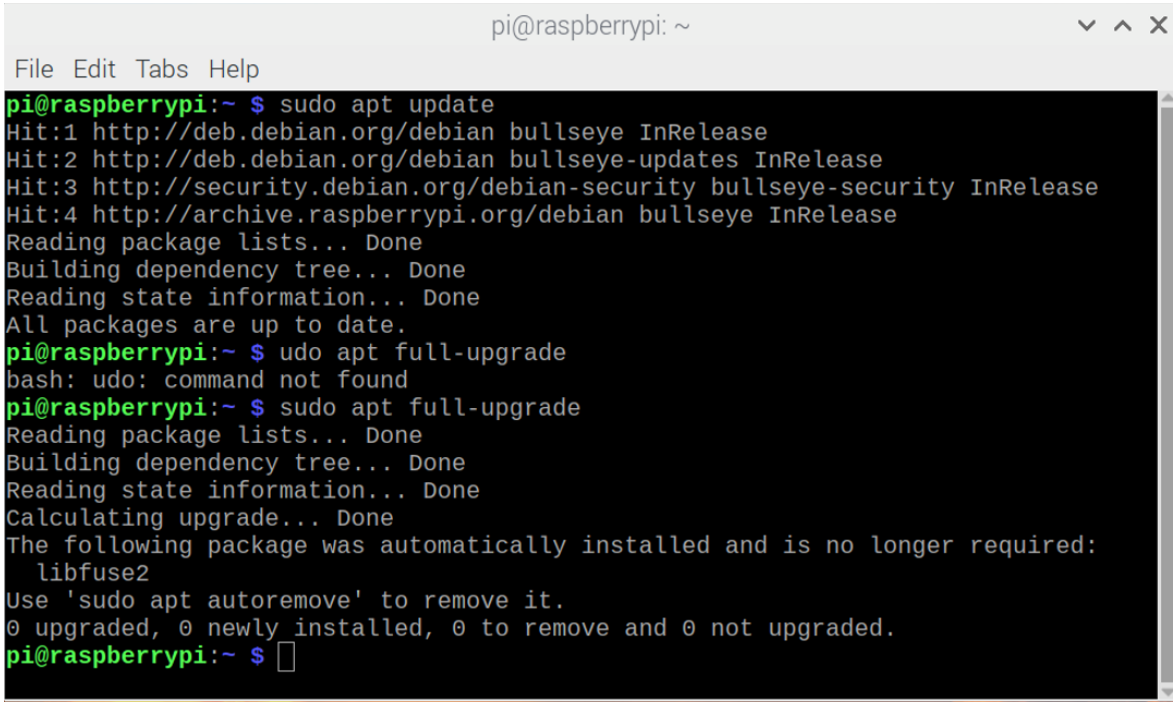
Raspberry Pi OS-käyttöjärjestelmässä on vasemmalla alhaalla ohjelma, jota työssä tullaan käyttämään usein. Ohjelman nimi on terminal. Terminaali-ohjelmalla voidaan helposti antaa komentoja ja konfiguroida Raspberry Pi:tä vain komentoriä käyttämällä. Avataan terminaaliohjelma ja ensimmäiseksi ladataan ja asennetaan päivitykset Raspberry Pi:lle. Tämä tapahtuu kirjoittamalla terminaaliohjelmaan seuraavasti:

```
$ sudo apt update
```

Seuraavaksi kirjoitetaan terminaaliohjelmaan komento:

```
$ sudo apt full-upgrade
```

Tämä komento asentaa kaikki päivitykset Raspberry Pi:lle (KUVA 5). Tämä komento on hyvä suorittaa muutaman kuukauden välein, jotta Raspberry Pi:n turvallisuus ja toimintakyky olisi mahdollisimman hyvällä tasolla.



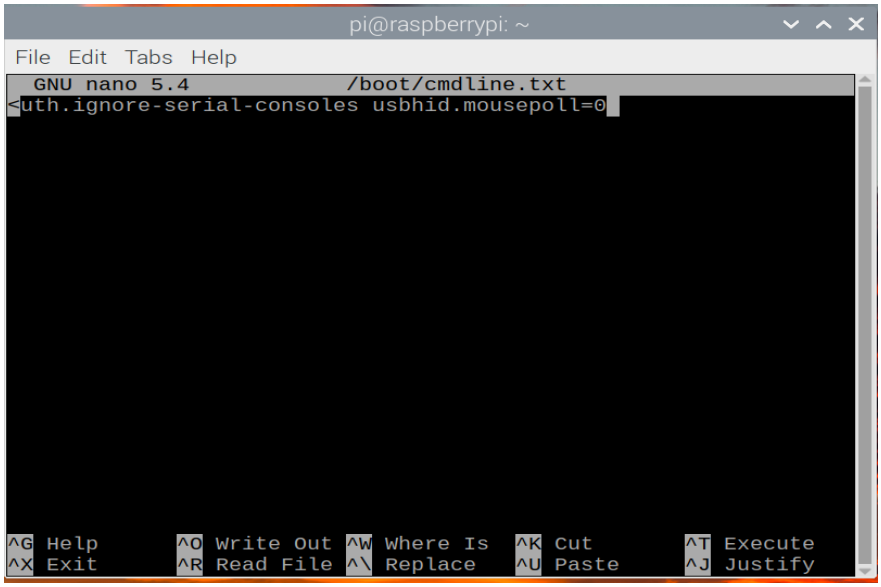
KUVA 5. päivitykset asennettu

4.4.1 Hiiren konfigurointi

Monesti Raspberry Pi:llä hiiren pollausnopeus on kovin hidas. Hiiren pollausnopeus tarkoittaa nopeutta, jolla hiiri kommunikoi tietokoneen kanssa. Mitä suurempi pollausnopeus, sitä enemmän kuormaa tulee keskusytimelle. Laitetaan seuraava komento terminaaliohjelmaan:

```
$ sudo nano /boot/cmdline.txt
```

Tämä komento avaa tekstitiedoston jonne kirjoitetaan **usbhid.mousepoll=0** (KUVA 6). Tallennetaan muokattu tiedosto painikkeilla **Control + O** ja käynnistetään uudelleen Raspberry Pi.



KUVA 6. cmdline.txt tiedosto

Työssä vaihdetaan näppäimistö Suomen kielelle, jotta näppäimistöä olisi helpompi käyttää. Tämä tapahtuu menemällä aloituspainikkeelle ja valitsemalla Preferences > Keyboard and Mouse. Täältä valitaan Keyboard Layout ja Finnish.

4.4.2 Käyttäjän luominen Raspberry Pi:llä

Seuraavaksi luodaan Raspberry Pi:lle uusi käyttäjä ja annetaan tälle pääkäyttäjän oikeudet. Nämä tapahtuvat komennoilla:

```
$ sudo adduser admin
$ sudo adduser admin sudo
```

Tämän jälkeen voidaan Raspberry Pi:lle kirjautua toisella käyttäjällä nimeltä admin. Raspberry Pi:n asetuksista voidaan myös valita, että mille käyttäjälle kirjaudutaan automaattisesti käynnistyksessä.

4.4.3 Tietohakemiston muuttaminen

Tietohakemiston muuttaminen parantaa tietoturvaa Raspberry Pi:ssä. Raspberry Pi:ssä on perusasetuksena tapa tallentaa kaikki tieto kotikansioon. Työssä laitetaan kaikki ohjelmien data erityiselle kansiolle. Tämän ansiosta datakansiota on helpompi siirtää eri asemien välillä. Datakansio luodaan terminaaliohjelmalla komennolla:

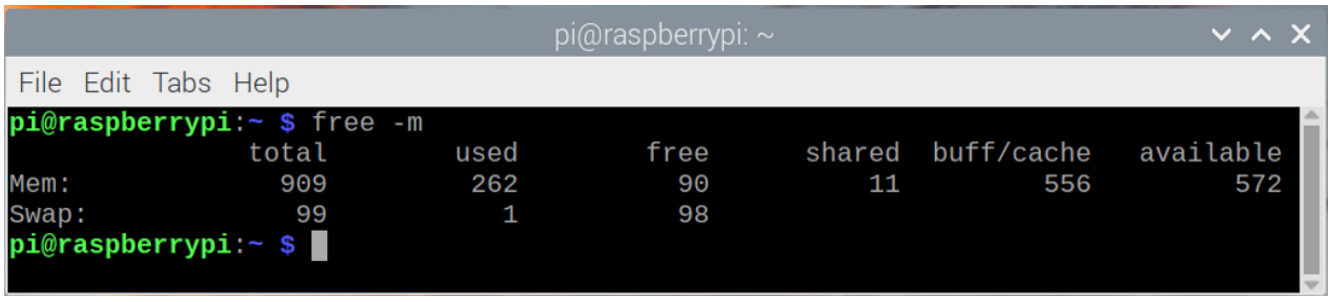
```
$ sudo mkdir  /data
$ sudo chown pi:pi  /data
```

4.4.4 Heittovaihtotiedoston tarkastaminen

Tietokone käyttää hajasaantimuistia (eng. Random Access Memory) ohjelmien ajamiseen. Jos hajasaantimuistia ei ole riittävästi tietokoneen ohjelmien ajamiseen, se käyttää heittovaihtotiedostoa lisämuistina ohjelmiensa suorittamiseen. Näin tietokone voi käyttää enemmän muistia, kuin siinä on hajasaantimuistia asettamalla siihen heittovaihtotiedoston. Heittovaihtotiedosto on osa massamuistia. Heittovaihtotiedosto on hitaampi, kuin hajasaantimuisti, mutta sillä saadaan aikaiseksi stabiilimpi järjestelmä, jos heittovaihtotiedosto on tarpeeksi suuri. Työssä käytettävään Raspberry Pi:hin riittää 1–2 gigatavun heittovaihtotiedosto. Tämä heittovaihtotiedoston koko tuo tarpeeksi stabiiliutta Raspberry Pi:lle, jotta vältytään hidastumisilta ja järjestelmän kaatumisilta. Tarkastan ensiksi, kuinka suuri on Raspberry Pi:n heittovaihtotiedosto ennen sen suurentamista. Tämä tapahtuu kirjoittamalla terminaaliohjelmaan:

```
$ free -m
```

Kuvasta nähdään, että heittovaihtotiedoston koko on 99MB ja käytettävissä 98MB (KUVA 7).



KUVA 7. heittovaihtotiedoston tarkastus

Ensiksi lopetetaan heittovaihtotiedoston käyttäminen, jotta siihen tekemät muutokset toimisivat mahdollisimman hyvin. Heittovaihtotiedoston lopetus tapahtuu terminaaliohjelmassa komennolla:

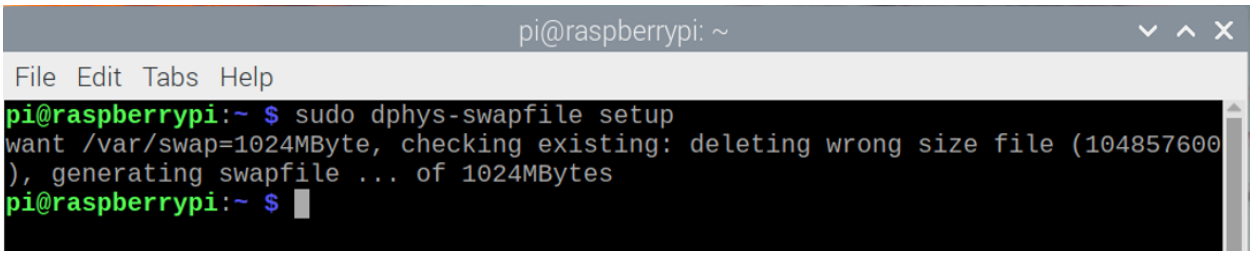
```
$ sudo dphys-swapfile swapoff
```

Heittovaihtotiedoston koon muutos tapahtuu avaamalla dphys-swap tiedosto terminaaliohjelmalla komennolla:

```
$ sudo nano /etc/dphys-swapfile
```

Terminaalissa näytettävässä tiedostossa lukee **CONF_SWAPSIZE=100**. Tämä tarkoittaa, kuinka monta megatavua on heittovaihtotiedoston koko. Vaihdetaan tähän **CONF_SWAPSIZE=1024**. Tallennetaan muutos painamalla **control + O.** Luodaan uusi heittovaihtotiedosto (KUVA 8) komennolla:

```
$ sudo dphys-swapfile setup
```



KUVA 8. heittovaihtotiedoston generointi

Seuraavaksi aloitetaan uuden heittovaihtotiedoston käyttäminen komennolla:

```
$ sudo dphys-swapfile swapon
```

Tämän jälkeen käynnistetään uudelleen Raspberry Pi, jotta uusi heittovaihtotiedosto saadaan käyttöön.

4.5 Raspberry Pi:n tietoturva

Raspberry Pi:llä luotu täyssolmu on näkyvillä internetissä. Tämän takia Raspberry Pi:n tietoturva on tärkeä osa täyssolmua. On tärkeä estää luvattomat pääsyt internetistä tai langattomista yhteyksistä Raspberry Pi:lle. Raspberry Pi:n salasana, säännölliset päivittämiset, palomuurin asennus ja käyttämättömien yhteysteiden poistaminen luovat hyvän tietoturvan Raspberrylle Pi:lle. Aloitetaan Raspberry Pi:n tietoturva konfiguraatio palomuurin asentamisesta.

4.5.1 Palomuurin asentaminen

Palomuuri kontrolloi, mitä dataa voi lähettää Raspberry Pi:lle ulkoa ja mitä dataa ohjelmat voivat lähettää ulos. Asennan Raspberry Pi:lle UFW:n. UFW eli ”mutkaton palomuuri” (eng. Uncomplicated Firewall) on helppokäyttöinen ohjelma netfilter-palomuurin hallintaan. Netfilter on Linux-ytmisessä toimiva järjestelmä verkkopakettien, osoitteenmuunnoksien ja porttimuunnoksien konfigurointiin. UFW:llä voidaan konfiguroida palomuuria komentorivin avulla. (Anicas & Heidi 2015.) Ladataan UFW terminaaliohjelmassa komennolla:

```
$ sudo apt install ufw
```

Tämän jälkeen poistetaan ulkopuoliset yhteydet ja sallitaan lähtevät yhteydet Raspberry Pi:ltä. Tämä tehdään terminaaliohjelmassa komennolla:

```
$ sudo ufw default deny incoming
$ sudo ufw default allow outgoing
```

Poistetaan vielä UFW:n kirjaukset. UFW kirjaa palomuuriviestit järjestelmälokiin. Tätä ei tarvita tässä työssä, joten poistetaan kirjaukset terminaaliohjelmassa komennolla:

```
$ sudo ufw logging off
```

Sitten laitetaan vielä UFW päälle ja asetetaan, että UFW käynnistyy aina automaattisesti, kun käynnistetään uudelleen Raspberry Pi. Nämä tehdään terminaaliohjelmassa komennolla:

```
$ sudo ufw enable
$ sudo systemcl enable ufw
```

Tarkastetaan vielä, että UFW status on active eli käynnissä terminaaliohjelmassa komennolla:

```
$ sudo ufw status
```

Seuraavaksi käynnistetään uudelleen Raspberry Pi, jotta saadaan uudet asetukset päälle.

4.5.2 Langattomien yhteyksien poistaminen

Raspberry Pi:ssä on valmiina Wifi -ja Bluethooth-yhteydet sisään rakennettu. Työssä ei tarvita näistä kumpaakaan, joten tietoturvallisuussyistä seuraavaksi poistetaan Wifillä ja Bluethoothilla pääseminen Raspberry Pi:lle. Halutaan, että Raspberry Pi:hin pääsee käyttämään ainoastaan näppäimistöllä ja hiirellä. Tämä tuo merkittävästi lisää turvallisuutta täyssolmuun. Poistetaan Bluethooth käytöstä avaamalla config.txt-tiedosto ja kirjoittamalla sinne **dtoverlay=disable-bt**. Wifin käyttö poistetaan kirjoittamalla samaan tiedostoon **dtoverlay=disable-wifi**. Config.txt-tiedoston avataan terminaaliohjelmassa komennolla:

```
$ sudo nano /boot/config.txt
```

Tallennetaan muutokset **control + O** näppäimillä ja käynnistetään uudelleen Raspberry Pi. Käynnistyksen jälkeen langattomat yhteydet eivät enää löydä Raspberry Pi:tä.

4.5.3 TOR-ohjelman asennus

Toimiessaan bitcointäyssolmu näyttää sen IP-osoitteen kaikille muille solmuille. Tämä tarkoittaa, että palvelut, joilla voidaan etsiä fyysinen paikka IP-osoitteen perusteella, näkyvät myös muille käyttäjille. Ilman TOR-ohjelmaa kaikki muut täyssolmut ja työssä toteutettava täyssolmu havaitaan internetistä. Tämän takia haluan tehdä täyssolmustani mahdollisimman anonyymin, että fyysistä paikkaa ei voida yhdistää sen IP-osoitteeseen. The Onion Router (lyhennettynä TOR) on avoimen lähdekoodin ohjelma, joka mahdollistaa anonyymiset yhteydet internetissä. TOR toimii vapaaehtoisiin välityspalvelimiin tai solmuihin ympärimaailmaa ja niiden yhteyksiin sipulireitityksen avulla. Sipulireititys (eng. onion routing) on tekniikka, jolla välityspalvelimet siirtävät ja salaavat tiedon alkuperän käyttämällä useita eri välityspalvelimia. TOR-verkossa jokainen välityspalvelin tietää ainoastaan sitä aikaisemman välityspalvelimen IP-osoitteen ja seuraavan välityspalvelimen IP-osoitteen, johon paketti lähetetään. (Vitosinschi 2016, 11–18.)

Ladataan ja asennetaan TOR-ohjelma Raspberry Pi:lle komennolla:

```
$ sudo apt install tor
```

TOR-ohjelman asentamisen jälkeen asetetaan, että bitcoin core -ohjelma käyttää TOR-verkkoa aina, kun bitcoin core -ohjelma käyttää internettiä. Avataan TOR-ohjelman konfiguraatiotiedosto terminaalissa komennolla:

```
$ sudo nano /etc/tor/torrc
```

Tähän tiedostoon kirjoitetaan, että:

ControlPort 9051
CookieAuthentication 1
CookieAuthFileGroupReadable 1

Tallennetaan tiedostomuutokset painikkeella **control + O** ja uudelleen ladataan vielä TOR-ohjelma komennolla:

```
$ sudo systemctl reload tor
```

Asetetaan bitcoin core -ohjelma myöhemmin käyttämään TOR-ohjelmaa internettiin yhdistäessä.

4.6 Bitcoin core -ohjelman asennus

Koko bitcoinlohkoketju on kooltaan vajaa 400 GB. Raspberry Pi 3:sen suoritusteho ei riitä lataamaan tai tarkastamaan koko lohkoketjua, vaan tähän tarvitaan pöytätietokonetta tai kannettavaa tietokonetta. Lohkoketjun lataamisen jälkeen Raspberry Pi 3:sen suoritustehot riittävät ylläpitämään täyssolmua. Ladataan koko lohkoketju pöytätietokoneella ja aikaa tähän kuluu monta päivää. Bitcoin core -ohjelma lataa koko bitcoinhistorian transaktiot kovalevylle, jota tullaan käyttämään Raspberry Pi:llä. Ladataan bitcoin core -ohjelma terminaaliohjelmassa komennolla:

```
$ wget https://bitcoincore.org/bin/bitcoin-core-23.0/bitcoin-23.0-
aarch64-linux-gnu.tar.gz
```

4.6.1 Bitcoin core -ohjelman aitouden tarkastaminen tarkistussummalla

Jotta bitcoinsolmun asennus olisi mahdollisimman turvallisesti toteutettu, halutaan tarkistaa, onko ladattu tiedosto ehjä. Tähän käytetään tarkistussummaa (eng. checksum). Tarkistussumma on koodi, jolla tietotekniikassa voidaan havaita, onko data ehjää vai onko siinä tallennusvirheitä. Tarkistussumma muodostaa algoritmilla datasta, jota tarkastetaan, vakiokokoisen luvun. Tämän jälkeen voidaan uudelleen laskea tarkistussumma ja tarkastaa, että tulos on sama, jotta data olisi ehjää. Seuraavaksi ladataan lista kryptografisesta tarkistussummasta komennolla:

```
$ wget https://bitcoincore.org/bin/bitcoin-core-23.0/SHA256SUMS
```

Ja ladataan myös allekirjoitukset, joilla todistetaan tarkistussumman oikeellisuuden komennolla:

```
$ wget https://bitcoincore.org/bin/bitcoin-core-23.0/SHA256SUMS.asc
```

Seuraavaksi halutaan tarkistaa, että vertauksellinen tarkistussumma tiedostossa **SHA256SUMS** on sama, kuin, mitä olen laskenut. Tämä tapahtuu terminaaliohjelmassa komennolla:

```
$ sha256sum --ignore-missing --check SHA256SUMS
```

Jotta bitcoin core -ohjelma olisi mahdollisimman turvallisesti asennettu, halutaan varmistaa, että se on aito. Jotta bitcoin core -ohjelman aitous voidaan varmistaa, täytyy ensiksi importoida julkinen avain, jota vastaa julkaistu aito bitcoin -core ohjelmaa vastaava allekirjoitus. Bitcoin core -ohjelma allekirjoitetaan jokaisessa julkaisussaan monen luotettavan henkilön osalta. Luotettava henkilö on esimerkiksi bitcoinohjelmistokehittäjä. Importoidaan luotettavien bitcoinkehittäjien julkiset avaimet komennolla:

$ wget https://raw.githubusercontent.com/bitcoin/bitcoin/master/contrib/builder-keys/keys.txt

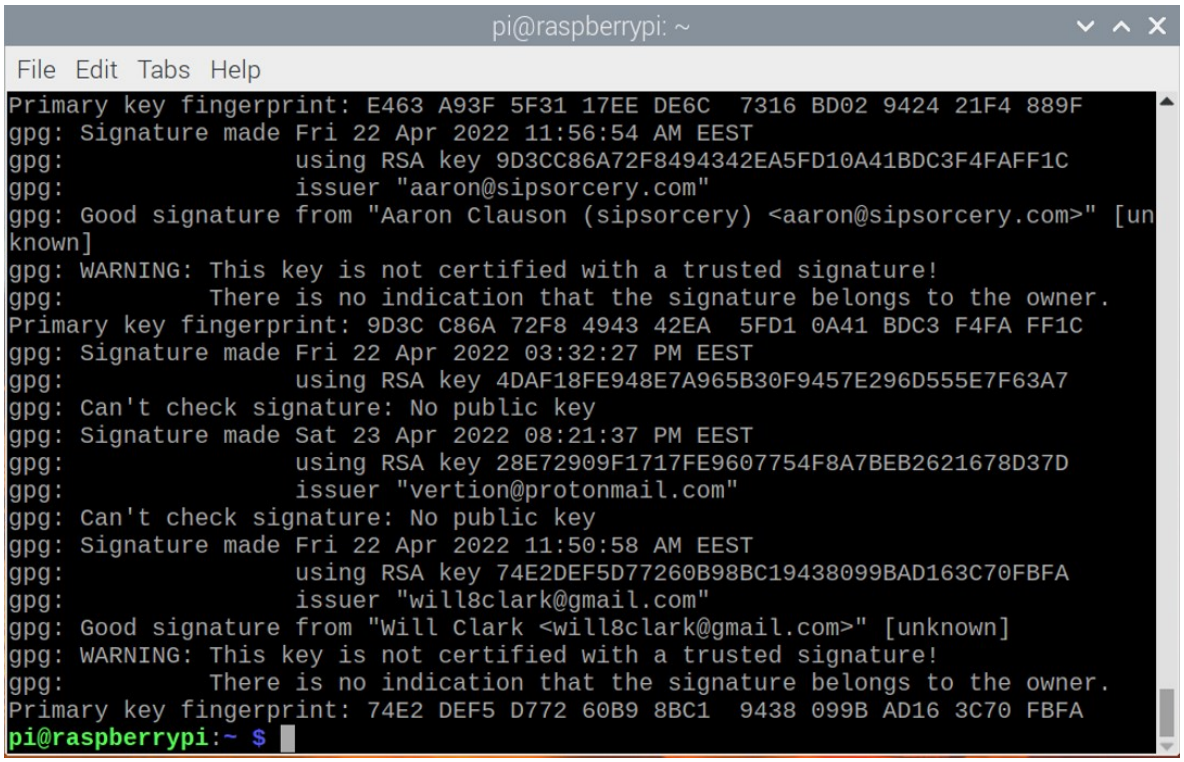
Ja seuraavaksi listataan importoidun tiedoston sisällöt komennolla:

```
$ while read fingerprint keyholder_name; do gpg --keyserver
hkps://keyserver.ubuntu.com --recv-keys ${fingerprint}; done <
./keys.txt
```

Seuraavaksi varmistetaan, että ladattu tarkistussummatiedosto on kryptografisesti allekirjoitettu julkisilla bitcoinkehittäjien avaimilla. Tarkistetaan jokainen avain erikseen. Turvalliseen bitcoin core -ohjelman asennukseen riittää, että muutamilla avaimilla lukee ”Good signature from”. Tämä tarkoittaa sitä, että tarkistussummatiedosto on allekirjoitettu bitcoinkehittäjän avaimella. Tehdään tämä komennolla:

```
$ gpg --verify SHA256SUMS.asc
```

Kuvasta (KUVA 9) näkyy, että muutamissa kohdissa lukee, että ”Good signature from” ja kehittäjän nimi. Tämä tarkoittaa, että bitcoin core -ohjelma, jonka latasin, voidaan todeta turvalliseksi ja aidoksi bitcoin core -ohjelmaksi allekirjoitustarkastuksen jälkeen.



KUVA 9. allekirjoitusten tarkastaminen tarkistussummalla

Seuraavaksi ladataan bitcoin core -ohjelma Raspberry Pi:lle komennolla:

```
$ tar -xvf bitcoin-23.0-aarch64-linux-gnu.tar.gz
```

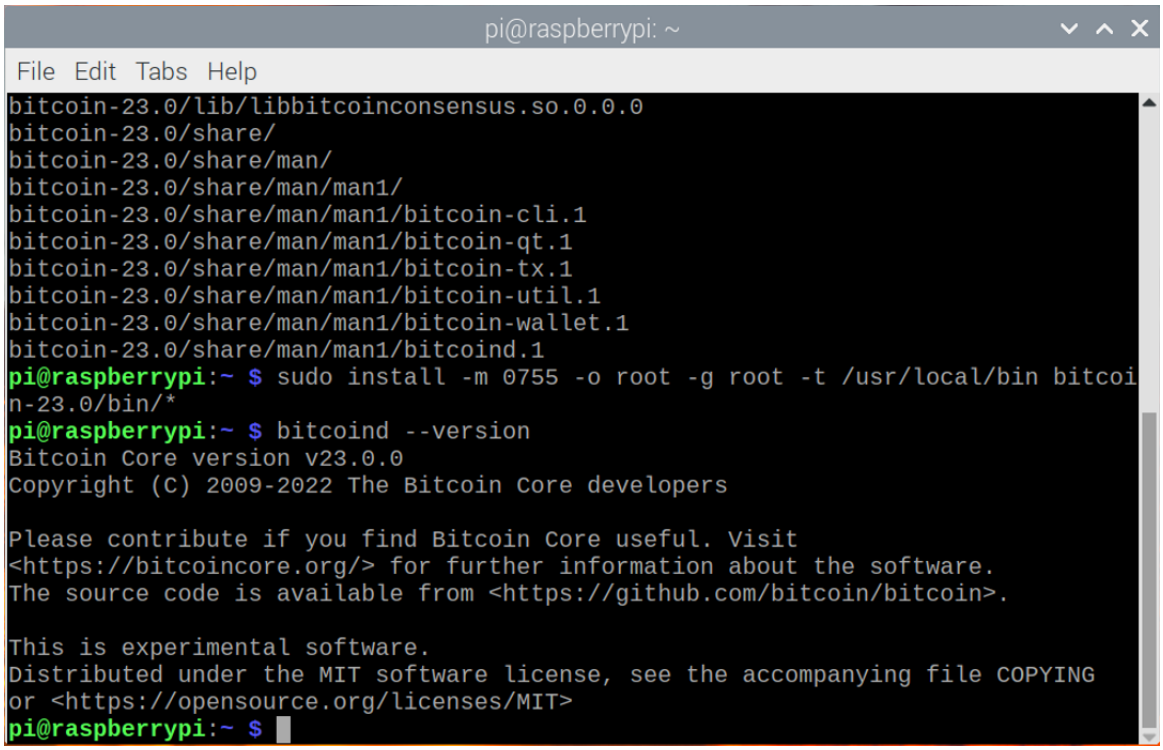
Sen jälkeen asennetaan bitcoin core -ohjelma komennolla:

```
$ sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-
23.0/bin/*
```

Bitcoin core -ohjelman asennuksen jälkeen tarkistetaan bitcoin core -ohjelman versio komentorivikomennolla:

```
$ bitcoind --version
```

Kuvasta (KUVA 10) nähdään, että bitcoin core -ohjelman versio on 23.0.0, mikä tarkoittaa uusinta versiota tällä hetkellä.



KUVA 10. bitcoin core -ohjelman asennus ja version tarkastaminen.

4.6.2 Bitcoinkäyttäjän luominen

Luodaan uusi käyttäjä Raspberry Pi:lle. Tällä käyttäjällä ajetaan täyssolmua. Uudelle käyttäjälle ei anneta järjestelmänylläpitäjän oikeuksia, jotta kukaan ei voi vaihtaa järjestelmäasetuksia, kun täyssolmu on käynnissä. Luodaan uusi käyttäjä nimeltä ”bitcoin” seuraavalla komentorivikomennolla:

```
$ sudo adduser –gecos ”” –disabled-password bitcoin
```

Annetaan käyttäjän ”bitcoin” oikeuden käyttää TOR-ohjelmaa lisäämällä sen ”debian-tor” ryhmään seuraavalla komentorivikomennolla:

```
$ sudo adduser bitcoin debian-tor
```

4.6.3 Tiedostokansion luominen

Asetetaan bitcoin core -ohjelma, niin, että se ei käytä oletuskansiota datan tallentamiseen, koska oletus kansio on käyttäjän kotikansiossa. Luodaan uusi datakansio komentorivikomennolla:

```
$ mkdir /data/bitcoin
$ sudo chown bitcoin:bitcoin /data/bitcoin
```

Seuraavaksi vaihdetaan käyttäjä ”bitcoin” komennolla:

```
$ sudo su - bitcoin
```

Sitten luodaan linkki .bitcoin, jolla päästään tähän kansioon. Tämä tapahtuu komentorivikomennolla:

```
$ ln -s /data/bitcoin /home/bitcoin/.bitcoin
```

4.6.4 Bitcoin-datakansion luominen

Vaihdetaan ”bitcoin” käyttäjän datakansio käyttäjän kotikansiosta paikkaan /data ja linkitetään sen kotikansiolle komentorivikomennolla:

```
$ mkdir /data/bitcoin
$ sudo chown bitcoin:bitcoin /data/bitcoin
```

Vaihdetaan käyttäjään ”bitcoin” komentorivikomennolla:

```
$ sudo su - bitcoin
```

Tehdään linkki .bitcoin, joka osoittaa /data kansioon komentorivikomennolla:

```
$ ln -s /data/bitcoin /home/bitcoin/.bitcoin
```

4.6.5 Luodaan käyttöoikeudet

Jotta salasanaa ja käyttäjänimeä ei tarvitse säilyttää asennustiedostossa tekstinä, joka ei ole tietoturvallisesti suositeltavaa, pitää luoda salasanasta tiiviste, jota bitcoin core -ohjelma voi verrata sitä tiivisteeseen, jossa säilytetään tiedostoa. Tämä mahdollistaa tavan käyttää bitcoin core -ohjelmaa niin, että bitcoin core -ohjelma kysyy salasanaa ja saa aidoilta sovelluksilta salasanan, mutta salasanaa ei säilytetä tekstimuodossa tietokoneella. Bitcoin core -ohjelmassa tulee mukana yksinkertainen Python-ohjelma, jolla voidaan luoda tiiviste. Aloitetaan vaihtamalla käyttäjään ”bitcoin” kirjoittamalla terminaaliin

```
$ cd .bitcoin
```

Seuraavaksi käyttäjällä ”bitcoin” ladataan RPCauth-ohjelma komentorivikomennolla:

```
$ wget
https://raw.githubusercontent.com/bitcoin/bitcoin/master/share/rpcau
th/rpcauth.py
```

Ajetaan scripti Python 3-tulkilla, asetetaan käyttäjä ”bitcoin” ja sen salasana komennolla:

```
$ python3 rpcauth.py raspibolt Salasana
```

Tästä saadaan RPCauth-tekstirivi, joka kopioidaan bitcoinkonfiguraatiotiedostoon seuraavanlaisesti. Avataan ”bitcoin” käyttäjänä konfigurointitiedosto komentorivikomennolla:

```
$ nano /home/bitcoin/.bitcoin/bitcoin.conf
```

Tähän konfiguraatiotiedostoon kirjoitetaan kaikki seuraavat asiat:

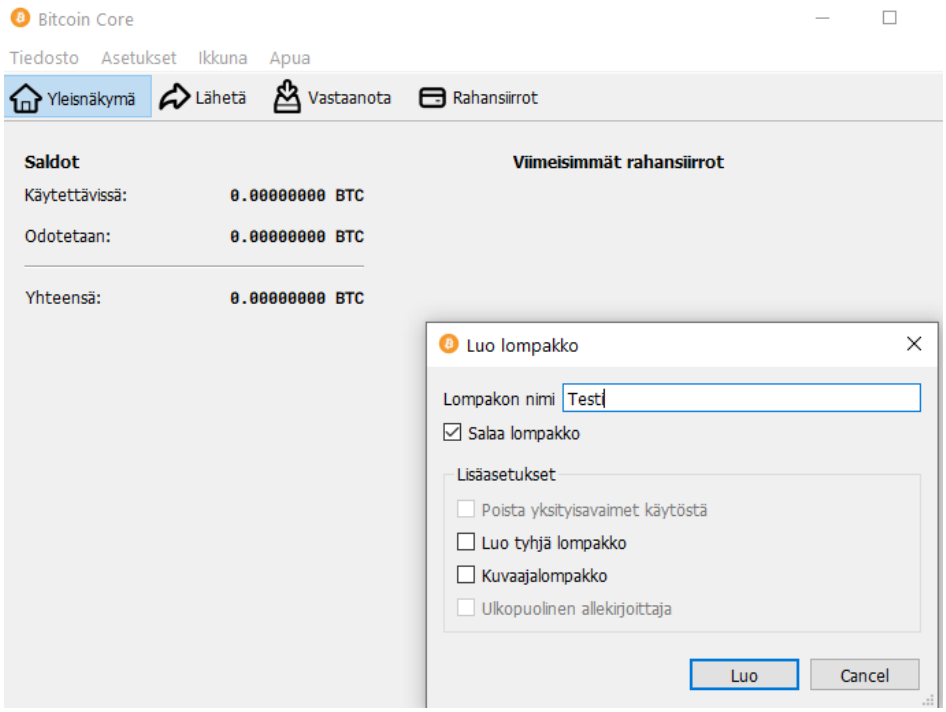
rpcauth=”tähän lisätään kopioimasi rpcauth-rivi”
server=1
txindex=1
listen=1
listenonion=1
proxy=127.0.0.1:9050
bind=127.0.0.1
zmqpubrawblock=tcp://127.0.0.1:28332
zmqpubrawtx=tcp://127.0.0.1:28333
whitelist=download@127.0.0.1
dbcache=6000

Seuraavaksi tallennetaan tiedosto ja asetetaan, että vain käyttäjä ”bitcoin” ja sen ryhmän jäsenet voivat lukea tätä tiedostoa. Komentorivikomento tälle on:

```
$ chmod 640 /home/bitcoin/.bitcoin/bitcoin.conf
```

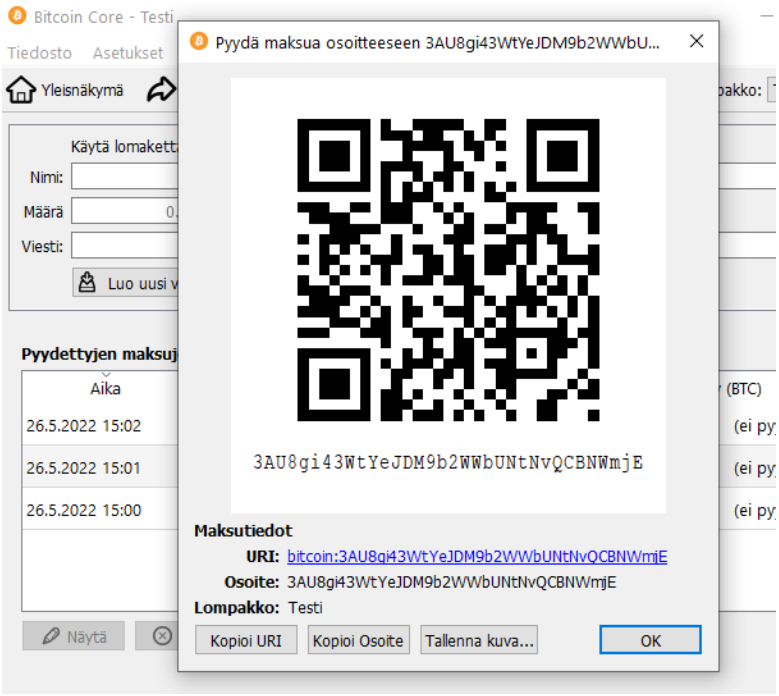
4.7 Bitcoinlompakon luominen ja testaus

Testataan luodun bitcointäyssolmun lompakon toimivuutta. Ensimmäiseksi luodaan uusi lompakko. Tämä tapahtuu painamalla Tiedosto > Luo lompakko… ja asetetaan lompakon nimeksi ”Testi” (KUVA 11).



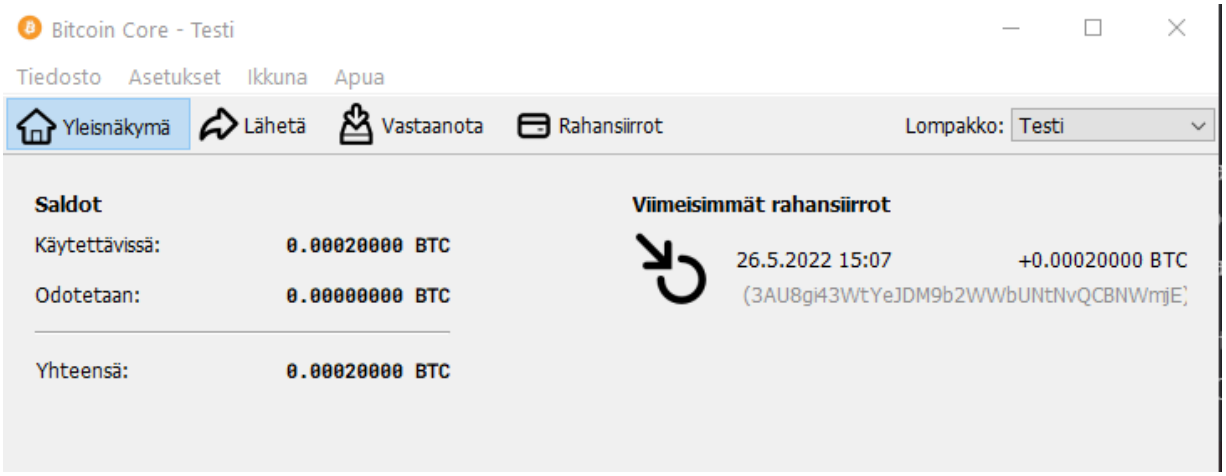
KUVA 11. bitcoinlompakon luominen

Tämän jälkeen laitetaan tunnuslause ja painetaan ”Luo”-näppäintä. Seuraavaksi luodaan uudelle lompakolle maksuosoite. Tämä tarkoittaa osoitetta, jonne voidaan vastaanottaa bitcoinia. Tämä tapahtuu menemällä kohtaan ”Vastaanota” ja painamalla ”Luo uusi vastaanotto-osoite”. Sitten painetaan ”Kopioi Osoite” ja lähetetään tähän osoitteeseen vähän bitcoinia toimivuuden testaamisen takia (KUVA 12).



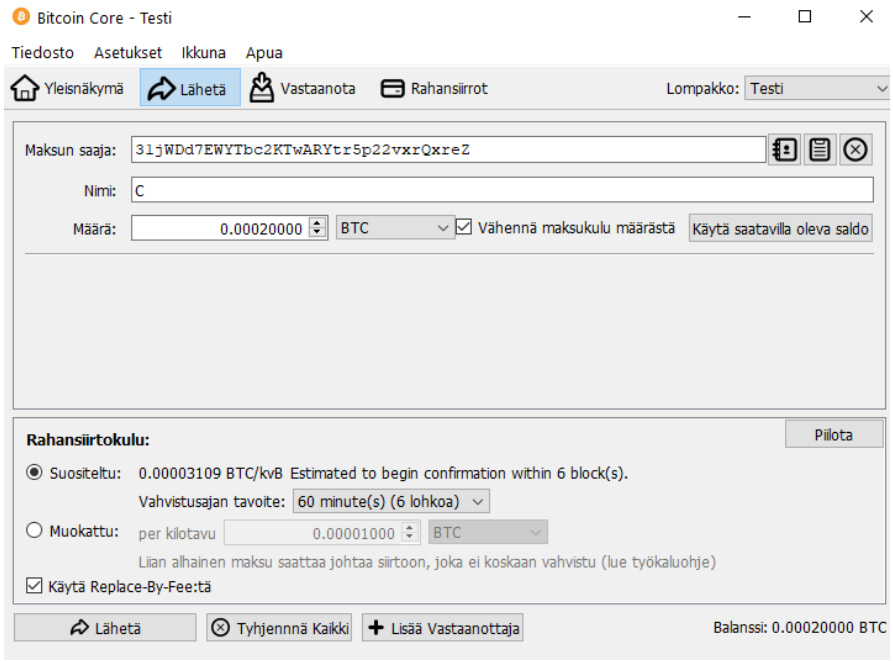
KUVA 12. vastaanotto-osoitteen luominen.

Hetken kuluttua bitcointransaktio on saapunut uudelle lompakolle (KUVA 13).



KUVA 13. bitcointransaktio on saapunut.

Lähetetään 0,002 BTC vielä takaisin, jotta sekin onnistuu. Valitaan ”Lähetä” ja lisätään kohtaan ”Maksun saaja” bitcoin osoite, jonne halutaan bitcoinia lähetettävän. Seuraavaksi lisätään nimi ja bitcoin määrä. Rahansiirtokulu jätetään ”suositeltu”. Tämä rahansiirtokulu on se kulu, kuinka paljon bitcoinia annetaan louhijoille palkinnoksi tämän maksun suorittamisesta. Mitä nopeampi transaktio sitä suurempi on transaktiokulu. Valitaan ”Suositeltu” ja painetaan ”Lähetä” (KUVA 14.



KUVA 14. bitcointransaktio

Bitcoinlompakko kysyy vielä tunnuslausetta, joka asetettiin lompakkoon aikaisemmin. Tunnuslauseen hyväksyttyä bitcoinlompakko varmistaa vielä, että transaktio on oikein ja sitten painetaan ”Allekirjoita ja lähetä”. Bitcoin transaktioiden saavuttua voidaan varmistaa, että bitcoinlompakko toimii oikein. Työssä on luotu turvallinen bitcointäyssolmu, joka toimii myös bitcoinlompakkona.

5 POHDINTA JA YHTEENVETO

Olen ollut jo monta vuotta koulussani ollessa erittäin kiinnostunut kryptovaluutoista, ja ajatukseni oli jo ammattikorkeakoulun alusta asti tehdä opinnäytetyö kryptovaluuttaan liittyen. Aiemmin ajattelin tehdä opinnäytetyön muista kryptovaluutoista esimerkiksi ethreumista tai cardanosta. Valitsin kuitenkin aiheeksi bitcoinin, sillä se on ensimmäinen kryptovaluutta, josta kiinnostuin ja opin koulussa ollessani bitcoinista sen, että bitcoinmaksupalvelujärjestelmä eroaa käytännössä vain yhden asian takia tavallisesta maksupalvelujärjestelmästä, tämä ero on luottaminen kolmanteen osapuoleen transaktioissa. Bitcoinmaksupalvelujärjestelmää käytettäessä luotetaan ainoastaan matematiikkaan ja algoritmiin, jolla bitcointransaktiot turvataan. Tämä tieto lisättynä siihen, että bitcointäyssolmuja on vain noin 10 tuhatta kappaletta maailmalla, halusin luoda projektin täyssolmun luomisesta. Bitcoin ja kryptovaluutat ovat tällä hetkellä hyvin ajankohtainen aihe, kun puhutaan taloudesta, keskuspankkien elvytyksestä tai inflaatiosta.

Opinnäytetyön tavoite onnistui mielestäni oikein hyvin. Yritin tehdä työstä mahdollisimman yksinkertaisen kuvien avulla, mitä olisi helppo seurata. Olisin voinut tehdä opinnäytetyön kokonaan englanniksi, jotta kaikkia englanninkielisiä sanoja ei olisi tarvinnut kääntää suomeksi. Monelle sanalle ei ole suoranaista käännöstä, joten se lisäsi hieman haastetta.

Opin paljon bitcoinmaksupalvelujärjestelmästä ja Linux-pohjaisten laitteiden toimivuudesta ja konfiguroinneista.

Lähteet

Anicas, M & Heidi, E. 2015. UFW Essentials: Common Firewall Rules and Commands. Saatavissa: https://www.digitalocean.com/community/tutorials/ufw-essentials-common-firewall-rules-and-commands. Viitattu 7.5.2022

Antonopoulos, A. 2017. Mastering Bitcoin Programming The Open Blockchain. Toinen painos. O'Reilly. Viitattu 1.2.2022.

Chohan, U. W. 2022. A History of Bitcoin. Saatavissa: https://ssrn.com/abstract=3047875 or http://dx.doi.org/10.2139/ssrn.3047875. Viitattu 5.3.2022.

Croman, K. Decker, C. Eyal, I. Gencer, A. Juels, A. Kosba, A. Miller, A. Saxena, P. Shi, E. G¨un, E. Sirer. Song, D. Wattenhofer, R. 2016. On Scaling Decentralized Blockchains. Saatavissa: https://www.comp.nus.edu.sg/~prateeks/papers/Bitcoin-scaling.pdf Viitattu 22.4.2022

Nakamoto, S. 2009. Bitcoin: A Peer-to-Peer Electronic Cash System
Saatavissa: https://bitcoin.org/bitcoin.pdf. Viitattu 6.3.2022

Pool Distribution. 2022. Saatavissa: https://btc.com/stats/pool. Viitattu 26.5.2022

Poon, J. & Dryja, T. 2016. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Saatavissa: https://lightning.network/lightning-network-paper.pdf. Viitattu 20.2.2022

Raspberry Pi Foundation. 2021. Saatavissa: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/. Viitattu 22.4.2022

Ghimire, S. 2019. Analysis of Bitcoin Cryptocurrency and Its Mining Techniques. UNLV Theses, Dissertations, Professional Papers, and Capstones. 3603. Saatavissa: http://dx.doi.org/10.34917/15778438. Viitattu 26.4.2022.

Taylor, D. 2018. Why Linux is better than Windows or macOS for security. Saatavissa: https://www.computerworld.com/article/3252823/why-linux-is-better-than-windows-or-macos-for-security.html. Viitattu 26.5.2022

Vitosinschi, A. 2016. Protecting Privacy Using TOR. Turun ammattikorkeakoulu. Degree Programme in Information Technology. Bachelor's thesis. Saatavissa: https://urn.fi/URN:NBN:fi:amk-2016062813397. Viitattu 4.5.2022