Opinnäytetyö (AMK)

Tietoliikenne

Ohjelmistotuotanto

2012

Jani Gustafsson

KÄYTTÖJÄRJESTELMÄOSION JA KIINTOLEVYN OSIOIDEN SALAUS

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietoliikenne | Ohjelmistotuotanto

Huhtikuu 2012 | 44 sivua

Tiina Ferm

Jani Gustafsson

KÄYTTÖJÄRJESTELMÄOSION JA KIINTOLEVYN SALAUS

Nykyaikana tieturvaan aletaan jo suhtautua vakavasti, mutta ongelmia esiintyy vielä varsinkin pienemmissä yhtiöissä perinteisen teollisuuden aloilla. Näissä yhtiöissä ei ehkä ole perinteisesti käytetty ehkä paljoa tietotekniikkaa, eikä siellä ole myöskään ammattitaitoista henkilökuntaa tältä alalta.Vaikka kaikilla on jo palomuurit kunnossa ja sähköpostit suojattu, niin harvoin otetaan huomioon suoraa datan varastamista tai häviämistä.

Tämän opinnäytetyön tarkoituksena on näyttää miten toteutetaan tietojen ja osioiden salaus siten, että sillä estetään tehokkaasti ulkopuolisten pääsy tietoihin käsiksi, vaikka se olisi heidän hallussaan. Tämä on ehkä tehokkain ja edullisin ratkaisu tilanteisiin, joihin muuten ei oikein voi varautua. Näitä ovat esimerkiksi kannettavan tietokoneen hukkaaminen tai sen joutuminen varastetuksi.

Opinnäytetyöstä ja lähteistä tarkempaa tietoa hakemalla saa selkeän kuvan siitä miten salaus suoritetaan ja mitkä ovat parhaimmat vaihtoehdot missäkin tilanteessa. Lisäksi työstä selviää vaadittavat toimenpiteet ennen kuin itse salaus voidaan aloittaa. Tämä kattaa varmuuskopioinnin, osioinnin ja käynnistyslevyn teot.

Työssä on selvitetty ja toteutettu käyttöjärjestelmäosion sekä ulkoisen kiintolevyn salaus 256 bittisellä AES-salauksella. Hashalgoritmeja ei olla työssä käytetty, mutta ne on käyty tärkeimmiltä kohdiltaan läpi. Lopputuloksena on tehokkaasti salattu käyttöjärjestelmä, johon murtauminen ei ole ilman salasanaa toteutettavissa järkevällä budjetilla ja aikajänteellä. Tämän lisäksi on salattu ulkoinen kiintolevy datan turvallista ylläpitoa varten. Näin estetään dataan käsiksi pääsy vaikka kone olisi käynnissä.

Tulevaisuudessa voisi tehdä selvityksen salattujen osioiden piilotuksesta. Lisäksi voisi selvittää tulevan salausstandardin ja sen soveltaminen. Näiden yhdistelmällä oltaisiin jälleen askelta pidemmällä datan turvallisuuden osalta.

ASIASANAT:

Salaus, kryptografia, turvallisuus, IT, AES, hash, algoritmi

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology| Software Engineering

April 2012 | 44 pages

Tiina Ferm

Jani Gustafsson

ENCRYPTION OF THE OPERATING SYSTEM PARTITION AND HARD DRIVE

Nowadays IT-security is taken seriously already, but there are some traditional fields of industry that are not so familiar with IT. Especially the smaller companies often lack the professional staff for building a secure IT-environment. They usually have a firewall and secured email, but what they do not take into consideration adequately, is the security of the data itself.

This thesis and the sources of information for more detail give the reader a general picture of what is needed to perform an encryption of data and partitions and to gain the optimal result of it. This is probably the least expensive and most secure way to prevent loss of information in situations like theft or loss of a laptop.

This study also describes the necessary actions, that need to be taken before the final encryption, e.g. partitioning, making bootdisk and backup. Moreover, there are some hints about the optimal solutions in different cases.

The work environment now have an 256 bits AES encrypted systempartition. Also, there is an AES encrypted harddrive, which is intended for storing data safely. There have not been used hashalgorithms, but they have been covered in general. As a result the system is now impossible to hack with a reasonable budget and timeframe. And even if the system is on, the critical data is still safe in its encrypted container.

In the future it would be good to do some research on hiding encrypted partitions and take to consideration the use of new encryption standard. With these methods we would be a step further on the agenda to get our data secured

KEYWORDS:

Encoding, cryptographics, protection, IT, AES, hash, algothm

KUVAT

KUVIOT

KÄYTETYT LYHENTEET

NIST	National Institute of Standards and Technology [1]
AES	Advanced Encryption Standard [2]
NSA	National Security Agency [3]
FIPS	Federal Information Processing Standars Publications [4]
NESSIE	New European Schemes for Signatures, Integrity, and Encryption [5]
MBR	Master Boot Record [6]
IEC	International Electrotechnical Commission [7]
ISO	International Organization for Standardization [8]
RAID	Redundant array of independent disks [9]

1 JOHDANTO

Tämän työn tarkoituksena on tehdä tietokoneesta käyttäjälleen ja/tai yritykselle turvallinen tietyissä poikkeustilanteissa. Näitä tilanteita voivat olla tietokoneen joutuminen varastetuksi tai sen häviäminen. Työssä käytetään TrueCryptiä salaukseen, kiintolevyjen osiointiin EASUS Partition Master 9.1.0 Home Editionia ja varmuuskopiointiin EASUS TODO Backup ohjelmaa. Salausmenetelmänä käytetään Advanced Encryption Standardia (AES). Periaatteessa AES on standardi ja salaussalgoritmi on muunnelma Rijndaelista, mutta nykyään AES on vakiintunut algoritmin nimenä. AES-salausalgoritmi on FIPS (Federal Information Processing Standard) hyväksytty. Tämä standardi on Yhdysvaltojen liittovaltioiden hallituksen tekemä ja hyväksymä. [10]

Työn pääpaino on käyttöjärjestelmäosion sekä kiintolevyjen muiden osioden salauksessa, mutta tässä käydään myös läpi hieman salaukseen läheisesti liittyvien hash-funktioiden toimintaa, varmuuskopiointia, osiointia ja käynnistyslevyjen luontia. Työssä käytetyt menetelmät on tätä työtä varten tehty henkilökohtaiselle tietokoneelle, mutta ne ovat myös yleisessä käytössä erään yhtiön kannettavissa tietokoneissa, koska näissä varkauden tai häviämisen riski on suurin ja arkaluontoisen tiedon olemassaolo todennäköistä.

Käyttöjärjestelmäosion tai tietyn kiintolevyosion salaamisella halutaan estää koneen luvaton käyttö, koska Windows käyttöjärjestelmän MBR:n (Master Boot Record) resetointi onnistuu nykyään käytännössä keneltä tahansa, joka osaa tietokoneen käytön perusteet. Itse salausohjelmana käytetään Truecrypt 7.1a ohjelmaa, joka on kaikkien saatavilla ilmaiseksi osoitteessa www.truecrypt.org. Ohjelman tekijöistä ei ilmeisesti vieläkään ole varmaa tietoa, mutta lähdekoodi on vapaa ja ohjelma on julkaistu TrueCrypt License Version 2.8 alla [11]. Ohjelmassa on käytössä kolme algoritmia salaamiseen, AES, Serpent ja Twofish, sekä muutama näiden yhdistelmä, AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES ja Twofish-Serpent. Tässä työssä käytetään AES-algoritmia. Teoria osuudessa käydään pintapuolisesti läpi algoritmien toi-

mintaperiaatteet, mutta tarkemmin AES:iin voi tutustua lukemalla seuraavan dokumentin: http://www.truecrypt.org/legal/license [12]. Serpentiin voi tutustua tässä dokumentissa: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf [13] ja Twofishiin tässä: http://csrc.nist.gov/archive/aes/round2/comments/20000513-pbora.pdf [14].

2 TEORIAA

Tässä kappaleessa käydään läpi TrueCryptin käyttämiä salausalgoritmeja eli AES, Twofish ja Serpent. Lisäksi kiinnitetään hieman huomiota nykyaikaisten salaustekniikoiden rakenteeseen yleisesti ja käydään läpi TrueCryptin tukemat hash funktiot.

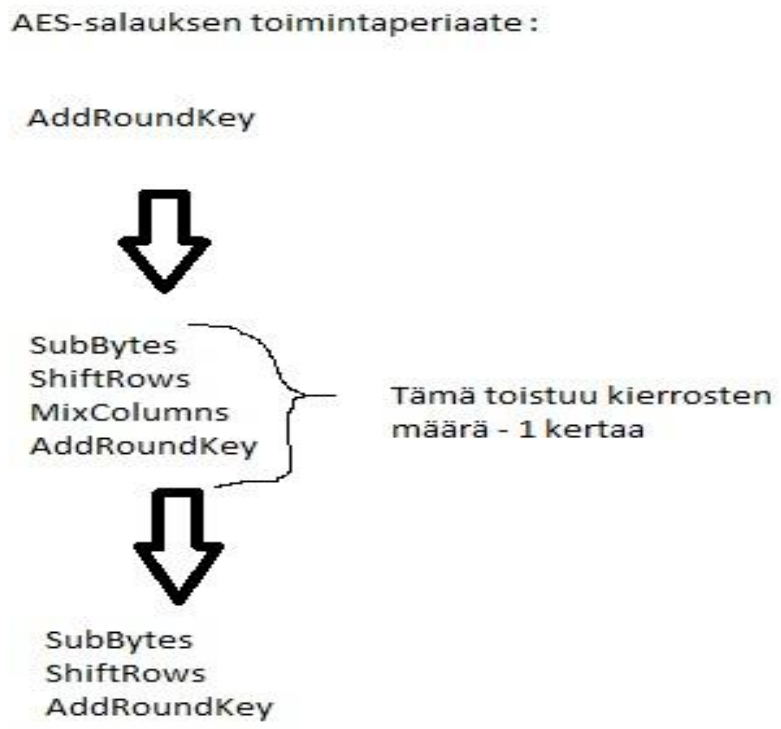
2.1 SALAUS ALGORITMIT

TrueCryptissä käytössä olevat salausmenetelmät AES, Twofish ja Serpent ovat AES kilpailun finalisteja ja kelpaavat Yhdysvaltojen virastojen virallisiksi salausmenetelmiksi. AES-256 aina ylimälle turvallisuus tasolle asti [15].

2.1.1 AES

AES algoritmi on Rijndael-algoritmista tehty muunnelma jonka ovat kehittäneet Joan Daemen ja Vincent Rijmen vuonna 1998. AES poikkeaa hyvin vähän alkuperäisestä Rijndaelista. AES on niin sanottu block cipher eli lohkosalain. [16.] Useimmissa salausalgoritmeissa käytetään Feistelin rakennetta, mutta AES käyttää kolmekerroksista muunnosta, joka koostuu lineaarisesta sekoituskerroksesta, epälineaarisesta kerroksesta ja avaimenlisäyskerroksesta. [7.] AES-salauksessa on kaksi pääkohtaa, kierrosavaimien luonti ja kierrosfunktio. Kierrosfunktiossa on neljä erillistä funktiota: SubBytes, ShiftRows, MixColumns ja AddRoundKey. Kierrosfunktion kierrosten määrä riippuu salausavaimen pituudesta. Se voi olla 128, 192, tai 256 bittiä pitkä ja vastaavasti kierrokset joko 10, 12 tai 14. [7.]

Kierrosavaimien luonti tapahtuu siten, että AES muuttaa salausavaimen avaintaulukoksi, jonka koko on 4*(salauskierrokset +1). AES käsittelee salattavaa dataa 128 bitin lohkoissa, joka on kaksiulotteisessa taulukossa. [12.]

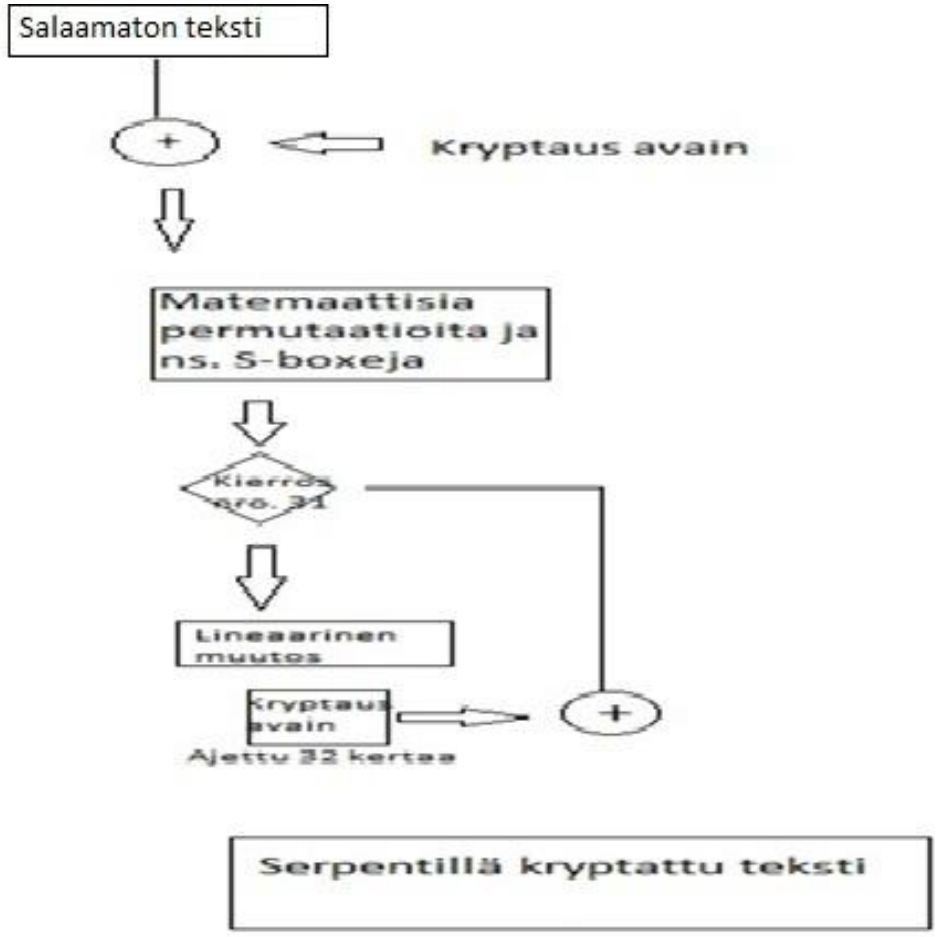


Kuvio 1. AES-algoritmin toiminta [7]

SubBytes etsii niin sanotusta S-Boxista tavuja vastaavat arvot ja kaikki tavut muunnetaan vuorollaan. ShiftRows taas siirtää taulukon tavuja vasemmalle, paitsi ensimmäistä riviä. Kun arvoja siirretään, siirtyy vasemman puoleinen arvo viimeiseksi oikealle. MixColumns taas tunnistaa taulukon sarakkeet nelijäsenisenä polynomina ja tämä polynomi kerrotaan ennalta määritetyllä polynomilla a(x) = {03}x3 + {01}x2 + {01}x + {02} ja tuloksesta otetaan jakojäännös polynomilla $x^4 + 1$. AddRoundKey taas nimensä mukaisesti lisää kierrosavaimen taulukkoon XOR-operaatiolla. Ensimmäinen avaimen lisäys tapahtuu jo ennen kierrosfunktiota. AddRoundKey-fuktio siis käy salauskierrosten määrän +1 kertaa jokaista lohkoa kohti. [12.] AES salausta purettaessa nämä funktiot toteutetaan käänteisesti, InvShiftRows, InvSubBytes, AddRoundKey ja InvMixColumns. [12.]

2.1.2 SERPENT

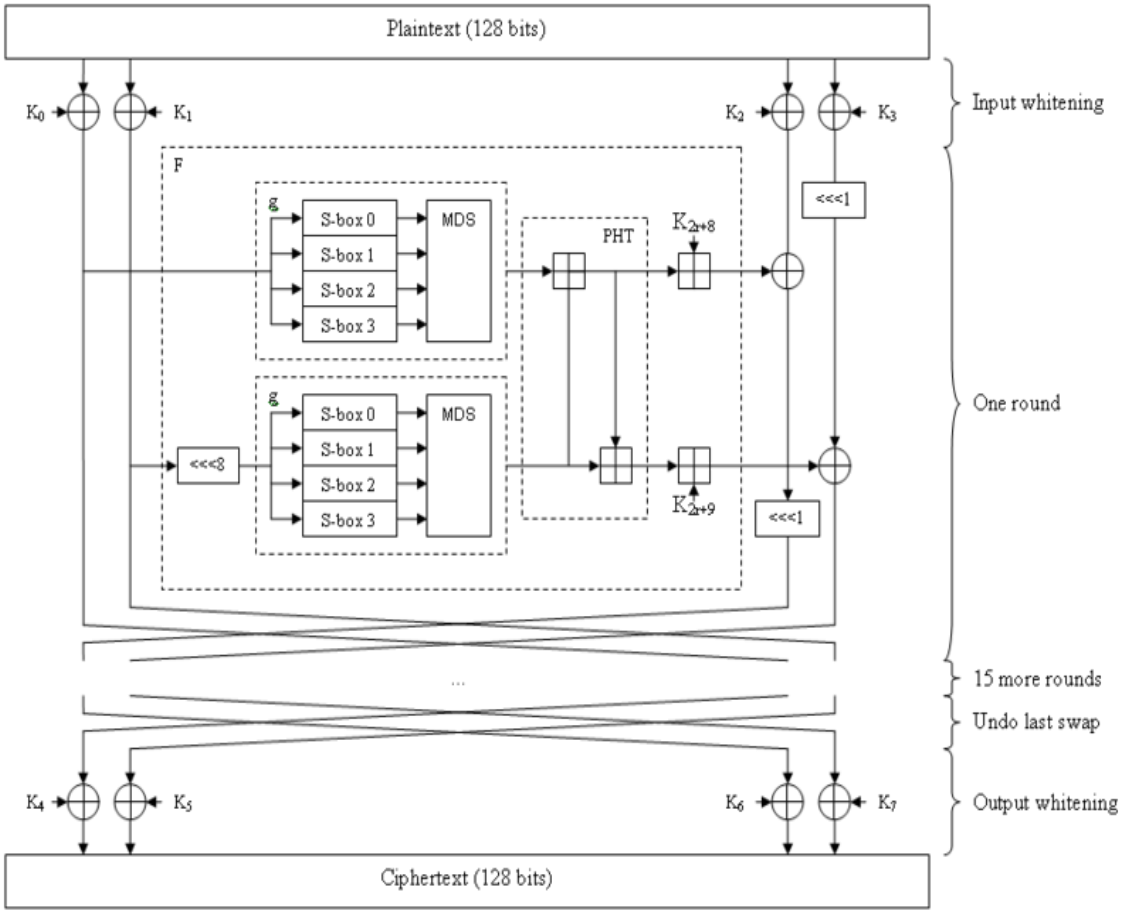
Serpent kuuluu myös niin sanottujen lohkosalainten ryhmään. Sen ovat suunnitelleet Ross Anderson, Lars Knudsen ja Eli Biham ja se julkaistiin vuonna 1998. Se on 128 bittinen lohkosalain, joka käyttää 32 kierrosta samaa algoritmia. Salaus ja salauksen purku ovat toiminnaltaan vastakohtia. Purkutoimenpiteessä toiminnot vain suoritetaan käänteisessä järjestyksessä.



Kuvio 2. Serpent algoritmin toiminta [13]

2.1.3 TWOFISH

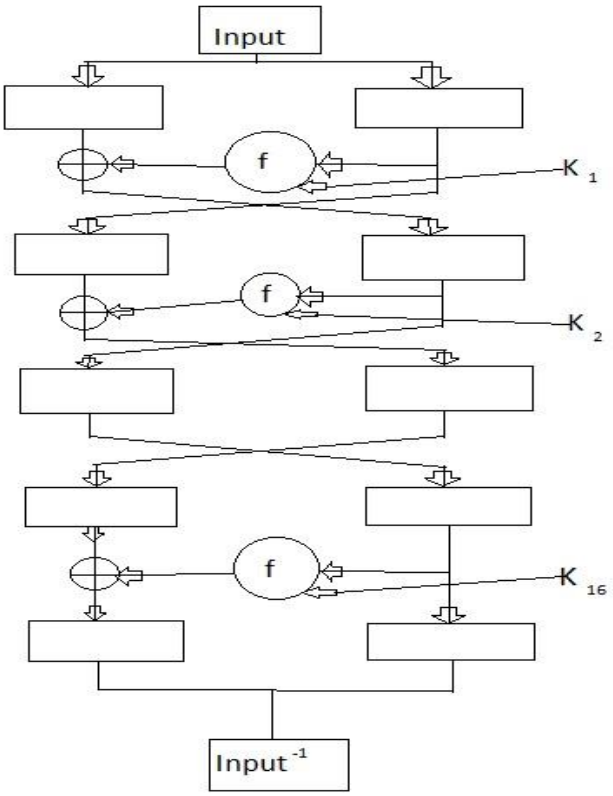
Twofish on myös lohkosalain, jonka ovat suunnitelleet John Kelsey, Bruce Schneier, David Wagner, Niels Ferguson, Chris Hall ja Doug Whiting. Siinä on 128 bitin kokoinen lohko, maksimissaan 256 bittiä pitkä avain ja se käyttää 16:sta kierrosta salaukseen.[14.]



Kuvio 3. Twofish salaimen toiminta. [14]

2.1.4 FEISTELIN RAKENNE

Feistelin funktio on Horst Feistelin luoma ja vuonna 1973 esitelty salausmenetelmä, jonka periaatetta käytetään vielä tänäkin päivänä useimmissa salausalgoritmeissa [15].



Kuvio 4. Feistelin toimintaperiaate. [15]

Feistelin funktiossa sisään tuleva data puolitetaan kahdeksi samankokoiseksi lohkoksi, joita yleisesti kutsutaan left (L) ja right (R). (Kuvio 15). Sitten näitä lohkoja ajetaan toistuvasti algoritmin läpi. Jokaisella kierroksella sekoitusfunktio (f) käytetään oikeanpuoleiseen lohkoon sekä avaimeen ja sekoituksen tulos XOR:taan vasempaan lohkoon. Tämän jälkeen XOR:n tuloksesta tulee uusi oikean puoleinen lohko ja alkuperäisestä oikeasta lohkosta tulee vasen lohko. Tätä toistetaan sitten useampia kierroksia. [15.]

2.1.5 S-BOX

S-boxien yksinkertaistettu toimintaperiaate on se, että niihin tulee sisään bittejä m ja se muuttaa ne tietyllä kaavalla muotoon n. Niiden tarkoitus on siis salata bitit ja pitää salassa alkuperäisten bittien ja ulostulon välinen yhteys, jotta salausta ei voitaisi purkaa tietämättä salasanaa.

2.2 HASH ALGORITMIT

Hash algoritmien tulosta voisi kuvailla tiivisteeksi alkuperäisestä tiedosta eikä siitä voida palauttaa alkuperäisiä tietoja. Enemmänkin on kysymys alkuperäisen tiedon oikeellisuuden tarkistamisesta. Kun luodaan alkuperäisestä tiedosta tiiviste ja tiiviste sekä alkuperäinen tieto lähetetään kohteeseensa, niin perillä voidaan tiivistettä verrata saatuun pakettiin ja todeta, onko se muuttunut alkuperäisestä muodostaan.

Otetaan esimerkiksi seuraava lause: tämä on generoitu sha-1:llä. 160-bittinen tiiviste näyttää seuraavalta: ce82712736651e4b2e8b866fdc5875f7c86f4379. Kun muutetaan viimeinen ä-kirjain a:ksi tulos on seuraava: 18d50ea4769deaf4c95e9a75f4d17814264a82aa. Kuten nähdään, lopputulokset ovat täysin erilaisia. Tämä nimenomaan on hash-funktioiden tarkoitus eli lopputuloksen olisi oltava mahdollisimman ennalta arvaamaton. Pitäisi myös olla mahdotonta olla olemassa kaksi erilaista pakettia joilla on sama tiiviste. Tämä ei ole totta minkään algoritmin kohdalla, mutta kysymys onkin siitä onko sama tiiviste saatavissa järkevässä ajassa ja järkevällä budjetilla vai ei.

Seuraavissa luvuissa käydään muutama hash-algoritmi lyhyesti läpi. Käydään läpi TrueCryptin tarjoamat Whirlpool, SHA-512 ja RIPEMD-160 sekä näiden lisäksi hieman SHA-1:tä ja SHA-3:a.

2.2.1 RIPEMD-160

RIPEMD-160:n ovat tehneet Hans Dobbertin, Antoon Bosselaers ja Bart Preneel Belgiassa. Ensimmäinen versio julkaistiin vuonna 1996 nimeltään RIPEMD.

Algoritmi tuottaa 160 bittistä ulostuloa ja on suunniteltu 32-bittiselle toimintaympäristölle, mutta toimii yhtälailla 64- tai 16-bittisessä ympäristössä. [16.]

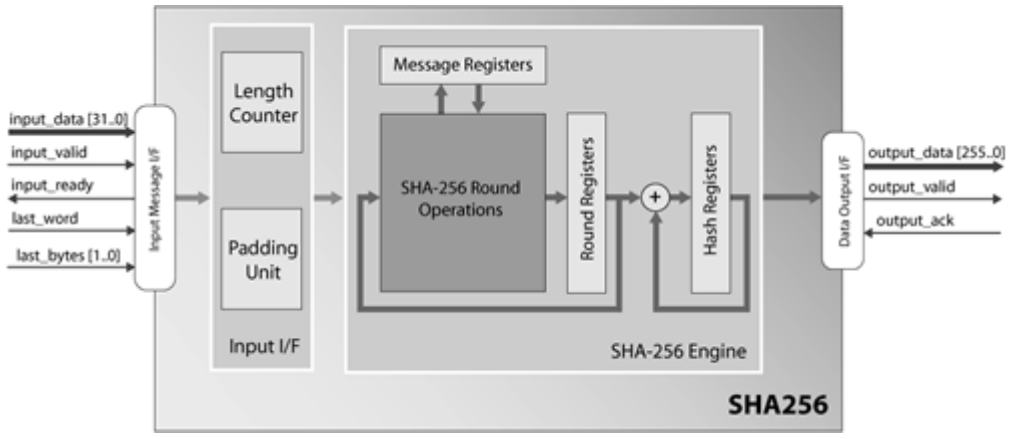
2.2.2 SHA-1

SHA-1 on NSA:n tekemä hash-funktio, joka on kehitetty SHA-0:sta ja julkaistu vuonna 1995. Nämä kaksi ovat hyvin samanlaisia, SHA-1:ssä on vain korjattu SHA-0:n virheet, jotka aiheuttivat sen helpon murtamisen.

SHA-1 on murrettavissa 2^{52} operaatiolla, tämän tekivät Australialaiset tutkijat Cameron McDonald, Philip Hawkes and Josef Pieprzyk vuonna 2009. Tämä tarkoittaa, että 2^{52}:lla operaatiolla tuli törmäys eli saadaan kaksi samanlaista tiivistettä eri alkuperäistiedostoilla. Tämä ei tarkoita sitä, että kuka vain voi väärentää tiivisteen, koska tähän tulokseen päästiin ajamalla supertietokonetta 13 vuorokautta yhtäjaksoisesti. [17.]

2.2.3 SHA-2

SHA-2 nimen alle lasketaan SHA-224, SHA-256, SHA-384 ja SHA-512. SHA-2 on kehitetty SHA-1:stä jälleen NSA:n toimesta. Näistä SHA-256, SHA-384 ja SHA-512 ovat NESSIE:n hyväksymiä.



Kuva 12. SHA-256:n toimintaperiaate [17]

2.2.4 WHIRLPOOL

Tämän on suunnitellut Vincent Rijmen ja Paolo S. L. Barreto. Ensimmäinen versio julkaistiin vuonna 2000 ja tämän hetkinen versio Whirlpool-T on ISO:n ja IEC:n hyväksymä vuonna 2004. Whirlpool on myös yksi kahdesta NESSIE:n hyväksymistä tämän kategorian algoritmeistä.

Algoritmi tuottaa lopputuloksena 512-bittisen tiivisteen alle 2^{256}:n bitin sisääntulolle eli se paloittelee kaiken sisään tulevan datan 2^{256}:n bitin palasiksi ja tekee sille 512 bittiä pitkän tiiviisteen. [19.]

2.2.6 SHA-3

SHA-3 funktioksi valittavasta tavasta on järjestetty NIST:n toimesta kilpailu, jonka pitäisi alkuperäisen suunnitelman mukaan päättyä vuonna 2012. Mukana kisassa on muun muassa Ron Rivestin vetämä ryhmä MD6, Daniel Bernsteinin CubeHash, Bruce Schneierin vetämä ryhmä Skein ja Bob Jenkinsin Maraca. Näistä kylläkin uusimman tiedon mukaan Maraca hylättiin jo ennen ensimmäistä kierrosta NISTin tutkimusten perusteella ja CubeHash tippui putosi kisasta ennen finaalikierrosta. Lisää voi lukea täältä: [20]

3 TIETOKONEEN VALMISTELU

Kun tietokonetta suunnitellaan salattavaksi, on syytä ottaa huomioon muutama asia. Yleensä tietokone toimitetaan yhdellä kiintolevyllä, jossa on kaksi osiota. Toisessa on käyttöjärjestelmä ja toisessa käyttöjärjestelmän palautusosio tehdasasetuksilla. Tämä ei ole optimaalinen järjestely salauksen kannalta, vaan käyttöjärjestelmä tulee eriyttää omaksi osiokseen ja jäljelle jäävä kiintolevytila omaksi osiokseen. Tämä on vähimmäisvaatimus sille, että salauksesta ei koidu ylimääräistä haittaa käyttäjälle ja, että järjestelmän palautustilanteessa toimeenpiteet pystytään suorittamaan ilman ylimääräistä varmuuskopiointia.

3.1 OSIOINNIN SUUNNITTELU

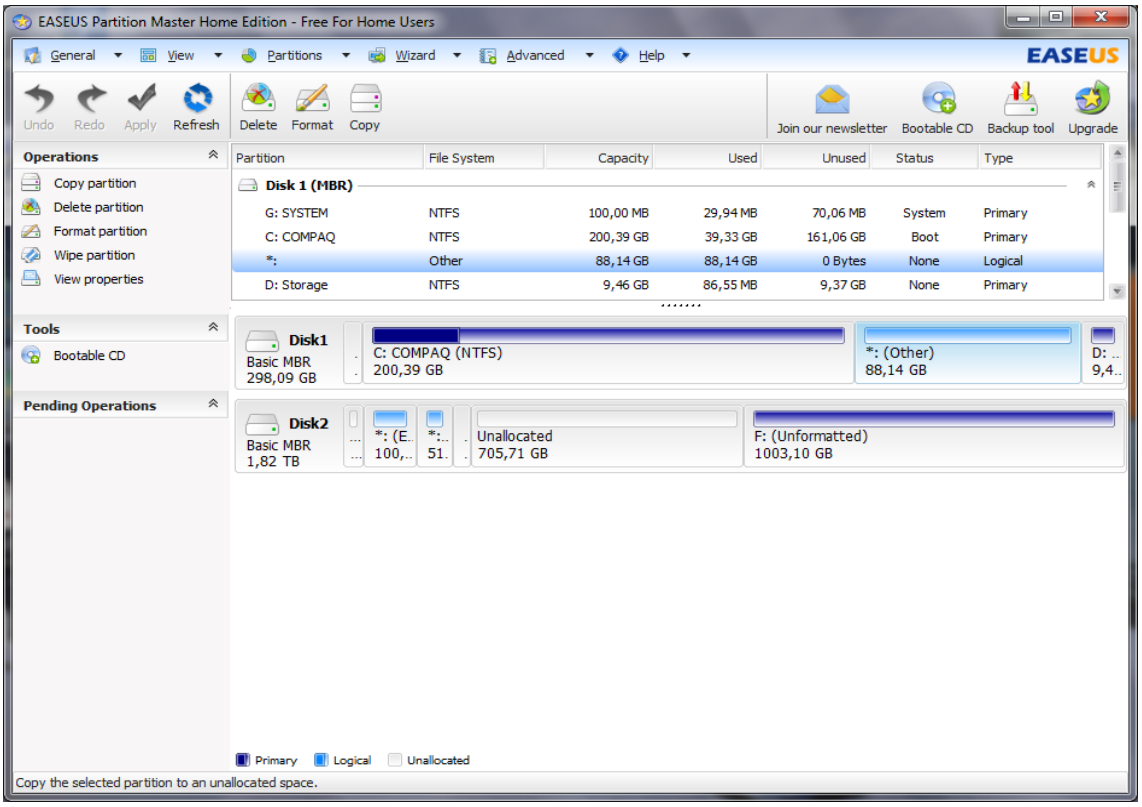
Osioinnin suunnittelu alkaa suunnittelemalla tarvittavat osiot tietokoneelle. Tässä tapauksessa tietokoneessa tulee olemaan yksi 200:n gigatavun osio käyttöjärjestelmälle, toinen noin 10:n gigatavun osio johon voi tallentaa väliaikaisesti tiedostoja sekä kolmas osio, jossa on käyttöjärjestelmän kopio tehdasasetuksilla.

Edellä mainittujen osioiden lisäksi koneessa tullaan käyttämään 2:n teratavun ulkoista kiintolevyä, johon tallennetaan kaikki tarvittava data. Tässä järjestelmässä kymmenen gigatavun osio on vain väliaikainen varasto, josta tiedostot siirretään ulkoiselle kiintolevylle. Tämä järjestely siksi, että ulkoinen kiintolevy tulee olemaan salattu, jolloin sinne tallentaminen on hieman vaivalloisempaa.

Osiointia suunniteltaessa kannattaa ainakin tehdä yksi osio käyttöjärjestelmälle ja toinen tallennettaville tiedostoille. Lisää osioita voi tehdä henkilökohtaisen tarpeen mukaan. Käyttöjärjestelmän ja tallennettavien tiedostojen osioiden eriyttäminen toisistaan tekee varmuuskopioinnista helpompaa sekä vähentää turhaa kopioimista. Myös käyttöjärjestelmän palauttaminen on helpompaa ja järjestely vähentää huomattavasti riskiä menettää tiedostoja.

3.2 OSIOINNIN TOTEUTUS

Osiointiin käytetään EASUS Partition Master 9.1.0 ohjelmaa.



Kuva 1. Tietokoneen osiot

Kuvassa 1 nähdään tietokoneen kiintolevyt ja niiden osiot. Edellä mainittujen osioiden lisäksi kuvassa 1 näkyy noin 88 gigatavun osio. Tämä osio on Backtrack 5 käyttöjärjestelmälle eikä liity tähän työhön. Osiointi suoritetaan pääpiirteittäin edellisessä kappaleessa mainitulla tavalla.

Itse osiointi on erittäin helppoa kyseisellä ohjelmalla ja tarkemmat ohjeet saa osoitteesta http://www.partition-tool.com/.

4 SALAUS

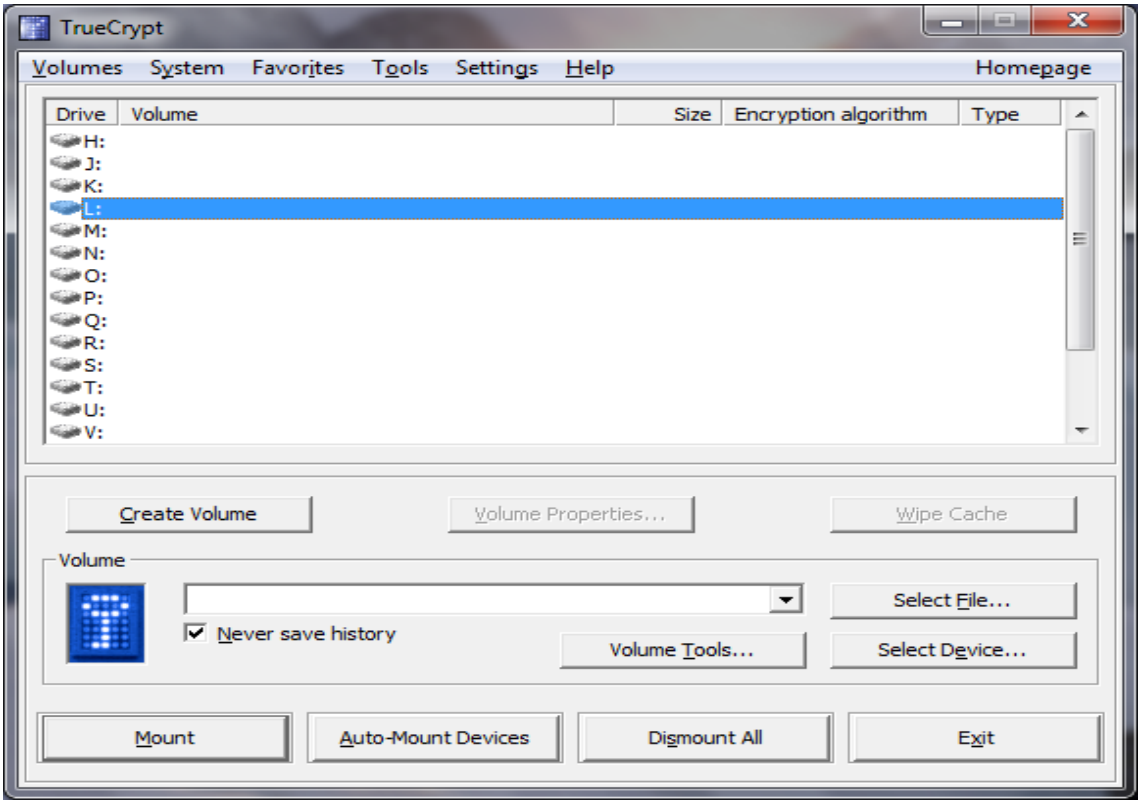
4.1 SALAUKSEN SUUNNITTELU

Ennen kuin salaus aloitetaan, pitää miettiä mitä salataan, miten se salataan ja mitä menetelmää käytetään. Käyttöjärjestelmäosion salauksessa siitä tehdään käynnistyvä eli koneen käynnistyessä ensimmäiseksi tulee Truecryptin luoma bootloader, joka pyytää salasanaa. 10:n gigatavun väliaikaisvarastoa ei salata, koska nimensä mukaisesti siellä ei säilytetä dataa kovin pitkään. Ulkoiselle kiintolevylle tehdään salaus, joka piilottaa aseman ja kun se Truecryptilllä avataan, pyytää se käyttäjältä salasanan ennen kiintolevyn purkua. Näillä toimenpiteillä varmistetaan, että koneelle ei pääse ei haluttuja henkilöitä. Vaikka kone olisi käynnissä, niin kukaan ei haluttu henkilö ei pääse tiedostoihin käsiksi, vaikka koneelle pääsisikin.

Jokaisesta salattavasta käyttöjärjestelmästä tullaan luomaan myös käynnistyslevy. Tämä tehdään sen takia, että jos TrueCryptin bootloader syystä tai toisesta ei toimi enää, niin ilman tätä levyä kyseinen kiintolevy on käyttökelvoton. Yhtiössä, jossa näitä toimenpiteitä tehdään, on suositeltavaa, että nämä levyt ohjeineen laitetaan lukittuun tilaan ja samaan levykoteloon laitetaan myös kyseisen koneen TrueCrypt salasana, koska jos käyttäjä unohtaa salasanan, niin tässäkin tapauksessa kyseisen koneen kiintolevy on käytännössä käyttökelvoton.

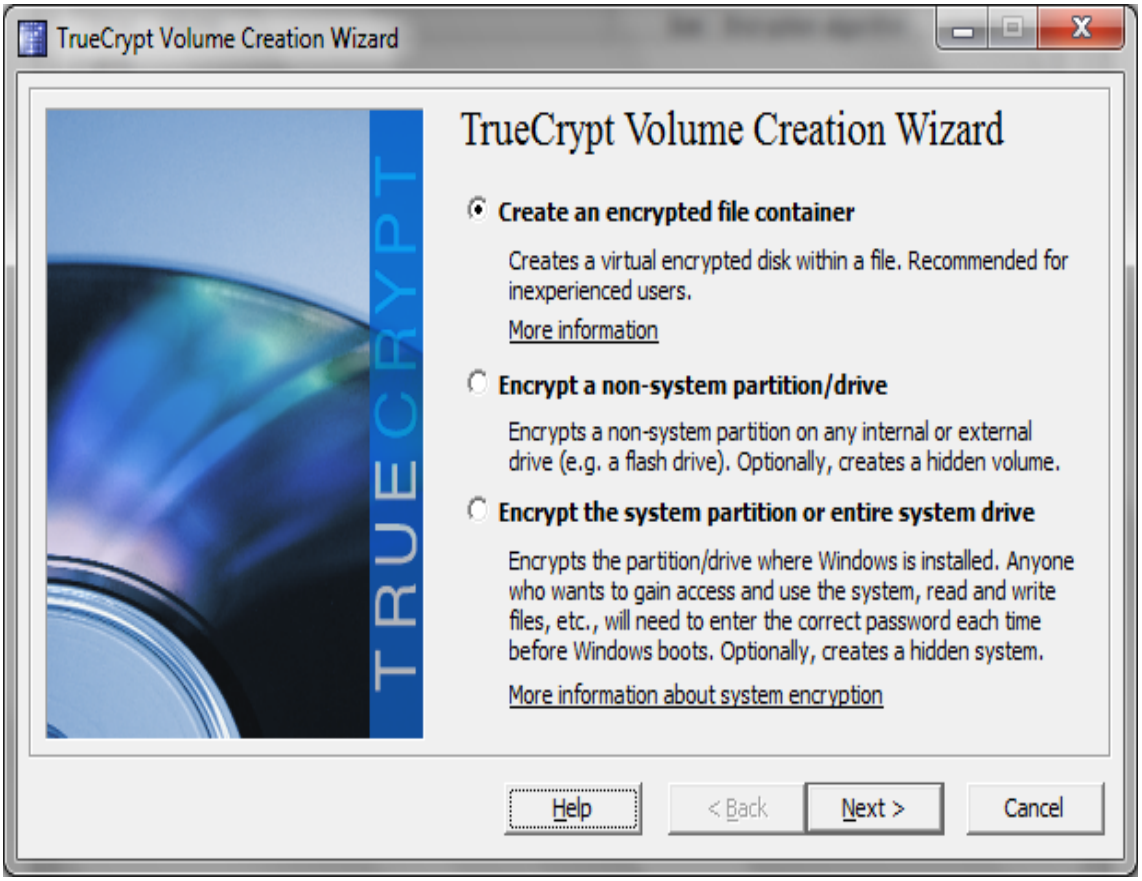
4.2 KÄYTTÖJÄRJESTELMÄOSION SALAUS

Kun TrueCrypt ohjelma avataan aukeaa eteen TrueCryptin alkunäkymä.



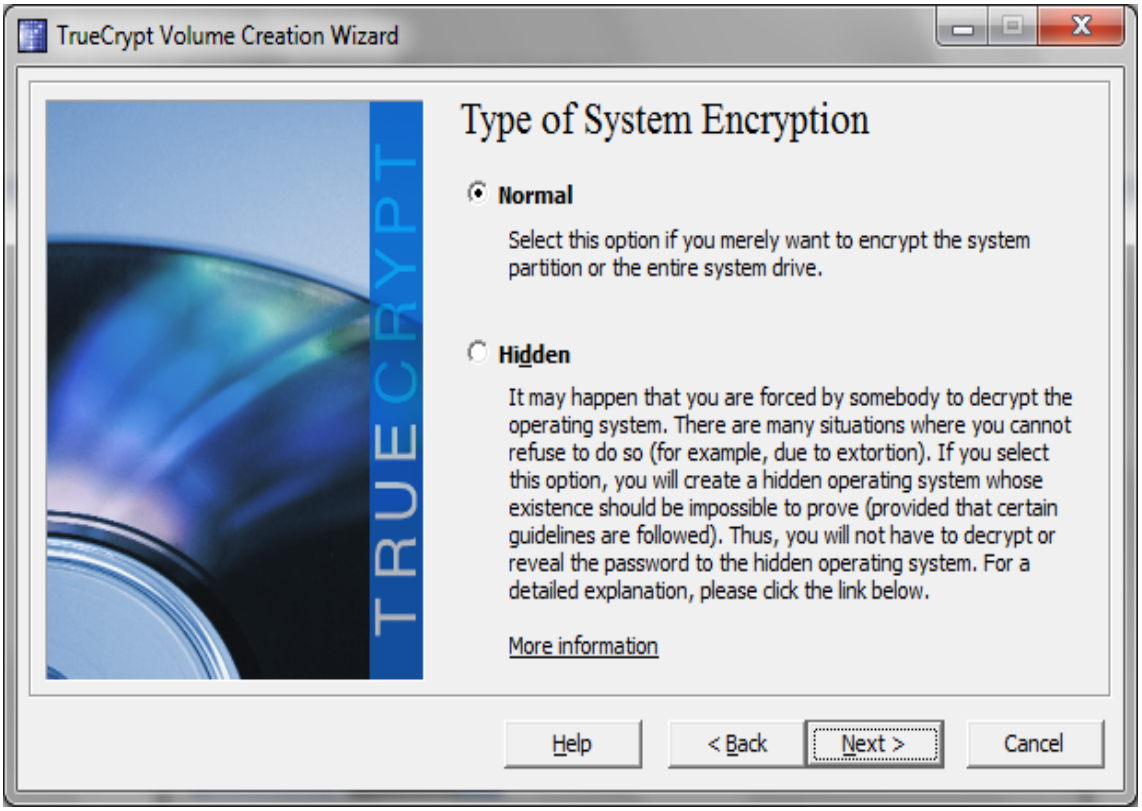
Kuva 2. TrueCryptin aloitusnäkymä

Salaus aloitetaan valitsemalla Create Volume. Tämän jälkeen eteen aukeaa kuvan 3 näkymä.



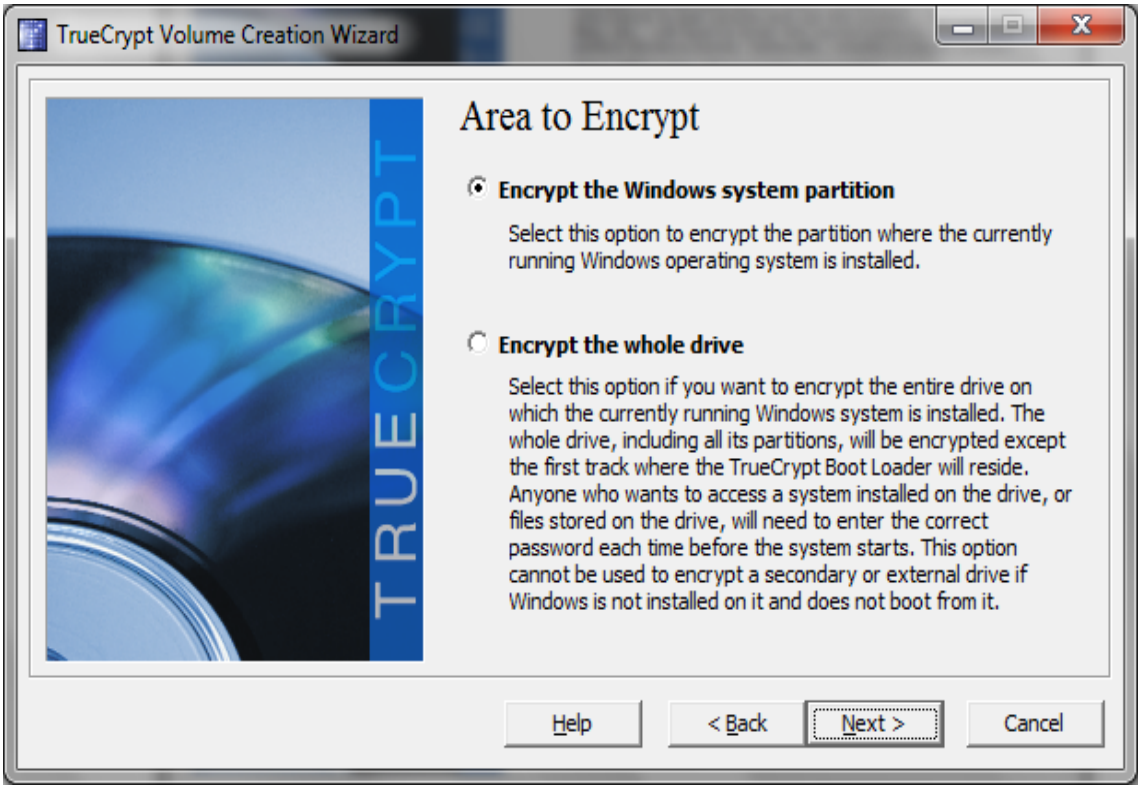
Kuva 3. Mikä osa halutaan salata

Tästä valitaan valitaan Encrypt the system partition or entire system drive, koska haluamme salata käyttöjärjestelmäosion. Tämän jälkeen ohjelma kysyy haluatko tehdä normaalin vai piilotetun järjestelmän.



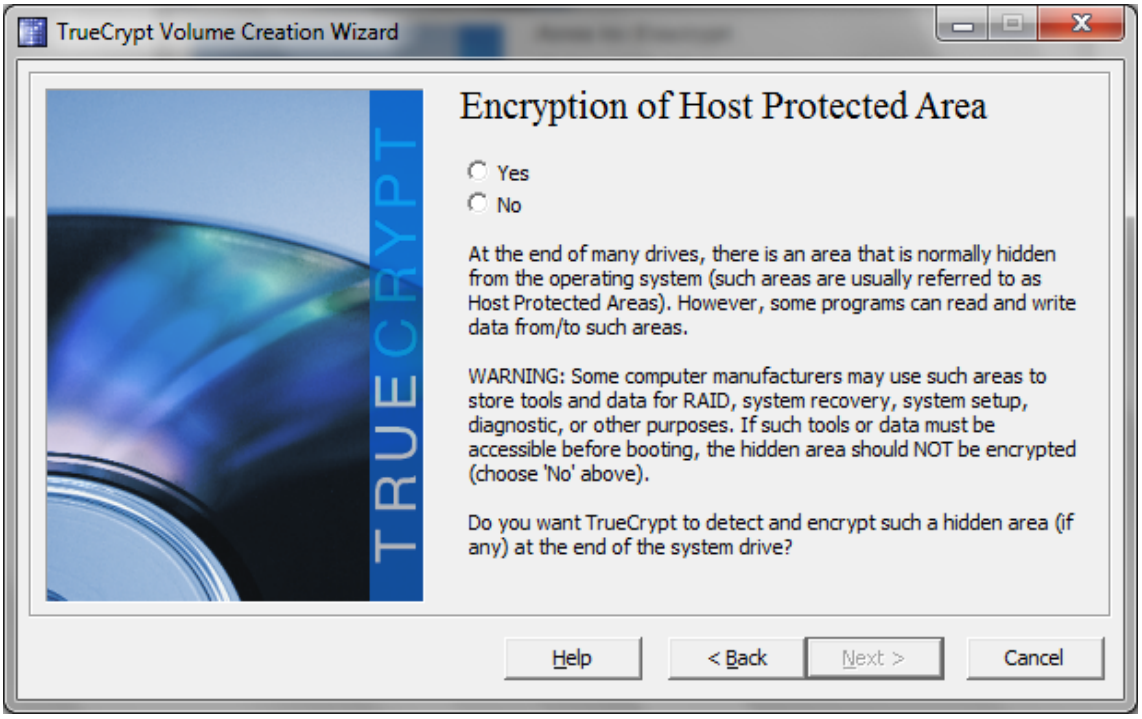
Kuva 4. Normaalin tai piilotetun salauksen luonti

Tässä tapauksessa valitsemme normaalin tavan, koska ei ole tarvetta järjestelmän piilottamiselle. Seuraavaksi ohjelma kysyy salataanko koko kiintolevy, jossa käyttöjärjestelmä sijaitsee vai pelkästään käyttöjärjestelmäosio.



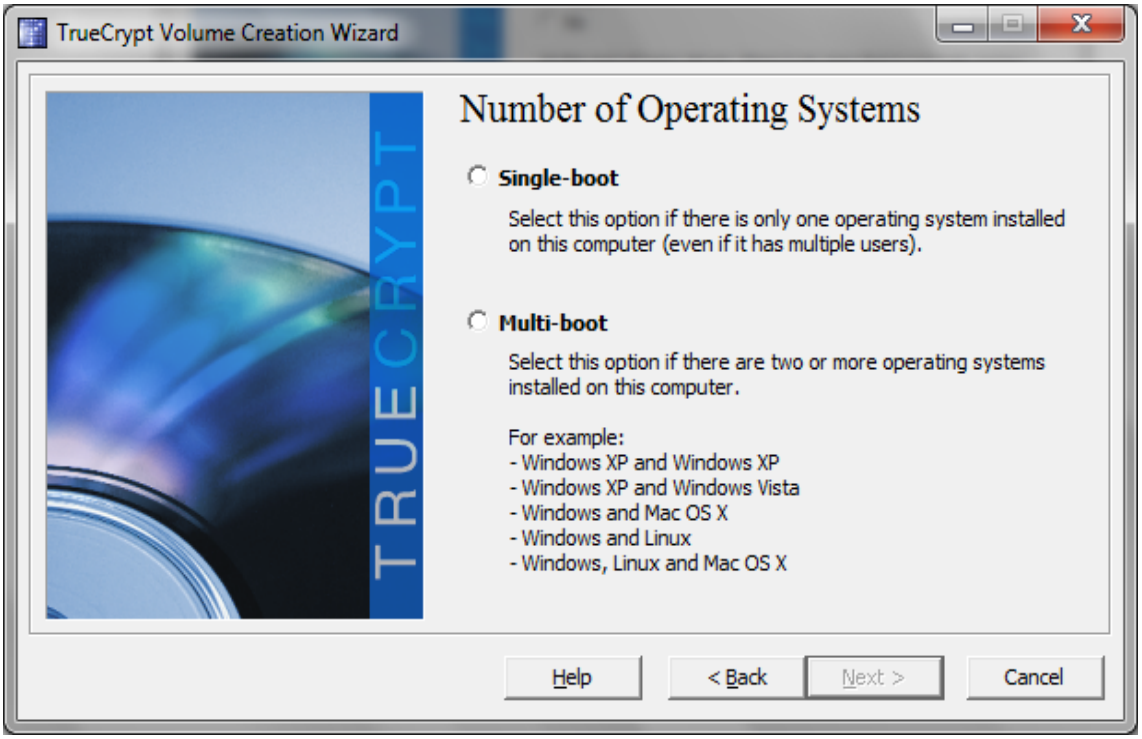
Kuva 5. Järjestelmäosion tai koko kiintolevyn salausvalinta

Tässä tapauksessa valitaan koko kiintolevyn salaus, koska tällä varmistamme, että ketään ei missään tapauksessa pääse kiintolevyyn käsiksi ilman salasanaa. Tämän jälkeen ohjelma kysyy salataanko niin sanottu Host Protected Area. Monilla kiintolevyillä on tällainen alue levyn loppuosassa, joka on piilotettu käyttöjärjestelmältä. Jotkin valmistajat käyttävät tätä osaa tallentaakseen sinne työkaluja RAIDia, järjestelmän palautusta ja muita vastaavia toimenpiteitä varten. Jos bootin yhteydessä tarvitaan jotain näistä työkaluista, niin tätä osiota ei tule salata. Tässä tapauksessa sitä ei salata, koska ei ole tarvetta piilottaa sitä.



Kuva 6. Host Protected Arean mahdollinen salaus

Seuraavaksi valitaan single boot. Vaikka tässä koneessa on kaksi käyttöjärjestelmää, ne eivät toimi niin sanottuna dual boottina, vaan toisen käyttöjärjestelmän valinta vaatii aina muutoksen BIOSissa kiintolevyjen boottaus järjestykseen.



Kuva 7. Single- tai Multi-bootin valinta

Tämän jälkeen valitaan haluttu salausalgoritmi tai niiden yhdistelmä sekä haluttu hash algoritmi, jos käytetään niin sanottua avaintiedostoa. Tässä tapauksessa hash algoritmilla ei ole merkitystä, koska ei tulla käyttämään avaintiedostoa ja algoritmiksi valitaan AES.

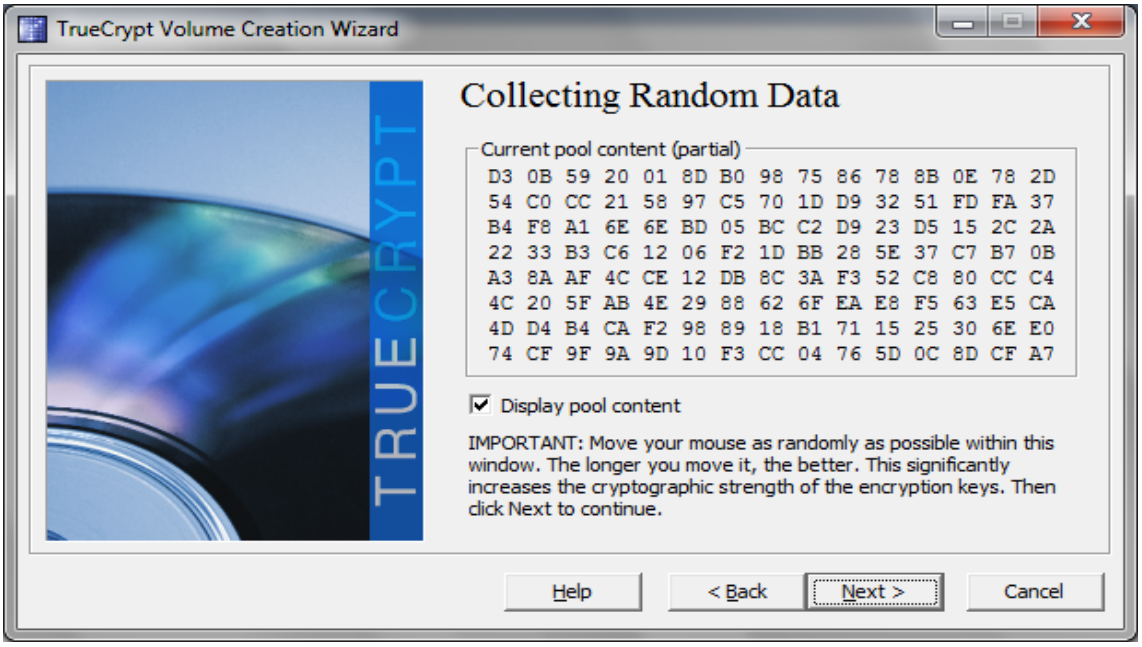
Kuva 8. Salausalgoritmin valinta

Seuraavaksi valitaan salasana salaukselle. Tämä salasana on syytä muistaa tai tallentaa johonkin varmaan paikkaan, koska kiintolevy on käyttökelvoton ilman tätä. Jätetään Use keyfiles tyhjäksi, koska tässä työssä ei käytetä avaintiedostoja.

Kuva 9. Salasanan valinta

Jos valitaan alle 20:n merkin salasana, ohjelma ilmoittaa, että olisi suositeltavaa käyttää vähintään 20 merkkiä pitkää salasanaan. Tämä ei estä käyttämästä lyhyempää, mutta salasanan pituus vaikuttaa suoraan salauksen pitävyyteen.

Jos kuitenkin joku käyttää avaintiedostoja, niin seuraavaksi on niin sanotun avaintiedoston luonti kohta. Tässä kohdassa hiirtä satunnaisesti ruudussa liikuttamalla muodostetaan avaintiedosto joko RIPEMD-160:llä, SHA-512:ta tai Whirpoolilla.

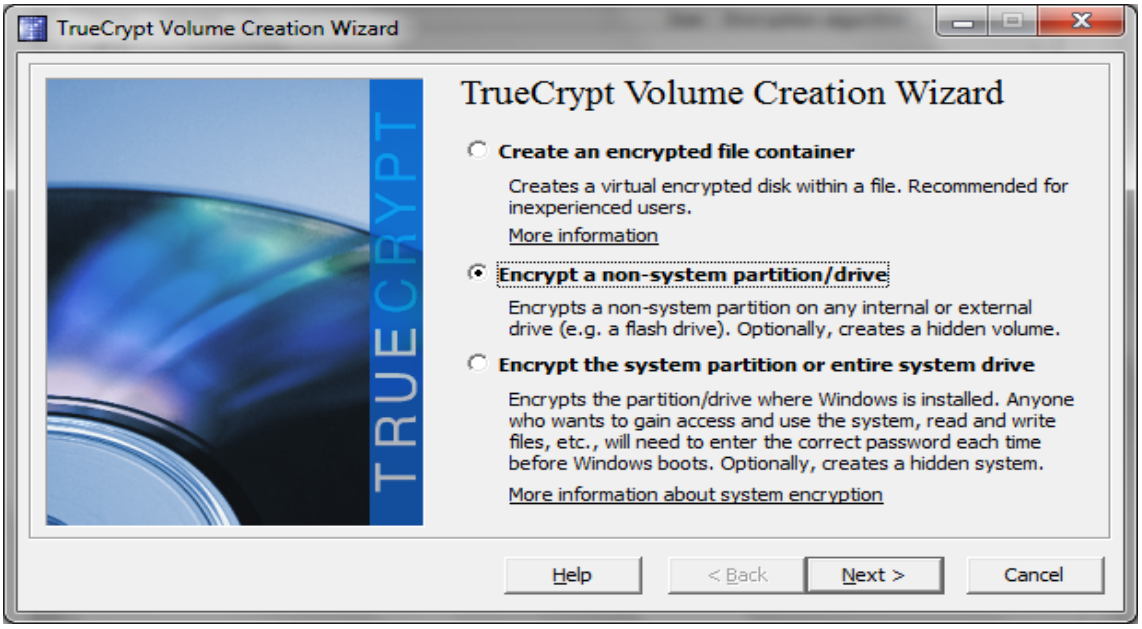


Kuva 10. Avaintiedoston luonti

Tämän jälkeen on vuorossa käynnistyslevyn luonti, johon palataan siihen varatussa luvussa. Seuraavaksi ohjelma aloittaa kiintolevyn salaamisen, tähän menee aikaa noin 2:sta tunnista aina vuorokauteen, riippuen kiintolevyn koosta ja tietokoneen tehosta. Jos kysymyksessä on ulkoinen kiintolevy, niin myös usb-väylä hidastaa prosessia. Kun salaus on suoritettu ja kone käynnistetään uudelleen, niin ensimmäiseksi avautuu eteen mustalla pohjalla valkoinen teksti TrueCrypt 7.1 ja Password johon syötetään asennuksessa valittu salasana. Tämän jälkeen käyttöjärjestelmä käynnistyy aivan normaalisti eikä salaus hidasta myöskään koneen käyttöä.

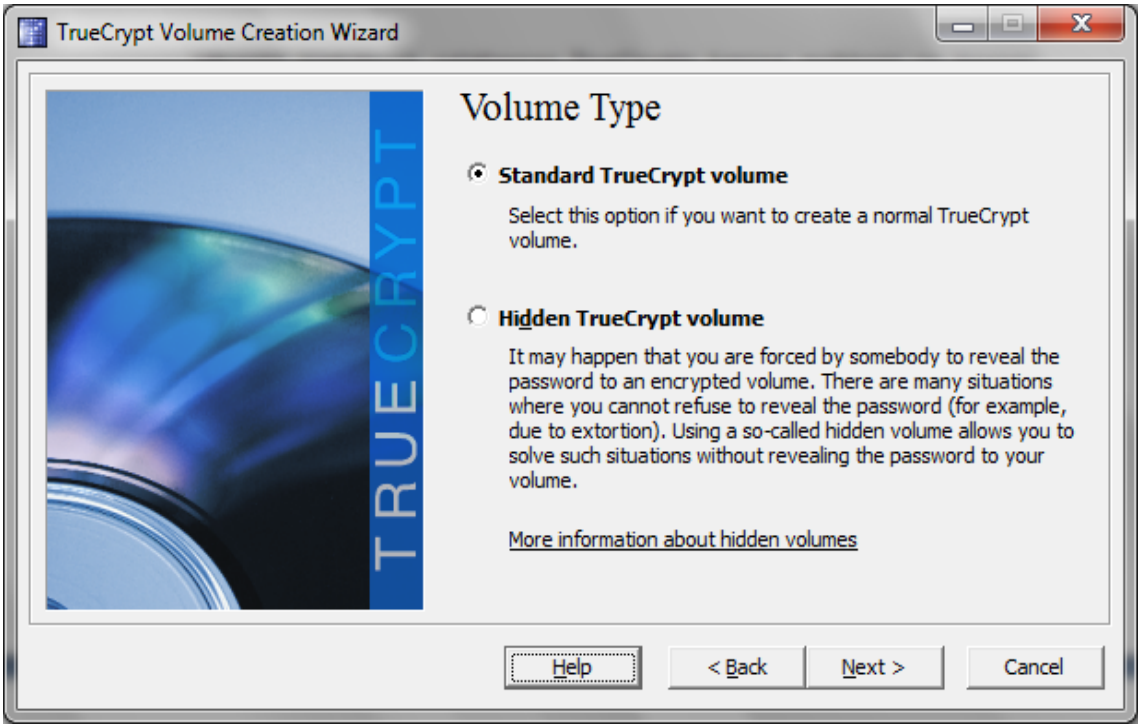
4.3 ULKOISEN KIINTOLEVYN SALAUS

Ulkoista kiintolevyä salattaessa TrueCryptin kanssa poikkeaa se hieman käyttöjärjestelmäosion salauksesta.



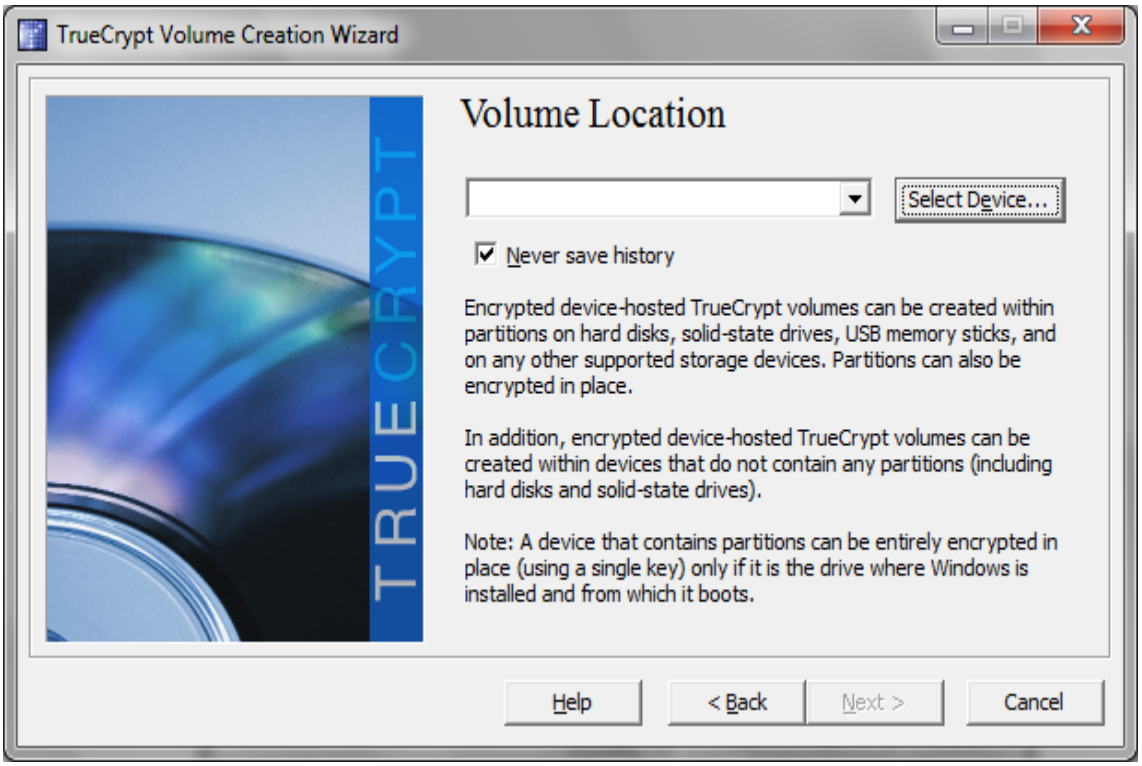
Kuva 10. Ulkoisen kiintolevyn salaus

Tässä tapauksessa salaustavaksi valitaan Encrypt a non-system partition/drive ja tämän jälkeen tulee eteen valintamahdollisuus piilotetun ja ei-piilotetun osion välillä.



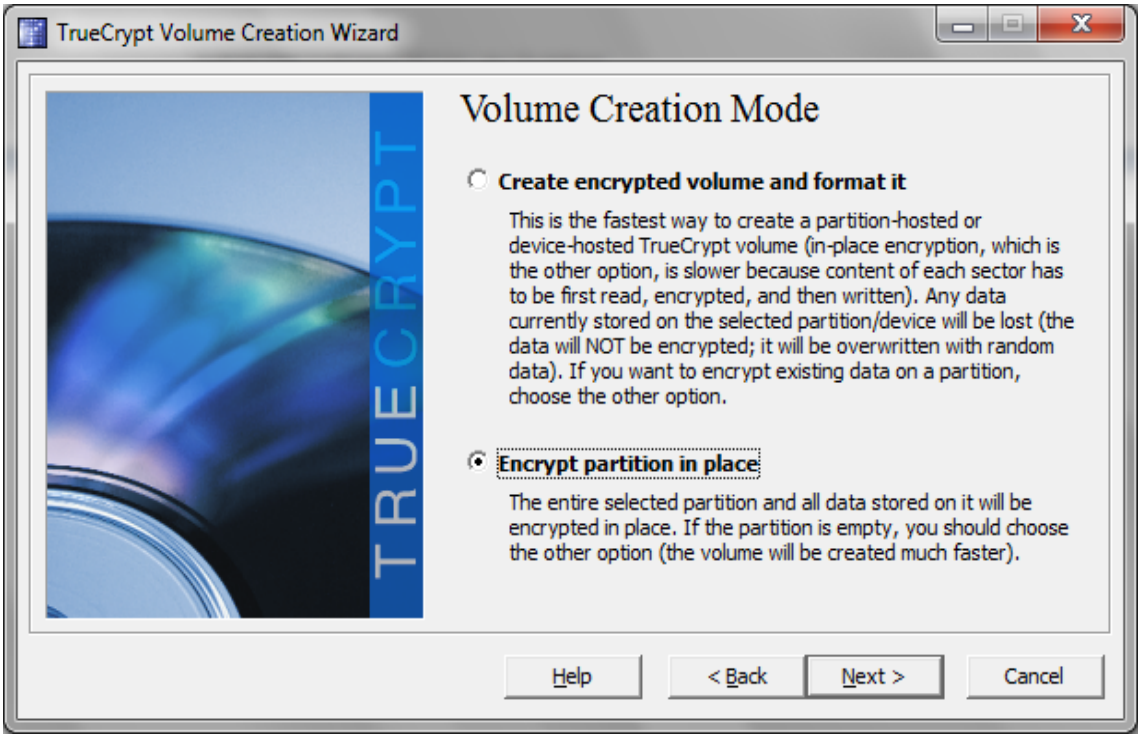
Kuva 11. Osion piilotusvalinta

Yleensä ei ole tarvetta osion piilotukselle, mutta sekin on mahdollista jos joku kokee sitä tarvitsevansa. Tässä työssä valitaan Standard TrueCrypt volume, jonka jälkeen valitaan salattava osio.



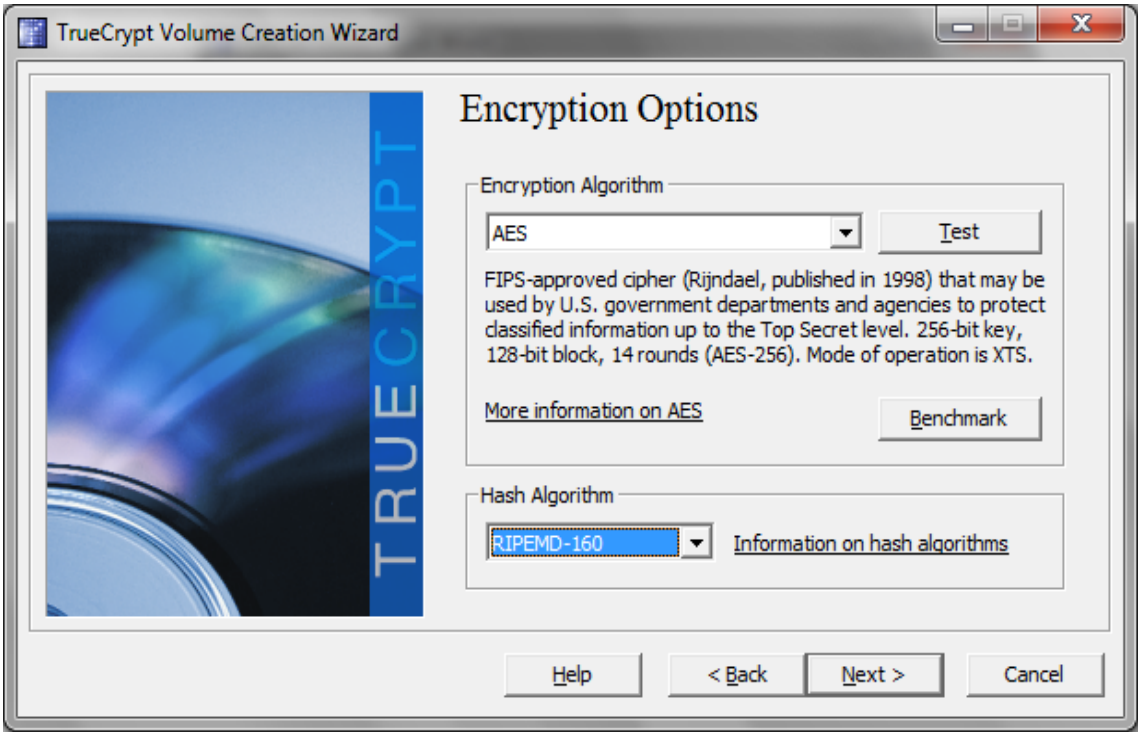
Kuva 12. Salattavan osion valinta

Seuraavaksi osion voi joko formatoida ja salata tai salata sellaisenaan. Jälkimmäinen vaihtoehto on suositeltava siinä tapauksessa, että osio sisältää jo dataa. Jos taas osio on tyhjä niin formatointi on suositeltavaa. Tässä tapauksessa ei formatoida, koska dataa on jo osiolle tallennettuna.



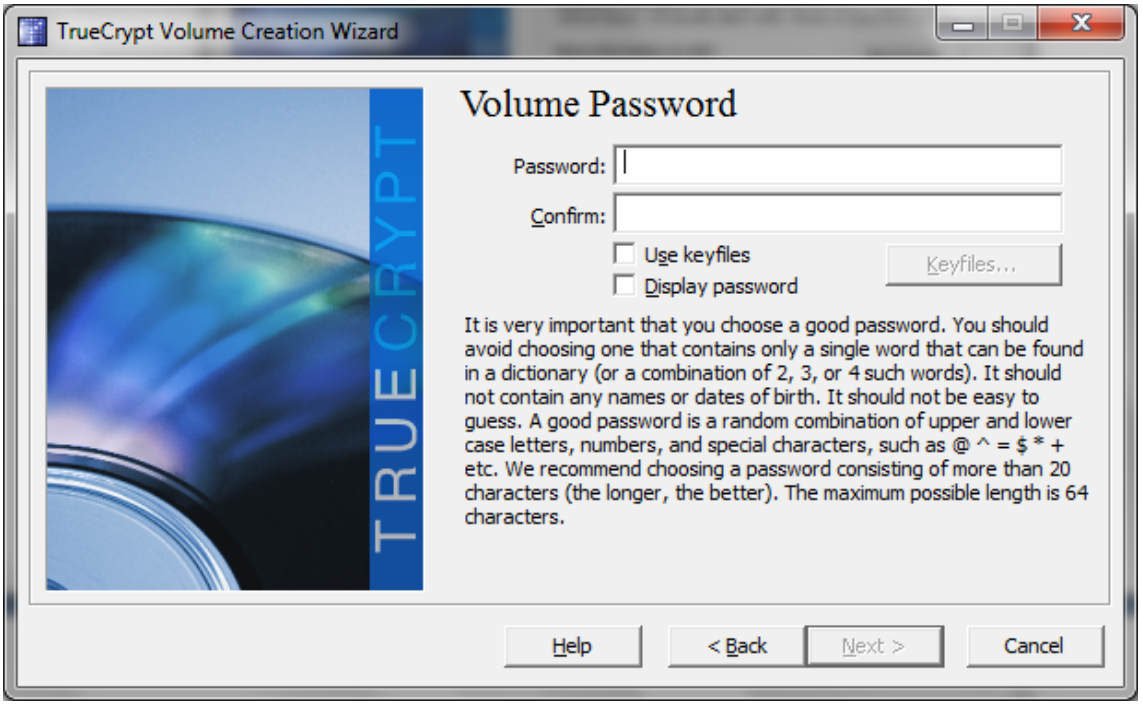
Kuva 13. Osion formatoinnin valinta

Tämän jälkeen valitaan salausalgoritmi ja tarvittaessa hashalgoritmi. Salausalgoritmiksi valitaan AES ja hashalgoritmilla ei ole väliä, koska sitä ei käytetä tässä työssä.



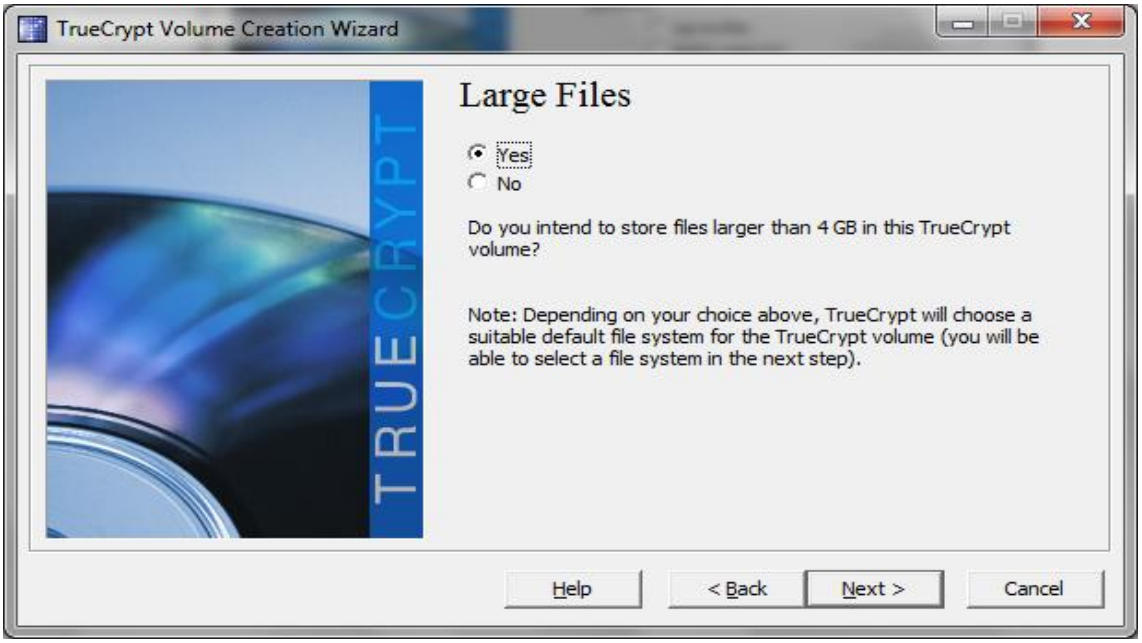
Kuva 14. Salaus- ja hashalgoritmin valinta

Sitten TrueCrypt varmistaa osion koon ja tämän jälkeen asetetaan osiolle salasana. Ohjelman suositusten mukaan minimi 20 merkkiä, mutta se hyväksyy myös lyhyemmät varmistuskyselyn jälkeen.



Kuva 15. Salasana asetus

Tämän toimenpiteen jälkeen TrueCrypt kysyy käyttäjältä, tuleeko hän tallentamaan yli neljän gigabitin tiedostoja tähän osioon. Tämä vaikuttaa valittavaan tiedostojärjestelmään.Jos ei tarvise yli neljän gigabitin tallennusta, on tiedostojärjestelmä FAT muuten NTFS. On hyvä valita neljän giganbitin tallennusvaihtoehto, vaikka ei heti olisikaan tarvetta näin suurille tiedostoille, muuten jos tulee tarve myöhemmin, joutuu purkamaan salauksen ja tekemään uudestaan.



Kuva 16. Tiedostojen koko

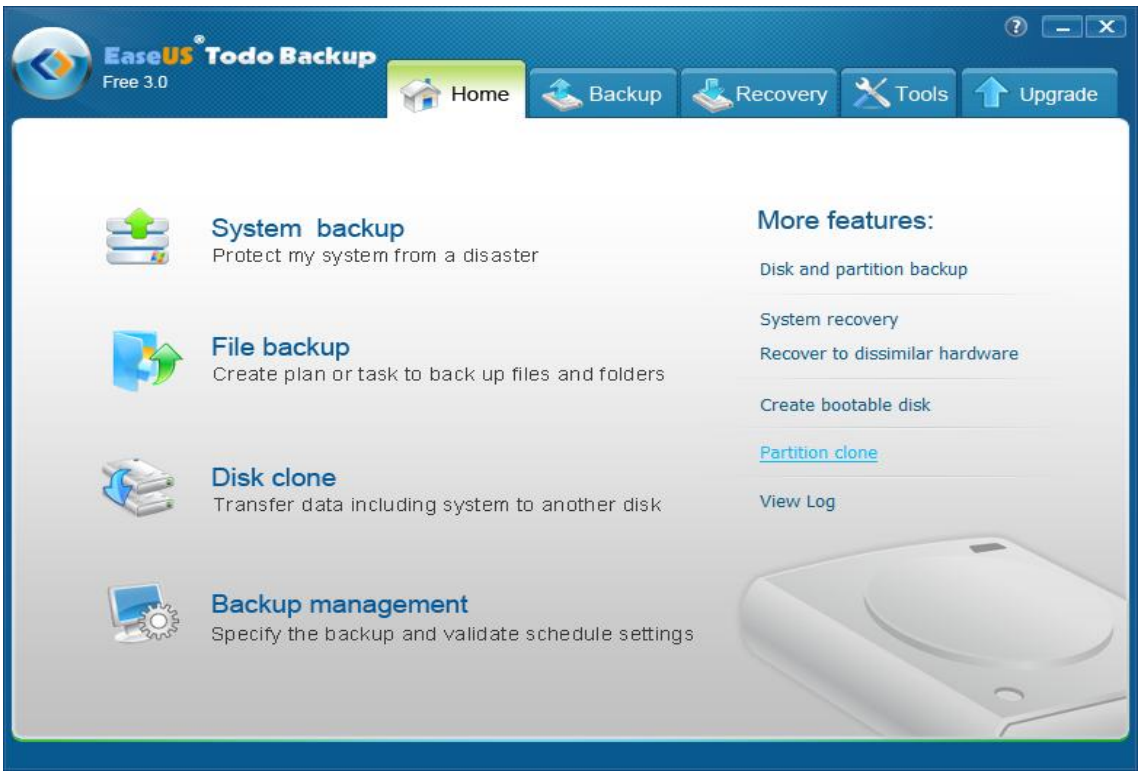
Lopuksi ohjelma aloittaa salauksen tai jos valittu on osion formatointi, niin ohjelma formatoi sen ensin, jonka jälkeen alkaa salaus.

5 VARMUUSKOPIOINTI JA KÄYNNISTYSLEVY

Ennen kuin tietokoneen käyttöjärjestelmää tai muuta osaa kiintolevystä aletaan salaamaan, niin on suositeltavaa tehdä varmuuskopio salattavasta datasta. Tämän voi tehdä esimerkiksi EaseUS Todo Backupilla. Tämä sen vuoksi, että jos salausohjelma jostain syystä keskeytyy hallitsemattomasti, niin datasta voi tulla käyttökelvotonta.

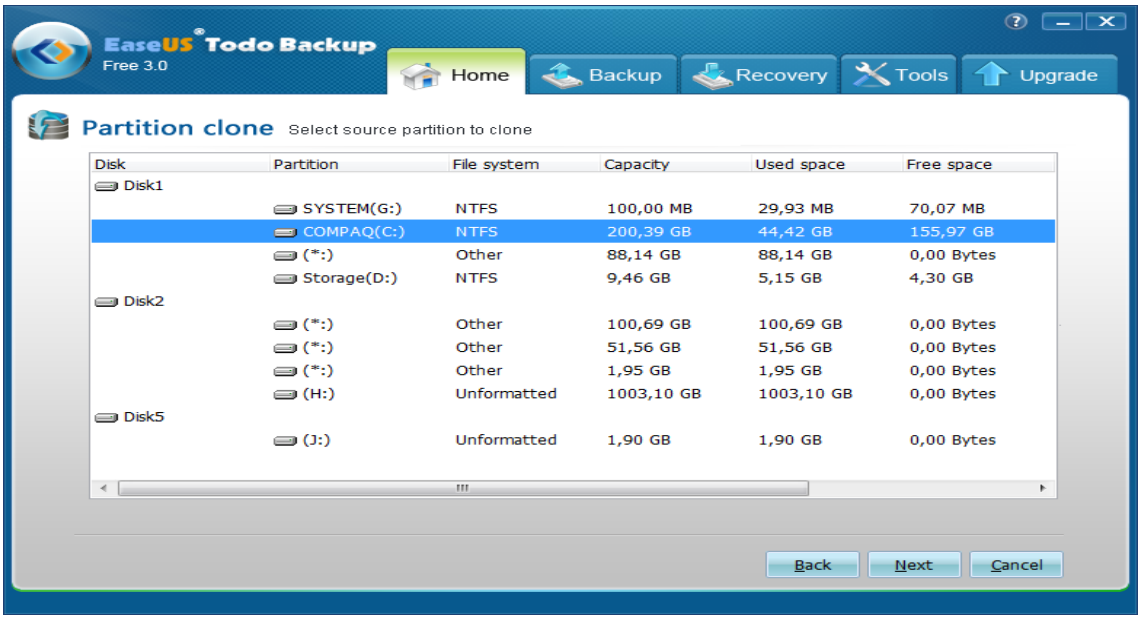
5.1 VARMUUSKOPIOIDEN TEKO

Tässä tapauksessa kun salataan käyttöjärjestelmäosio, niin on järkevintä kloonata kyseinen osio ja sama pätee myös ulkoiseen kiintolevyyn tai muuhun kiintolevyn osioon, jos siellä on dataa ennen salausta.



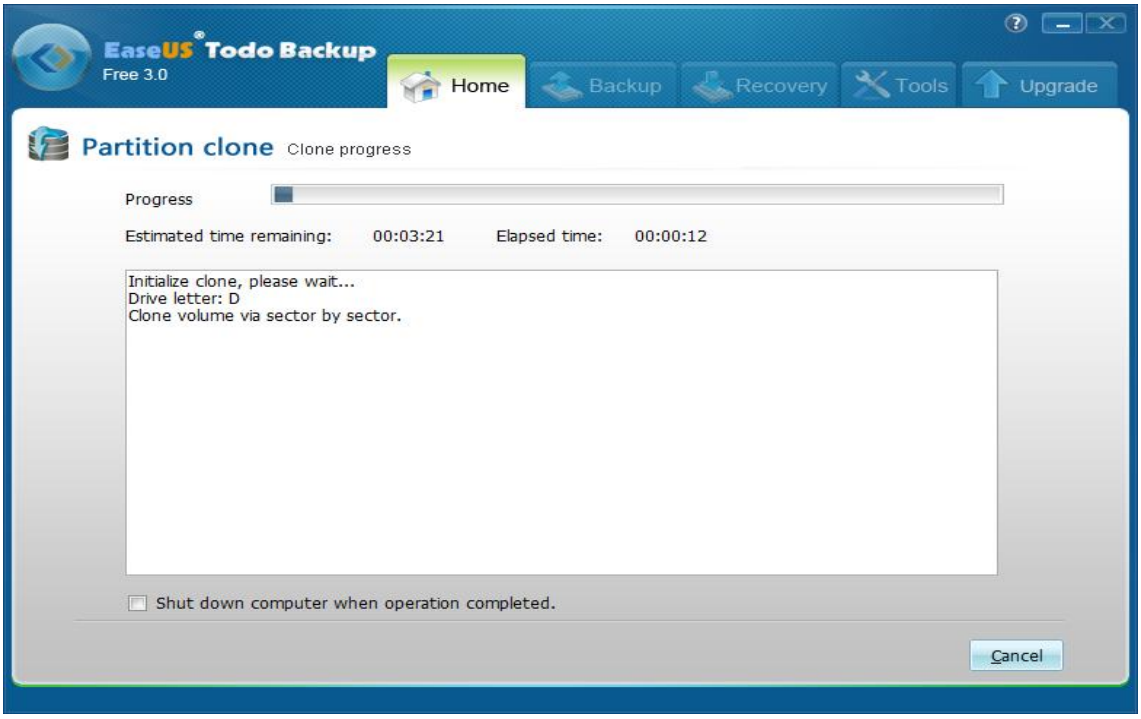
Kuva 17. EaseUS alkunäkymä

Valinnan jälkeen EaseUS näyttä käyttäjälle tämän koneen kaikki kiintolevyt ja niiden osiot, josta valitaan kloonattava osio. Tämän jälkeen samasta näkymästä käyttäjä valitsee haluamansa paikan johon kloonaus tallennetaan.



Kuva 18. Kloonattavan osion valinta

Valintojen jälkeen ohjelma varmistaa käyttäjältä valinnat, jonka jälkeen kloonaus alkaa.



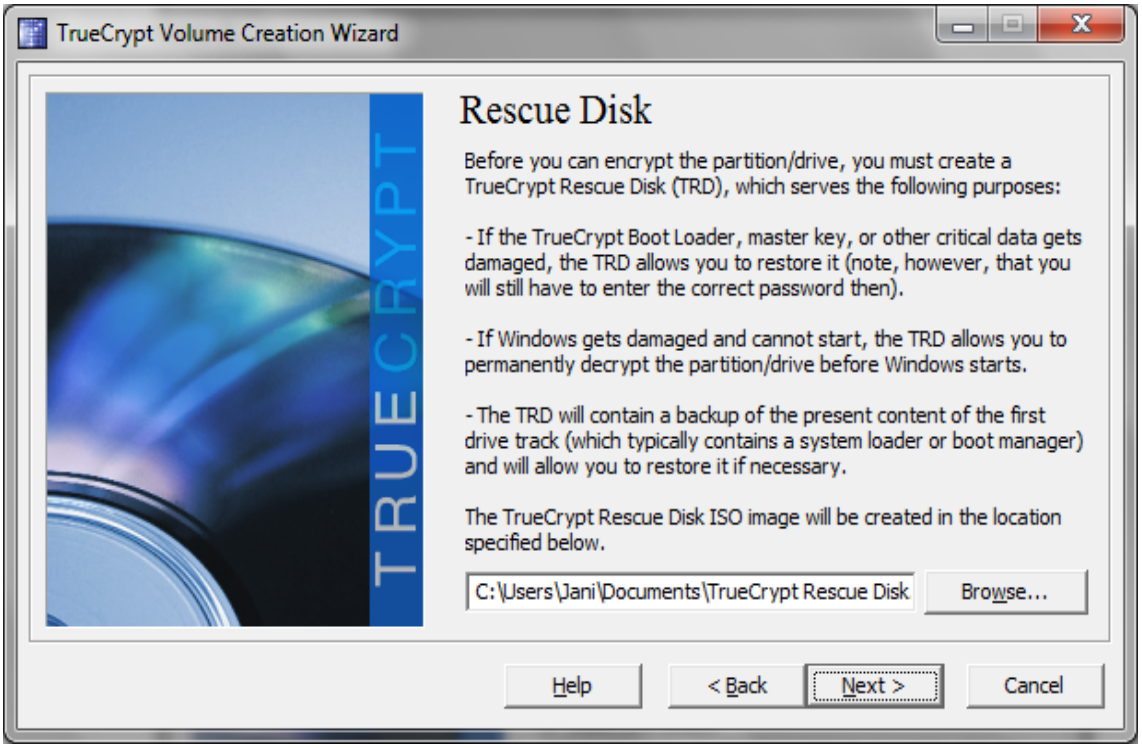
Kuva 19. Osion kloonaus

Sama toimenpide suoritetaan kaikille salattaville osiolle, jonka jälkeen salaaminen on turvallista, koska osiot voidaan palauttaa jos jokin menee pieleen.

5.2 KÄYNNISTYSLEVYN TEKO

TrueCrypt muodostaa bootloaderista levykuvan, josta tehdään käynnistyslevy siltä varalta, että koneellä oleva bootloader hajoaa. Tämä levy ei kuitenkaan auta siina tapauksessa, että käyttäjä unohtaa salasanansa.

Kun käynnistyslevy on tehty, niin TrueCrypt tarkistaa, että levy on toimiva.



Kuva 20. Käynnistyslevyn teko.

6 EMPIIRINEN TOTEUTUS

Työn suunnittelu ja toteutus on ollut melko suoraviivainen prosessi syksystä 2011 alkaen. Suunnitteluvaiheessa selvitettiin tarpeet, mahdolliset toteutustavat, aikataulu ja kustannukset.

Työ alkoi syksyllä 2011, jolloin suunnitteluvaihe alkoi ja kesti noin kaksi viikkoa. Opinnäytetyön aiheeksi työ alkoi muodostumaan 2011 lopulla osittain siksi, että sopiva työ tuli vastaan ja osittain työn mielenkiinnon vuoksi.Lopullisesti se on sovittu alkuvuodesta 2012. Tässä vaiheessa työn toteutus oli jo loppusuoralla yhtiössä ja tämän takia on tehty esimerkkisalaus työn tekijän henkilökohtaiselle koneelle.

Aineiston keruu alkoi käytännössä samalla kun työn suunnittelu alkoi. Materiaalia on haettu kirjastosta sekä internetistä. Suurena apuna on ollut FIPSin, NISTin ja NESSIEn kotisivut sekä algoritmien tekijöiden julkaisemat kirjat, dokumentit ja kotisivut.

Työ on ensin dokumentoitu ja sitten kirjoitettu Lokakuun 2011 ja Huhtikuun 2012 välillä. Yhtiössä tehty työ on niin sanottu jatkuva toimintatapa eli kaikki toimihenkilöiden koneet osioidaan ja salataan ennen käyttöönottoa. Työn esitys on ollut 4.4.2012 Turun ammattikorkeakoululla Salon toimipisteessä.

7 TYÖN LUOTETTAVUUS

Työn luotettavuuden varmistamiseksi teoriaosuuden tiedot on pyritty saamaan alkuperäisestä lähteestä ja välttämään wikipedian kaltaisia tiedonlähteitä. Itse työn tulokset taas ovat suoraan käytettävän algoritmin ja salasanan pituuden tuottama tulos, josta voidaan laskea salauksen pitävyys. Todellisuudessa ei ole realistista murtaa 256 bittistä AES salausta ainakaan tällä hetkellä tunnetulla tekniikalla järkevässä ajassa. Tämän lisäksi pitää ottaa huomioon yhtiön saamat edut tapauksissa, jossa kone varastetaan tai hukataan.

Lähteiden käytössä on pyritty käyttämään lähdekritiikkiä niissä kohdin, kun se on ollut tarpeen. Suurin osa tiedoista on kuitenkin dokumentoituja standardeja, joihin on voitava luottaa. Huomioon on myös otettu tekijän omat mahdolliset ennakkokäsitykset ja pyritty mahdollisimman objektiiviseen lopputulokseen.

Työn luotettavuutta vahvistaa se, että suurin osa materiaaleista on alkuperäisiä tekijöiden dokumentteja, kirjoja tai muita julkaisuja sekä NISTin, NESSIEn ja FIPSin kaltaisten instanssien virallisia dokumentteja ja muita julkaisuja.

Heikkona puolena työssä voisi pitää hieman liian laajaa aihealuetta, joka johtaa siihen, että mihinkään alueeseen ei olla syvennetty niin paljon kuin hieman kapeampi aihealue olisi mahdollistanut.

8 POHDINTA

Työtä aloittaessani oli tarkoituksena, että aihe olisi hieman laajempi, mutta hyvin pian tulin huomaamaan, että tässä aihealueessa on enemmän kuin tarpeeksi. Aihe olisi saanut ehkä olla vielä kapeampi, jotta siihen olisi pystynyt perehtymään syvemmin. Tiedon määrä, joka tuli vastaan perehtymällä salaukseen ja siihen liittyviin toimenpiteisiin oli valtava. Itse sain ainakin jollain tasolla kuvan nykypäivän salausalgoritmeista ja hash tiivisteiden teosta, sekä niiden historiasta.

Työn ideana on näyttää, miten tehdään käyttäjälleen huomaamaton ja varma salaus, jotta voidaan olla melko varmoja siitä, että kukaan ei pääse koneella olevaan dataan käsiksi jos kannettava häviää tai varastetaan. Samalla käydään pintapuolisesti läpi eri salaustekniikoita ja lopuksi on toteutettu salattu työympäristö, jonka murtaminen on äärimmäisen hankalaa salasanaa tietämättä.

Kun otetaan huomioon tilastot varastetuista [21] [22] ja kadotetuista kannettavista sanoisin, että tämä järjestely olisi syytä toteuttaa jokaisessa organisaatiossa ja henkilökohtaisissa kannettavissa tietokoneissa jos ne sisältävät vähänkään arkaluontoista tietoa. Työpaikallani se tarkoittaa yhtiön omia piirustuksia, sähköpostia, sopimuksia ynnä muuta sellaista. Tämän lisäksi tulee ottaa huomioon muut yhtiöt, joiden kanssa toimitaan yhteistyössä joko ostajana tai myyjänä, ja heidän arkaluontoiset tiedot, jotka ovat meidän hallussa. Tässä on jo osittain yhtiön imagokin ja luotettavuus kysymyksessä.

Tämän työn lopputulos on pieni osa siitä kaikesta tietoturvasta, joka vaaditaan tietojen turvassa pitämiseksi, mutta tämä ei maksa mitään toisin kuin antivirus- ja palomuuriohjelmistot. Tämänkin tosiasian takia olettaisi yhtiöiden toteuttavan vastaavia toimenpiteitä hieman hanakammin, varsinkin kun otetaan huomioon prosessin tuoma turva datalle tietyissä tilanteissa.

Tämän työn lisäksi kun huolehditaan kunnollisesta, keskitetystä palomuurista, VPN-yhteyksien ammattimaisesta luomisesta sekä jos yhtiöllä on verkkojakoja esimerkiksi asiakkaille, niin nämä on syytä tehdä erittäin huolellisesti.

Nämäkään kaikki yhteensä eivät tee järjestelmästä läpäisemätöntä, mutta vaatii hyökkääjältään erityistä osaamista, jotta tietää ja hallitsee keinot, jolla järjestelmään voi hyökätä. Tämän lisäksi se vaatii myös erityisen pitkäjänteistä työtä, joten tämä todennäköisesti karsii ensimmäiset harrastelijat.

Tulevaisuudessa tavoitteena olisi saada vastaava käytäntö kaikkiin toimipisteisiin sekä mahdollisesti suunnitella servereiden tiedostojako uusiksi ja sijoittaa kriittisimmät tiedostot salauksen taakse. Lisäksi verkossa jaettaville tiedostoille voisi harkita niiden salaamista ja aukaisu onnistuisi ainoastaan oikealla avaimella. Tällöin tietomurron tapahtuessa tiedostot ovat joka tapauksessa erittäin hyvin suojattu.

LÄHTEET

1. NISTin kotisivu. [Verkkodokumentti] Viitattu 2.4.2012. https://www.cosic.esat.kuleuven.be/nessie/

3. NSAn kotisivu. [Verkkodokumentti] Viitattu 2.4.2012. http://www.nsa.gov/

4. FIPSin kotisivu. [Verkkodokumentti] Viitattu 2.4.2012. http://www.itl.nist.gov/fipspubs/

5. NESSIEn kotisivu. [Verkkodokumentti] Viitattu 2.4.2012. https://www.cosic.esat.kuleuven.be/nessie/

6. MBR Wikipedia [Verkkodokumentti] Viitattu 2.4.2012. http://en.wikipedia.org/wiki/Master_boot_record

7. AES Proposal: Rijndael, Joan Daemen ja Vincent Rijmen, 1999. Viitattu 18.2.2012.

7. IEC Wikipedia. [Verkkodokumentti] Viitattu 2.4.2012. http://en.wikipedia.org/wiki/International_Electrotechnical_Commission

8. ISO Wikipedia. [Verkkodokumentti] Viitattu 2.4.2012. http://en.wikipedia.org/wiki/ISO

9. RAID Wikipedia. [Verkkodokumentti] Viitattu 2.4.2012. http://en.wikipedia.org/wiki/RAID

10. SHA-1:n murtaminen [Verkkodokumentti] Lainattu 3.3.2012. http://lukenotricks.blogspot.com/2009/05/cost-of-sha-1-collisions-reduced-to-252.html

10. FIPS-standarsointi. [Verkkodokumentti] Viitattu 20.2.2012. http://en.wikipedia.org/wiki/Federal_Information_Processing_Standard

11. TrueCrypt License 2.8 [Verkkodokumentti] Viitattu 18. 2.2012. http://www.truecrypt.org/legal/license

12. AES algoritmi. [Verkkodokumentti] Viitattu 20.2.2012. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

13. Serpent algoritmi. [Verkkodokumentti] Viitattu 18.2.2012. http://csrc.nist.gov/archive/aes/round2/comments/20000513-pbora.pdf

14. Twofish algoritmin toimintaperiaate [Verkkodokumentti] Viitattu 18.2.2012. http://www.google.fi/url?sa=t&rct=j&q=twofish%20algorithm%20horatiu&source=web&cd=1&ved=0CCUQFjAA&url=https%3A%2F%2Fusers.cs.jmu.edu%2Fabzugcx%2Fpublic%2FStudent-Produced-Term-Projects%2FCryptology-2002-SPRING%2FTwofish-Encryption-Algorithm-by-Horatiu-Paul-Stan-cu.ppt&ei=UwBvT7udPOLT4QTI88C_Ag&usg=AFQjCNFJWOFskAkYeZpcIGpNwHJ2VNtTJA

15. Feistelin toimintaperiaate. [Verkkodokumentti] Lainattu 22.2.2012. http://www.freesoft.org/CIE/Topics/143.htm

16. RIPEMD-160 [Verkkodokumentti] Viitattu 26.3.2012. http://homes.esat.kuleuven.be/~bosselae/ripemd160/pdf/AB-9601/AB-9601.pdf

17. SHA-2:n toimintaperiaate [Verkkodokumentti] Lainattu 3.3.2012. http://www.cast-inc.com/ip-cores/encryption/sha-256/index.html

18. AES algoritmin toiminta. [Verkkodokumentti] Lainattu 20.2.2012. http://www2.it.lut.fi/kurssit/03-04/010628000/Seminars/AES.pdf

19. Whirlpoolin toiminta. http://www.seas.gwu.edu/~poorvi/Classes/CS381_2007/Whirlpool.pdf.

20. SHA-3 projekti [Verkkodokumentti] Viitattu 3.3.2012. http://csrc.nist.gov/groups/ST/hash/timeline.html

21. Varastetut kannettavat. [Verkkodokumentti] Viitattu 1.4.2012. http://en.wikipedia.org/wiki/Laptop_theft

22. Varastetut kannettavat. [Verkkodokumentti] Viitattu 1.4.2012. http://www.skadoit.com/news/2-1-billion-in-lost-laptops-highlights-need-for-remote-backup-tracking/

23. NISTin AES dokumentointi. [Verkkodokumentti] Viitattu 2.4.2012. http://csrc.nist.gov/archive/aes/index.html

24. Opinnäytetyön kirjoittamisen käytännön opas. Jyväskylän ammattikorkeakoulu, 2010.