Usaldusnimekirja krüptovõtmeted

Loomine ja nõuded protseduurile

RaulWalter

esmaspäev, 29. detsember 2025

Dokumendi info	
Loomise kuupäev	neljapäev, 27. november 2025
Projekt	RIA-D2302-T07
Saaja(d)	Riigi Infosüsteemi Amet
Autor(id)	Lauris Kaplinski, Raul Metsma, Raul Kaidro
Versioon	1.1

Turvaklassifikatsioon	
PIIRATUD (Asutusesiseseks kasutamiseks)	Dokument sisaldab piiratud kasutusega tehnilist teavet ja tööprotseduuride kirjeldusi, mis ei ole mõeldud avalikuks levitamiseks. Dokumendile on lubatud juurdepääs ainult isikutele, kellel on tööalane vajadus selles sisalduvat teavet kasutada. Dokumendi edastamine, kopeerimine, avalikustamine või muutmine väljaspool selle ettenähtud kasutuskonteksti on lubatud üksnes RIA pädeva halduri loal. Dokumendi sisu on koostatud heas usus ning tugineb tööde kirjelduses sätestatud nõuetele ja arendusmeeskonna poolt teostatud tehnilistele tegevustele. Dokument ei sisalda delikaatseid isikuandmeid ega riigisaladust. Dokumendis kirjeldatud protseduuride järgimine on vastutus dokumenti kasutaval asutusel ja selle volitatud töötajatel. RaulWalter ja dokumenti koostanud isikud ei vastuta otsese ega kaudse kahju eest, mis tuleneb dokumenti kasutava organisatsiooni otsustest või protseduuride täitmisest.

Versiooni info

Kuupäev	Versioon	Muudatused/märkmed
27.11.2025	0.1	Esmane mustand
28.11.2025	0.2	Kosmeetilised täiendused. Lisatud nõuded dokumenteerimisele ja hävitamise protseduurile.
07.12.2025	0.3	Parandused ja täiendused testimise käigus avastatud mittevastavusele; kosmeetilised täiendused.
15.12.2025	0.4	Täiendatud eraldi lisaga 1, kus on loetletud loodud/kasutatud failid ja nende SHA256 sõrmejäljed.
18.12.2025	0.5	4 peatüki korrastamine, parandatud ettevalmistused offline-protseduuriks, USB-mälupulga ja RP ettevalmistus; kosmeetilised parandused.
19.12.2025	1.0	Uuendatud skriptide sõrmejäljed peale pisiparandusi. Lisatud 4.2 alampeatükid iga operatsioonisüsteemi jaoks eraldi.
29.12.2025	1.1	Täiendatud vastavalt RIA ettepanekutele. Kosmeetilised parandused.

Sisukord

1. Ülevaade ja eesmärk

Käesolev juhendmaterjal kirjeldab täielikku, standardiseeritud ja kordusvõimelist protseduuri, mille alusel Riigi Infosüsteemi Amet saab luua usaldusnimekirja e-allkirjastamiseks vajalikud krüptovõtmed ja nende vastavad sertifikaadid turvalises, kontrollitud ning võrguühenduseta keskkonnas. Juhend hõlmab kogu tööprotsessi Raspberry Pi ettevalmistamisest kuni YubiKey-dele genereeritud võtmete testimise, turvaümbrikusse paigutamise ja seadmete hävitamiseni.

Juhendmaterjal on koostatud Riigi Infosüsteemi Ameti (RIA) tööde kirjelduse nõuete alusel ning sisaldab kõiki tegevusi, skripte, konfiguratsioone ja kontrollnõudeid, mis on vajalikud selleks, et tagada krüptovõtmete ning sertifikaatide ohutus, terviklus ja standarditele vastavus.

Juhendi eesmärk on tagada, et võtmete loomise protsess oleks:

- turvaline, et kogu protsess toimub kontrollitud, offline keskkonnas;
- korratav, et kõik sammud on skriptitud ning dokumenteeritud üheselt mõistetavalt;
- auditikindel, et protsess on jälgitav, protokollitav ning reprodutseeritav;
- standarditele vastav, et sertifikaadi parameetrid vastavad ETSI TS 119 612 ja ETSI TS 119 312 nõuetele;
- töökorras, et iga loodud võti ja sertifikaat on testitud ning valideeritud.

Juhend on mõeldud RIA tehnilisele personalile ja tööprotsessis osalevatele määratud rollidele. Selle järgimiseks ei ole vaja süsteemi arenduslikku tausta, kuid eeldatakse:

- elementaarset Linuxi käsurea kasutamise oskust,
- arusaamist PIV(*Personal Identity Verification*)-põhisest võtmehaldusest,
- võimekust kontrollida failide krüptograafilisi sõrmejälgi,
- oskust käsitleda füüsilisi turvaseadmeid (YubiKey, mälupulgad, SD-kaardid).

Juhend ei eelda RIA-lt kõrvalvahendeid ega täiendavat infrastruktuuri. Kõik vajalikud skriptid ja välised sõltuvused antakse ette juhendi lisades ning skriptide sõrmejäljed võimaldavad tagada iga kasutatava komponendi tervikluse.

Juhend tagab, et kõiki samme järgides on võimalik luua kaks sõltumatut, erineva kehtivusajaga võtmepaari ja nende sertifikaadid, paigutada need YubiKey seadmetesse ning valmistada need ette tulevaseks kasutamiseks usaldusnimekirja allkirjastamisel.

2. Riistvara ja tarkvarakomponendid

Käesolev peatükk kirjeldab kõiki vahendeid, mida on vaja usaldusnimekirja allkirjastamiseks kasutatavate krüptovõtmete loomiseks. Protsess on projekteeritud töötama täielikult offline režiimis Raspberry Pi seadmel, kasutades skriptitud ja eelnevalt kontrollitud töövahendeid. Kõik komponendid, alates SD-

kaardist kuni YubiKey seadmeteni, peavad olema uued, tehasepakendis ning tööks ettevalmistatud ainult selle protsessi tarbeks.

Riistvarakomplekt ja tarkvarakomponendid on määratud tööde kirjeldusega ning neid tuleb käsitleda kui fikseeritud turvaelementide komplekti. Kõik kõrvalekalded (näiteks teise mudeli YubiKey kasutamine, erinev Ubuntu versioon, muud skriptid) ei ole lubatud ilma RIA sisemise turvameeskonna kirjaliku kooskõlastuseta.

Järgnevates alapeatükkides defineeritakse:

- riistvaranõuded, sealhulgas Raspberry Pi konfiguratsioon, SD-kaart, mälupulkade kasutus ja YubiKey'de nõuded;
- tarkvarakomponendid ja sõltuvused, mis on vajalikud võtmete genereerimiseks ja sertifikaatide loomiseks;
- skriptid ja konfiguratsioonid, mis viiakse üle mälupulgale ning paigaldatakse Raspberry Pi töökeskkonda;
- failistruktuur ja vajadus sõrmejälgede kontrolliks, et tagada terviklus enne käivitamist.

Selle peatüki eesmärk ei ole kirjeldada protseduuri ennast, vaid luua raamistik, mis tagab:

- et kõik vajalikud vahendid on protsessi alustamise hetkeks olemas,
- et kõik komponendid on terviklikud, kontrollitud ja nõuetekohased,
- et protsessi käigus ei tekiks olukorda, kus puuduv komponent peatab või kompromiteerib võtmete loomise.

Riist- ja tarkvarakomponentide korrektne valik ja ettevalmistus on võtmete loomise protsessi kõige olulisem eeldus. Kõik järgnevad sammud eeldavad, et selles peatükis kirjeldatud vahendid on olemas ja kontrollitud.

2.1. Riistvara

Võtmete genereerimise protsess tugineb kontrollitud, piiratud ja ühekordselt kasutataval riistvarakomplektil. Kõik komponendid peavad olema uued, eelnevalt kasutamata ning ette nähtud ainult käesoleva töö tarbeks. Ühtegi seadmetest ei tohi kasutada muul eesmärgil, ega ühendada võrku väljaspool juhendis kirjeldatud kontrollitud protsesse.

Alljärgnev tabel kirjeldab kõiki võtmete loomise protsessi jaoks nõutud riistvarakomponente, nende tehnilised nõuded ja rollid tööprotsessis.

Komponent	Miinimumnõuded	Roll protsessis	Märkused
Raspberry Pi	Raspberry Pi 5 või samaväärne	Offline võtmete genereerimise keskkond	Kasutatakse ühekordselt; hävitatakse pärast protsessi
SD-mälukaart	≥16 GB, Class 10 või parem	Ubuntu OS kirjutamine ja käivitamine	Uus kaart; hävitatakse pärast protsessi
USB-A mälupulk 1	≥16 GB, FAT32 või ext4	Skriptide ja failide transport RIA tööjaam -> RP	Võib hävitada või tühjendada vastavalt RIA reeglitele
USB-A mälupulk 2	>= 16 GB, FAT32	Sertifikaadi, sõrmejäljefailide ja logifaili säilitamiseks	Säilitatakse ja kasutatakse vastavalt RIA reeglitele

Komponent	Miinimumnõuded	Roll protsessis	Märkused
YubiKey FIPS turvavõtmed	YubiKey 5C FIPS, ilma NFC-ta	Privaatvõtmete ja sertifikaatide lõplik hoidla	Kaks seadet: 5- ja 6-aastase kehtivusajaga sertifikaatidele
RIA standardtööjaam	Windows/Linux/ macOS, võrguga	ISO ja .deb pakettide allalaadimine, sõrmejälgede kontroll	Ei puutu kokku võtmetega; töötab ainult ettevalmistuse vahendina
Klaviatuur, hiir, monitor	USB ühendus	Raspberry Pi esmane seadistus ja skriptide käivitamine	Ei kuulu hävitamisele
Toiteadapter	RP5 originaal- või tootja nõuetele vastav	Tööprotsessi käigus stabiilse toite tagamine	Ei kuulu hävitamisele
USB-A - USB-C adapter	Standardne kommertsiaalne adapter	YubiKey ühendamiseks Raspberry Pi kõlge	Võib hävitada või säilitada vastavalt RIA reeglitele
Turvaümbrikud ja pitseerimismaterjal	Tamper-proof ümbrikud, pitseerimisteip	YubiKey'de ja PIN-koodide turvaline pakkimine	Kaks ümbrikku võtmetele, kaks PIN-idele

2.2. Tarkvara

Alljärgnev tabel sisaldab kõiki tarkvarakomponente ja sõltuvusi, mis on nõutud võtmete loomise protsessi edukaks läbiviimiseks. Kõik tarkvara allalaadimised tuleb teha RIA standardtööjaamast, kontrollida sõrmejälgi ning viia Raspberry Pi (RP) seadmesse ainult USB-mälupulga kaudu.

Komponent	Versioon/nõue	Roll protsessis	Märkused
Ubuntu Raspberry Pi OS	Ubuntu 24.04 LTS (desktop, ARM64)	Raspberry Pi operatsioonisüsteem	ISO peab olema allalaaditud ainult ametlikust kanalist; SHA256 kontroll kohustuslik. Desktop versiooni kasutatakse sest see võimaldab kõiki operatsioone teostada ilma internetiühenduseta.
Raspberry Pi Imager	Viimane ametlik versioon	ISO kirjutamine SD-kaardile	Kasutatakse RIA tööjaamas
`libccid`	ARM64 .deb pakett	Smartcard reader tugi RP-s	Paigaldatakse installiskriptiga USB mälupulgalt
`pcscd`	ARM64 .deb pakett	PC/SC teenus YubiKey kasutamiseks	Paigaldatakse installiskriptiga USB mälupulgalt
YubiKey PKCS#11 draiver	Kompileeritud RP peal	YubiKey PIV seadistus ja võtmete ning sertifikaadi loomine	Paigaldatakse installiskriptiga USB mälupulgalt
YubiKey PIV Tool	Kompileeritud RP peal	YubiKey PIV seadistus ja võtmete ning sertifikaadi loomine	Paigaldatakse installiskriptiga USB mälupulgalt

Komponent	Versioon/nõue	Roll protsessis	Märkused
openssl PKCS#11 tugi	Kompileeritud RP peal	Võtmete ja sertifikaadi loomine	Paigaldatakse installiskriptiga USB mälupulgalt
Skript: `install.pl`	Juhendis määratud versioon	RP ettevalmistamine (pakettide paigaldus, teekide kopeerimine)	Paigaldatakse installiskriptiga USB mälupulgalt
Skript: `gencert.pl`	Juhendis määratud versioon	Privaatvõtmete genereerimine YubiKey's, sertifikaadi loomine	Paigaldatakse installiskriptiga USB mälupulgalt; küsib PIN-koodi osi kahest osakonnast; Genereerib võtmed ja sertifikaadid; salvestab väljundfailid
Skript: `testcert.pl`	Juhendis määratud versioon	Testallkirja loomine ja valideerimine	Paigaldatakse installiskriptiga USB mälupulgalt; kontrollib, et sertifikaat ja privaatvõti töötavad korrektselt

2.2.1. Tarkvara terviklikkuse kontroll

Kõik allalaaditud komponendid (ISO, .deb paketid, teegid, skriptid) peavad olema:

1. kontrollitud ametliku, vähemalt SHA256 sõrmejäljega,
2. kantud kontrolltabelisse,
3. kopeeritud USB-mälupulgale ainult pärast kontrolli.
4. Kontrollitud veel kord peale kopeerimist

Tarkvara tervikluse kontroll on kohustuslik, sest protsess toimub offline keskkonnas, kus puudub võimalus allalaadimisi jooksvalt valideerida.

2.2.2. Tarkvara ettevalmistuse väljund

Peatüki lõpptulemuseks peab USB-mälupulk sisaldama järgmisi faile:

- `/install`
  - `install.pl` *(paigaldusskript)*
  - `libccid_1.5.5-1_arm64.deb`
  - `pcscd_2.0.3-1build1_arm64.deb`
  - `ykman-users.rules`
- `/install/lib`
  - `libykcs11.so.2.7.2`
  - `libykpiv.so.2.7.2`
  - `pkcs11.so`
- `/install/cert`
  - `yubico-piv-tool`
  - `gencert.pl`
  - `testcert.pl`
  - `openssl.cnf` *(sertifikaadi konfiguratsioonifail)*

Sellisel kujul USB-pulk on valmis kasutamiseks Raspberry Pi seadmes.

SHA256 sõrmejälgede kontrollimise tulemused võib USB-mälupulgale eraldi failina kopeerida, kui see on vajalik protsessi dokumenteerimiseks ja järjepidevuse tagamiseks.

3. Protsessi ülevaade

Käesolev peatükk annab ülevaate usaldusnimekirja allkirjastamiseks kasutatavate krüptovõtmete loomise protsessist. Protsess on täielikult skriptitud ning toimib offline keskkonnas ühe töötsükli jooksul, mis lõppeb kõigi ajutiste vahendite hävitamisega.

Tööprotsess koosneb kuuest põhifaasist:

Nr.	Nimetus	Sisu/tegevused	Väljund
1	Ettevalmistus RIA tööjaamas	- Ubuntu distributsiooni allalaadimine - Skriptide, teekide ja konfiguratsioonifailide kopeerimine - Kõigi failide SHA256/512 tervikluse kontroll	USB-mälupulk, mis sisaldab kõiki vajalikke faile ja kontrollsummafaile
2	Raspberry Pi ettevalmistamine	- Ubuntu OS kirjutamine SD-kaardile - RP käivitamine ja esmased seadistused (keel, klaviatuur, ajavöönd) - Võrguühenduste vältimine (WiFi keelatud) - Installiskripti käivitamine USB-pulgalt	Raspberry Pi on valmis võtmete genereerimiseks; vajalik tarkvara on paigaldatud
3	YubiKey seadmete ettevalmistamine	- Uute FIPS-sertifitseeritud YubiKey'de avamine - YubiKey kontroll	Turvaliselt ette valmistatud YubiKey'd, mis on valmis võtmete genereerimiseks
4	Võtmete ja sertifikaatide loomine (kaks korda kummagi sertifikaadi jaoks eraldi)	- gencert.pl skripti käivitamine - Sertifikaadi seadete määramine - Kaheosalise PIN-koodi sisestamine (kaks kätt) - Privaatvõtme genereerimine YubiKey sees - Sertifikaadi loomine ETSI nõuete järgi - Avaliku võtme ja sõrmejälgede salvestamine USB-pulgale	Kaks sertifikaadikomplekti (5a ja 6a): privaatvõtmed YubiKeys, avalikud võtmed ja SHA256 failid USB-pulgal
5	Võtmete testimine	- testcert.pl skripti käivitamine - Testfaili loomine - Allkirjastamine YubiKey privaatvõtmega - Allkirja valideerimine openssl'iga - Testfailide kustutamine	Test kinnitab, et privaatvõti ja sertifikaat töötavad korrektselt
6	Lõpetamine ja hävitamine	- YubiKey'de paigutamine turvaümbrikutesse - PIN-koodide poolte paigutamine eraldi turvaümbrikutesse - Pitseerimine ja protokollimine - SD-kaardi ja Raspberry Pi füüsiline hävitamine - Logide ja mälupulkade üleandmine	Valmis ja verifitseeritud võtmed krüptopulgal, avalikud failid mälupulgal, kõik ajutised vahendid hävitatud ja protsess lõpetatud

4. Rasberry Pi ettevalmistamine (offline keskkond)

Raspberry Pi valmistatakse ette kui ühekordselt kasutatav, täielikult võrguühenduseta keskkond, milles viiakse läbi nii skriptide käivitamine kui ka võtmete loomine. Kõik alljärgnevad tegevused tuleb läbi viia rangelt määratud järjekorras.

4.1. Ubuntu distributsiooni allalaadimine ja kontroll

Kõik allalaadimised teostatakse RIA standardtööjaamast.

Sammud:

1. Laadi alla Ubuntu 24.04 LTS ARM64 Raspberry Pi desktop-distributsioon ametlikust allikast:
   `https://cdimage.ubuntu.com/releases/noble/release/`

   Dokumendi avaldamise hetkel on failinimi:
   `ubuntu-24.04.3-preinstalled-desktop-arm64+raspi.img.xz`

2. Laadi alla distributsiooni ametlik SHA256 kontrollsummafail.
   `https://cdimage.ubuntu.com/releases/noble/release/SHA256SUMS`

3. Arvuta tööjaamas ISO faili SHA256 sõrmejälg:
   `shasum -a 256 ubuntu-24.04.3-preinstalled-desktop-arm64+raspi.img.xz`

   Dokumendi avaldamise hektel allalaetud faili sõrmejälg on:
   `04a87330d2dfbe29c29f69d2113d92bbde44daa516054074ff4b96c7ee3c528b`

4. Võrdle tulemusi ametliku kontrollsummaga.

5. Märgi kontrolli tulemus SHA256 kontrolltabelisse.

Väljund: Verifitseeritud Ubuntu distributsioonifail, valmis SD-kaardile kirjutamiseks.

4.2. Raspberry Pi Imager-i installimine

Selle etapi eesmärk on ette valmistada RIA tööjaam Ubuntu distributsiooni kirjutamiseks SD-kaardile.

Raspberry Pi Imager on ametlik tööriist Raspberry Pi operatsioonisüsteemide allalaadimiseks ja kirjutamiseks SD-kaardile või USB-andmekandjale. Tööriist võimaldab lisaks eelseadistada kasutaja, SSH-ligipääsu ja võrguseaded, mistõttu on see eelistatud lahendus standardtöövoogudes.

4.2.1. Paigaldamine Linuxi (Ubuntu) tööjaamas

RIA standardtööjaamas (Ubuntu) paigaldatakse Raspberry Pi Imager APT paketihalduri kaudu järgmise käsuga:

```
sudo apt install rpi-imager
```

Paigaldamise käigus:
- kasutatakse Ubuntu ametlikke paketirepositooriume;
- kontrollitakse pakettide terviklust ja päritolu allkirjastatud paketiloendite (GPG) alusel;
- paigaldatakse vajalikud sõltuvused automaatselt.

Pärast edukat paigaldamist on rakendus kättesaadav nii graafilisest menüüst kui ka käsurealt käsuga rpi-imager.

Ubuntu kontrollib automaatselt kõigi ametliku distributsiooni pakettide näpujälgi allkirjastatud nimekirja vastu.

4.2.2. Paigaldamine macOS tööjaamas

Kui kasutusel on macOS-i tööjaam, paigaldatakse Raspberry Pi Imager eraldiseisva rakendusena.

Kui tööjaamas on kasutusel Homebrew paketihaldur, on võimalik Raspberry Pi Imager paigaldada käsurealt:

```
brew install --cask raspberry-pi-imager
```

Alternatiivse lahendusena käib macOS-ile tüüpilise rakenduse paigaldusprotsessi kaudu.

Paigaldamise sammud:

1. Laadi alla Raspberry Pi Imageri macOS-i paigalduspakett (.dmg) alla Raspberry Pi ametlikult veebilehelt https://www.raspberrypi.com/software/
2. Ava allalaaditud .dmg fail.
3. Lohista rakendus Raspberry Pi Imager kausta Applications.
4. Käivita rakendus kaustast Applications.

Esmasel käivitamisel võib macOS kuvada turvateate tundmatult arendajalt pärit rakenduse kohta. Kui rakendus on alla laaditud ametlikust allikast, on selle avamine lubatud vastavalt macOS-i turvapoliitikale.

4.2.3. Paigaldamine Windowsi tööjaamas

Kui kasutajal on MS Windowsi tööjaam, paigaldatakse Raspberry Pi Imager eraldiseisva paigaldusprogrammi abil.

Paigaldusprotsess:

1. Laadi alla Raspberry Pi Imageri Windowsi paigaldusfail ametlikult Raspberry Pi veebilehelt.
2. Käivita paigaldusprogramm tavakasutaja õigustes.
3. Järgi paigaldusviisardi juhiseid.

Väljund: Raspberri Pi Imager on installitud.

4.3. SD-kaardi ettevalmistamine

SD-kaart on ühekordne ja kasutatakse ainult selle protsessi tarbeks.

Sammud:

1. Aseta SD-kaart RIA tööjaama kaardilugejasse.
2. Ava Raspberry Pi Imager
3. Vali õige Raspberry Pi mudel: *Raspberry Pi 5*
4. Vali “**Use Custom**” ning vali allalaaditud Ubuntu distributsioon.
5. Vali kirjutamiseks 1. punktis sisestatud SD-kaart.
6. Jäta eelkonfiguratsioon vahele (RP seadistatakse käsitsi).
7. Kirjuta valitud operatsioonisüsteem SD-kaardile.
8. Väljuta SD-kaart pärast kirjutamist, kui kasutad Linux.

Väljund: Valmis SD-kaart Ubuntu OS-iga.

4.4. Mälupulga ettevalmistamine

Selle etapi eesmärk on ette valmistada USB-mälupulk, mis sisaldab võtmete ja sertifikaatide loomiseks vajalikke installatsioonifaile ja skripte. Kõik toimingud viiakse läbi RIA standardtööjaamas enne offline-keskkonda liikumist.

Sammud:

1. Sisesta USB-mälupulk RIA tööjaama USB-pessa.
2. Veendu, et mälupulk on operatsioonisüsteemile loetav ja kirjutatav. Vajadusel eemalda tootjapoolne kirjutuskaitse.
3. Vorminda USB-mälupulk FAT32 failisüsteemiga kui ta on vormindamata.
4. Kopeeri käesoleva juhendiga kaasasolev `/install` kataloog USB-mälupulga juurkataloogi.
5. Vajadusel kontrolli kopeeritud failide terviklust, võrreldes nende SHA256 sõrmejälgi käesoleva juhendi Lisa 1 andmetega.
6. Eemalda USB-mälupulk operatsioonisüsteemist turvaliselt (`umount` / `eject`).
7. Eemalda USB-mälupulk füüsiliselt USB-pesast.

Väljund: Valmis USB-mälupulk, mis sisaldab `/install` kataloogi koos installatsioonifailide ja käivitusskriptidega ning on valmis kasutamiseks offline-keskkonnas.

Pärast selle etapi lõppu ei ole võtmete ja sertifikaatide loomiseks RIA tööjaama enam vaja.

4.5. Raspberry Pi esmane seadistamine (offline)

Raspberry Pi tuleb käivitada täielikult offline režiimis. WiFi ja muud võrguliidesed peavad jääma konfigureerimata ja välja lülitatuks.

Sammud:

1. Sisesta SD-kaart Raspberry Pi seadmesse.
2. Ühenda monitor, klaviatuur, hiir ja toiteadapter.
3. Lülita Raspberry Pi sisse.
4. Esmakonfiguratsioonis:
   - 4.1. vali õige keel ja klaviatuur,
   - 4.2. ära seadista WiFi ühendust,
   - 4.3. määra ajavöönd (Estonia Time),
   - 4.4. loo kasutajakonto `user` ja *parool**

   **Parooli on vaja ainult Raspberry Pi-sse sisselogimiseks ja skriptide installatsiooniks, seega ei pea siin kasutama turvalist parooli.*

5. Taaskäivita Raspberry Pi pärast seadistuste lõppu (automaatne).
   - 5.1. keela telemeetria („Improve Ubuntu“ – Skip).
   - 5.2. Ubuntu võib mingil hetkel küsida, kas uuendada pakette - vasta eitavalt.
6. Ava terminal ja seadista õige kuupäev ning kellaeg

```
sudo date -s 20251229EET13:25:00
```

See on vajalik, et logikirjed oleks õigete aegadega

7. Võrgureeglid:
   7.1. WiFi seadistamine on keelatud.
   7.2. Ethernet-kaablit ei tohi ühendada.
   7.3. Ühtegi automaatset uuendust ei tohi lubada.

Väljund: Offline Ubuntu keskkond, mis on valmis skriptide paigaldamiseks.

4.6. Failide paigaldamine RP-sse (offline)

Selles etapis kantakse RIA tööjaamas ette valmistatud sõltuvused ja skriptid RP-sse.

Sammud:

1. Sisesta USB-mälupulk Raspberry Pi seadmesse.
2. Ava terminal.
3. Navigeeri installikataloogi (USB-pulga nimi võib erineda):

```
cd /media/user/KINGSTON/install
```

4. Juhul kui kasutati macOS RIA tööjaama, puhasta ._* macOS metadata artefaktidest mälupulgal:

```
find . -type f -name '._*' -delete
```

5. Kontrolli kõigi installatsioonifailide sha256 sõrmejälgi (Lisa 1):

```
find . -type f -exec sha256sum {} \;
```

Offline-režiimis installitavate Ubuntu ametlike pakettide ametlikud sõrmejäljed on kirjas distributsiooni andmebaasis (https://cdimage.ubuntu.com/releases/noble/release/SHA256SUMS)

6. Kontrolli, kas `libccid` ja `pcscd` on saadaval:

```
apt-cache show libccid
apt-cache show pcscd
```

7. Käivita installiskript:

```
sudo perl install.pl
```

8. Sisesta Raspberry Pi kasutajakonto parool.

9. Jälgi, et skript paigaldab korrektselt:
   9.1. `libccid`,
   9.2. `pcscd`,
   9.3. YubiKey teegid `/usr/local/lib`

Paigaldamise kontroll: Pärast skripti lõpetamist peab terminalis olema näha kiri "***Installation complete***".

10. Kontrolli kas sertifikaadi loomise skript kopeeriti kausta `~/cert` käsuga:

```
ls -la ~/cert
```

11. Välju USB pulga kataloogist ja haagi USB pulk lahti.

```
cd ~
sudo umount /media/user/KINGSTON
```

12. Sulge terminal.

13. Eemalda USB-mälupulk RP USB pesast.

Väljund: Raspberry Pi on paigaldatud tarkvaraga, mis võimaldab võtmeid genereerida ja sertifikaate loodavale YubiKey tokenile salvestada.

4.7. Offline keskkonna lõpetamine

Pärast kõikide komponentide paigaldamist tuleb veenduda, et Raspberry Pi jääb täielikult offline.

Kontroll:
Avage terminal ja andke terminalis käsud:

- `ip a`

  näitab ainult loopback liidest, WiFi ja ethernet on seadistamata (`NO CARRIER`).

- `systemctl status pcscd --no-pager`

  näitab, et smartcard teenus on töövalmis (`Loaded:`)

  Kui peale systemctl käsku on lõpus kiri `Lines...(END)`, vajutage '**q**'

Väljund: Raspberry Pi on täielikult ettevalmistatud võtmete genereerimiseks ja vastab offline-operatsiooni nõuetele.

5. YubiKey ettevalmistamine

YubiKey turvavõtmed toimivad ainsa füüsilise kohana, kuhu usaldusnimekirja allkirjastamise privaatvõtmed luuakse ja kuhu need ka jäävad. Ühtegi privaatvõtit ei salvestata RP failisüsteemi ega edastata sellest välja. Seetõttu peab YubiKey'de ettevalmistus toimuma kontrollitud ja dokumenteeritud protseduurina.

Kogu protsess viiakse läbi välise ühenduseta (offline) Raspberry Pi seadmes, mis on eelnevalt seadistatud vastavalt käesoleva dokumendi peatükile 4.

5.1. Nõuded YubiKey seadmetele

Nõue	Kirjeldus
Mudel	YubiKey 5C FIPS (ilma NFC-ta)
FIPS-tase	FIPS 140-2 valideeritud
Kogus	2 seadet (üks 5-aastase ja üks 6-aastase sertifikaadi jaoks)
Seisund	Uus, tehasepakendis, eelnevalt kasutamata
Füüsiline kontroll	Pakendi terviklus kontrollitakse enne avamist

NB: *Kui pakend on kahjustatud, seadet kasutada ei tohi.*

5.2. YubiKey kontroll

YubiKey peab olema eelnevalt kasutamata tehaseseadetega.

Sammud:

1. Ühenda YubiKey Raspberry Pi USB-porti.
2. Ava terminal RP-s.
3. Navigeeri kataloogi ~/cert

```
cd ~/cert
```

4. Anna rakendusele käivitusõigus:

```
chmod 755 yubico-piv-tool
```

5. Kontrolli, et seade tuvastub:

```
./yubico-piv-tool --action=status
```

Kontroll: yubico-piv-pool näitab, et seadmes pole ühtegi võtit (*All non-listed slots are empty*)

Väljund: PIV moodul on puhas ja valmis uute võtmete genereerimiseks.

5.3. Kaheosalise PIN-koodi protseduur

PIN jagatakse kaheks subjektiivselt sõltumatuks osaks (A ja B), kus:

- Osakond A teab ainult PIN-i esimest poolt (PIN_A),
- Osakond B teab ainult PIN-i teist poolt (PIN_B).

Kumbki ei tohi kunagi teada tervet koodi.

Protseduur skripti käigus:

1. Skript küsib PIN-i sisestamist.
2. Osakond A sisestab PIN_A (esimene pool).
3. Osakond B sisestab PIN_B (teine pool).
4. Skript kombineerib need üheks PIN-iks.
5. PIN-i terviklikku väärtust ei kuvata ega salvestata väljaspoole YubiKey-d.
6. Skript määrab PUK koodiks juhusliku arvurea mida ei kuvata ega salvestata. Seega pole peale võtmete loomist PIN koodi enam võimalik muuta.

Turvanõuded:

- PIN_A ja PIN_B peavad paiknema eraldi turvaümbrikes.
- Ümbrikke ei tohi hoida samas ruumis enne protseduuri algust.
- Protseduuri ajal peavad mõlemad osakonnad viibima sama terminaliseerija juures.
- PIN-sisestust ei tohi logida (terminali ajalugu ei salvestata).

5.4. YubiKey PIV slotide kasutus

Võtmed luuakse ainult slot 9c (Digital Signature key) jaoks.

Slot	Kasutus	Staatus	Märkused
9a	Authentication	Ei kasutata	Tuleb jätta tühjaks
9c	Digital Signature	Kasutatakse	Privaatvõti genereeritakse siia
9d	Key Management	Ei kasutata	Jääb tühjaks
9e	Card Authentication	Ei kasutata	Jääb tühjaks

Slot 9c on ainus koht, kuhu genereeritakse privaatvõti ning kus asub sertifikaat protsessi lõpuks.

Skrip genereerib ka juhusliku YubiKey haldusvõtme, mida ei salvestata. Seega pole enam võimalik privaatvõtit ega sertifikaati YubiKey-s üle kirjutada ega uuesti genereerida.

5.5. Kontroll enne võtmete loomist

Enne skripti käivitamist peab:

1. YubiKey olema ühendatud.
2. YubiKey peab olema tehaseseadetega.
3. PIN_A ja PIN_B olema füüsiliselt kohal (eraldatud ümbrikud).
4. Raspberry Pi olema offline ja pcscd töötama.

Terminali kontrollkäsud (peatükid 4.7 ja 5.2):

```
systemctl status pcscd --no-pager
./yubico-piv-tool --action=status
```

Kui teenus töötab, võib alustada võtmete loomise faasiga.

6. Võtmete ja sertifikaatide loomine

Selles peatükis kirjeldatakse protseduuri, mille käigus luuakse kaks eraldi krüptovõtmete komplekti (5-aastane ja 6-aastane), genereeritakse sertifikaadid ning kõik vajalikud väljundfailid. Privaatvõtmed luuakse otseselt YubiKey seadmes ning neid ei ekspordita ega salvestata RP failisüsteemi.

Protsess on skriptitud ja toimub täielikult offline Raspberry Pi keskkonnas.

6.1. Eeldused enne alustamist

Enne võtmete loomise alustamist peavad olema täidetud järgmised tingimused:

Kontrollpunkt	Kirjeldus	Eeltingimuse staatus
Raspberry Pi valmis	Tarkvara paigaldatud, offline	Peab olema täidetud
pcscd töötab	Smartcard teenus aktiivne	Kontrollitakse käsuga: `systemctl status pcscd`
YubiKey lähtestatud	Yubikey tehaseseadetega	`yubico-piv-tool` peab näitama tühje slotte

Kontrollpunkt	Kirjeldus	Eeltingimuse staatus
PIN_A ja PIN_B olemas	Kaks sõltumatut “kätt” kohal	Mõlemad poolkoodid vajalikud
USB-mälupulk ühendatud	Mälupulk kuhu salvestatakse avalikud failid (sertifikaat, sõrmejäljed ja log) sisestatud Raspberry Pi USB pesasse.	Vajalik failide salvestamiseks
Võrk puudub	WiFi/Ethernet täielikult keelatud	Kohustuslik turvanõue

Kui ükskõik milline eeltingimus ei ole täidetud, protseduuri ei alustata.

6.2. Sertifikaadi loomise skripti ülesanne

Skripti `gencert.pl` eesmärk on privaatvõtme loomine YubiKey slot 9c-s, sertifikaadi genereerimine, avaliku võtme ja SHA256 failide salvestamine, testallkirjastamine.

`gencert.pl` skript teeb järgmised toimingud:

1. Küsib kasutajalt kataloogi, kuhu failid salvestada
2. Küsib sertifikaadifaili nime.
3. Küsib sertifikaadi kehtivuse algusaega (UTC).
4. Küsib sertifikaadi kehtivusaega päevades.
5. Kontrollib, et YubiKey PIN on tehase vaikeväärtus
6. Küsib PIN_A ja PIN_B ning kombineerib need üheks PIN-iks.
7. Salvestab PIN-i YubiKey-le
8. Kontrollib, et uus PIN on aktiivne
9. Genereerib juhusliku PUK koodi ja salvestab YubiKey-le
10. Genereerib juhusliku haldusvõtme ja salvestab YubiKey-le
11. Genereerib privaatvõtme otse YubiKey slot 9c sees.
12. Genereerib self-signed sertifikaadi määratud parameetritega etteantud kataloogi.
13. Laeb sertifikaadi samasse slotti 9c.
14. Arvutab ja salvestab:
    14.1. sertifikaadi SHA256 sõrmejälje,
    14.2. sertifikaadifaili SHA256 sõrmejälje.
15. Käivitab testallkirjastamise ja testallkirja valideerimise.
16. Teavitab, kui protsess õnnestus või ebaõnnestus.

6.3. Sertifikaadi loomine (samm-sammult)

Kõik alljärgnevad käsud täidetakse Raspberry Pi terminalis.

Samm	Tegevus	Käsk/sisend	Väljund
1	Ava sertifikaadiloome kataloog	`cd ~/cert`	Terminalis ollakse õiges kataloogis
2	Kontrolli skripti olemasolu	`ls -l gencert.pl`	Skript nähtav
3	Ühenda USB pulk	Sisesta väljundfailide jaoks ette nähtud USB pulk	Seade aktiveerub

Samm	Tegevus	Käsk/sisend	Väljund
4	Ühenda YubiKey	Sisesta üks uutest YubiKey'dest Raspberry Pi USB-porti.	Seade aktiveerub.
5	Kontrolli YubiKey info	`./yubico-piv-tool --action=status`	Seade tuvastatud. Kui seadme info ei ilmu, protsessi ei jätkata.
6	Käivita skript	`sudo perl gencert.pl`	Skript alustab tööd.
7	Sisesta väljundkataloog	Vaikimisi /media/user/KINGSTON/cert	Kataloog, kuhu lõppfailid salvestatakse. Kui kataloogi ei eksisteeri, see luuakse.
8	Sisesta sertifikaadi nimi	*Näiteks*: RIA_TL_2025_5y	Määrab väljundfaili prefiksi.
9	Sisesta sertifikaadi kehtivuse algusaeg (UTC)	YYYY-MM-DD HH:MM:SS	Määrab sertifikaadi kehtivuse alguse.
10	Sisesta kehtivusaeg päevades	5 aastane = 1825; 6 aastane = 2190	Määrab sertifikaadi kehtivuse lõpu.
11	Kinnita valikud	Y/n	Skript alustab võtmete ja sertifikaatide genereerimist.
12	Sisesta PIN_A	Osakond A	A-pool PIN-ist.
13	Sisesta PIN_B	Osakond B	B-pool PIN-ist.
14	Skript genereerib privaatvõtme	Automaatne	ECCsecp384r1 privaatvõti slot 9c-s.
15	Skript genereerib sertifikaadi	Automaatne	Self-signed X.509 sertifikaat, prefiks.crt väljundkataloogis.
16	Skript arvutab SHA256 sõrmejäljed	Automaatne	.sha256 failid USB-pulgal.
17	Skript sooritab testallkirja	Automaatne	Testfail allkirjastatud.
18	Skript valideerib allkirja	Automaatne	Test õnnestunud / error.
19	Skript väljastab kokkuvõtte	Terminali väljund	Lõplik kinnitus.

NB: *OpenSSL PKCS#11 draiver nõuab, et OpenSSL sertifikaadi loomise ja/või allkirjastamise protseduuris oleks PIN kood failina kättesaadav. Skripti töö käigus kirjutatakse see fail peale protseduuri lõppu korduvalt üle ja kustutatakse, s.h. ka siis kui mõni skripti samm ebaõnnestub.*

Siiski on soovitav sel juhul, kui skripti töö ebaõnnestus, anda täiendavalt käsk:

```
shred /tmp/pin.txt
```

* Sertifikaadi genereerimise etapis väljastab OpenSSL veateate: **"2050C81FFFF0000: error:0300007F: digital envelope routines: evp_pkey_get0_RSA_int:expecting an rsa key:../crypto/evp/p_legacy.C: 37:"**. See on normaalne ja ei indikeeri turvaprobleeme. Tegemist on bug-iga YubiKey PKCS11 moodulis (YKCS11/mechanisms.c:125), mis võtme tüübi määramiseks püüab kõigepealt avada avaliku võtme faili RSA mooduliga (ja kui see ebaõnnestub, määrab võtme tüübiks EC).

6.4. Skripti sisemised tehnilised operatsioonid

Skript teeb järgmised krüptograafilised operatsioonid:

Operatsioon	Käsk/kasutus	Toimingu eesmärk
Privaatvõtme loomine	`./yubico-piv-tool --slot=9c --action=generate --algorithm=ECCP384 --pin=$pin --pin-policy=ALWAYS --touch-policy=NEVER...`	Genereerib ECCP384 privaatvõtme YubiKey seadmele
Sertifikaadipäringu loomine	`openssl req -new -key pkcs11:id=%02 -out $csrfile -sha512 -subj $SUBJ`	Loob sertifikaadipäringu etteantud “Subject” andmetega
Sertifikaadi loomine	`openssl ca -selfsign -create_serial -keyfile pkcs11:id=%02 -out **$certfile** -startdate **$nb** -enddate **$na** -extensions v3_ca -in **$csrfile** -verbose -preserveDN -batch`	Loob self-signed sertifikaadi
Sertifikaadi import	`./yubico-piv-tool --slot=**9c** --action=import-certificate --input=**$certfile**`	Paigaldab sertifikaadi YubiKey sloti
Sertifikaadi sõrmejälg	`openssl x509 -in **$certfile** -noout -sha256 -fingerprint`	Tervikluskontroll
Faili sõrmejälgede arvutus	`sha256sum **$certfile**`	Auditeeritavuse tagamiseks

USB-mälupulgale tekkivad väljundfailid:

Fail	Kirjeldus
`<nimi>.crt`	Loodud sertifikaat
`<nimi>.sha256`	Sertifikaadi SHA256 sõrmejälg
`<nimi>.crt.sha256`	Sertifikaadifaili SHA256 sõrmejälg
`<nimi>.log`	Logifail

Privaatvõti jääb ainult YubiKey seadmesse. Privaatvõti ei välju YubiKey seadmest.

Teise võtme/sertifikaadi loomiseks tuleb eemaldada esmalt kasutatud YubiKey token ja sisestada uus (soovi korral eemaldada ja asendada ka mälupulk) ning jätkata protsessi nagu on juba kirjeldatud peatükis 6.3:

Peale esimese (5-aastase) sertifikaadi edukat loomist:

1. Eemalda esimene YubiKey.
2. Sisesta teine YubiKey.

3. (Valikuline) Haakida lahti mälupulk
4. (Valikuline) Eemaldada mälupulk
5. (Valikuline) Sisestada uus mälupulk
6. Korda kogu peatükis 6.3 kirjeldatud protsessi uue nime ja kehtivusajaga.

Tungivalt soovitatav on PIN_A ja PIN_B uue väärtuse kasutamine iga YubiKey tokeni jaoks eraldi.

6.5. Sertifikaadi parameetrid (normatiivne nõue)

Sertifikaat peab sisaldama järgmisi omadusi (vastavalt tööde kirjeldusele):

Sertifikaadi parameeter	Parameetri väärtus
Signature Algorithm	ecdsa-with-SHA512
Hash Algorithm	SHA512
Public Key	ECCsecp384r1
Issuer	Estonian Information System Authority
Subject	Estonian Information System Authority
Key Usage	Digital Signature (80)
Basic Constraints	End Entity, Path Length = 0
Validity	5 või 6 aastat loomise hetkest
Certificate Path	Estonian Trusted List Scheme Operator
Status	“Root CA is not trusted because it is not in the Trusted Root Certification Authorities store” (oodatav tulemus)

7. Võtmete testimise protseduur

Sertifikaadi loomise skript teeb testallkirjastamise YubiKey-sse salvestatud privaatvõtmega ja kontrollib allkirja kehtivust genereeritud sertifikaadifaili vastu.

Soovitav on peale iga sertifikaadi loomist testi korrata teises masinas või eemaldades YubiKey, tehes Raspeberry Pi-le restardi ja sisestades YubiKey uuesti. Sellega kontrollitakse, et:

1. privaatvõti töötab,
2. allkirjastamine õnnestub,
3. allkiri valideerub loodud sertifikaadi vastu,
4. sertifikaadi sõrmejälg vastab varem salvestatud väärtustele.

Kõik testid toimuvad offline Raspberry Pi keskkonnas, millesse on installitud YubiKey draiverid (kasutada sama `./install.pl` skripti, mis võtmete genereerimiseks).

7.1. Testimise eesmärk

Eesmärk	Kirjeldus
Võtme funktsionaalsus	Kontrollida, et YubiKey privaatvõti 9c slotis töötab ja võimaldab allkirja loomist
Sertifikaadi korrektne laadimine	Kontrollida, et sertifikaat 9c slotis vastab loodud sertifikaadile
Allkirja kontroll	Valideerida testallkiri *openssl*-i abil
Sõrmejälgede vastavus	Kontrollida, et SHA256 väärtused on identsed skripti loodud failidega

7.2. Testimisprotseduur

Mine kataloogi `~/cert` ja käivita skript `testcert.pl`

```
cd ~/cert
./testcert.pl
```

Testskript teeb läbi järgmised sammud:

Samm	Tegevus	Käsk/toiming	Oodatud väljund
1	Küsib PIN_A ja PIN_B		Skript teab kehtivat PIN koodi
2	Loob testfaili	`echo 'Allkirjastamise test' > /tmp/sign-test.txt`	Fail sign-test.txt loodud
3	Allkirjastab faili YubiKey abil	`openssl dgst -sha256 -sign pkcs11:%02 -out /tmp/sign-test.sig /tmp/sign-test.txt`	Fail sign-test.sig loodud
4	Ekspordib sertifikaadi YubiKey-st	`./yubico-piv-tool --slot=9c --action=read-certificate --output=/tmp/sign-test.crt`	Fail sign-test.crt loodud
5	Ekspordib avaliku võtme sertifikaadist	`openssl x509 -in /tmp/sign-test.crt -pubkey -noout > /tmp/sign-test.pub`	Fail sign-test.pub loodud
5	Valideerib allkirja	`openssl dgst -sha256 -verify /tmp/sign-test.pub -signature /tmp/sign-test.sig /tmp/sign-test.txt`	OpenSSL tagastab “Verified OK”
6	Väljastab sertifikaadi näpujälje	`openssl x509 -in /tmp/sign-test.crt -noout -sha256 -fingerprint`	Väärtus sama mis failis <nimi>.sha256
8	Kustutab testfailid	`rm /tmp/sign-test.*`	Ajutised failid eemaldatud
9	Kinnitab testi tulemuse	Skript näitab: “Signing successful”	Võti on kasutusvalmis

Testi tulemus salvestatakse faili `test-<CURRENTTIME>.log` jooksvas kataloogis.

7.3. Testi õnnestumise kriteeriumid

Test loetakse edukaks siis, kui kõik etapid on läbitud, skript trükib ekraanile “Signing successful” ja ekraanil kuvatud sertifikaadi sha256 näpujälg ühtib sertifikaadi loomisel faili `<nimi>.sha256` salvestatud näpujäljega.

Kontrollpunkt	PASS	FAIL
Sertifikaadi eksport	`Yubico-piv-tool --action=read-certificate` õnnestub; fail `/tmp/sign-test.crt` tekib	Yubico-piv-tool tagastab veateate, skript lõpetab töö
Privaatvõtme kasutus	PKCS#11 mootor suudab privaatvõtit kasutada allkirjastamiseks	PKCS#11 mootor ei suuda privaatvõtit kasutada (näiteks “*key reference error*”). Skript lõpetab töö
Allkirjastamine	Testfaili allkirjastamine õnnestub ja `testfile.sig` luuakse	Allkirjastamine ebaõnnestub; signatuuri faili ei looda, skript lõpetab töö
Allkirja kontroll	`openssl dgst` tagastab “*Verified OK*”	OpenSSL tagastab “*Verification Failure*” või muu veateate, skript lõpetab töö
Skript	Skript lõpetab sõnumiga “*Signing successful*”	Skript lõpetab veateatega või katkestab enneaegselt
Sertifikaadi sõrmejälg	Arvutatud SHA256 väärtus = `<nimi>.sha256`	SHA256 väärtused ei ühti
Kordustest (teisele YubiKey’le)	Mõlemad YubiKey’d läbivad identse *PASS* tulemuse	Üks või mõlemad võtmed ei läbi testi

NB: *Nagu ka gencert.pl skripti puhul on soovitav skripti töö ebaõnnestumise järel anda käsk:*

```
shred /tmp/pin.txt
```

8. Väljundite koondamine

Pärast iga sertifikaadi ja võtmekomplekti edukat loomist ning testimist salvestatakse kõik väljundfailid USB-mälupulgale, fikseeritakse sõrmejäljed ning koostatakse täielik ülevaade artefaktidest. Alljärgnev kirjeldab nõutavat failistruktuuri, failide tüüpe ning kontrollpunkte.

8.1. USB-pulga struktuur

Väljundfailid talletatakse järgmises struktuuris:

```
/cert/
    <nimi>.crt
    <nimi>.sha256
    <nimi>.crt.sha256

/logs/
```

```
    <nimi>.log
    test-<timestamp>.log (valikuline)

/meta/
    sha256_manifest.txt          (kõigi failide koond-sõrmejäljed)
    version.txt                  (skriptide versioon + RP OS info)
```

Märkus: Kui RIA defineerib täpsema sisereegli failistruktuuri kohta, kohandatakse see vastavalt.

8.2. Koondmanifesti koostamine

Manifestifail `sha256_manifest.txt` on terviklusdokumendi keskne osa.
Fail tekitatakse käsuga:

```
cd <sertifikaadifailide kataloog>
sha256sum * > ../meta/sha256_manifest.txt
```

Fail peab sisaldama kõiki `.crt`, `.pub` ja `.sha256` faile.

Näidis:

```
d5a8...  RIA_TL_2025_5y.crt
3f81...  RIA_TL_2025_5y.crt.sha256
41dd...  RIA_TL_2025_5y.pub
4c92...  RIA_TL_2025_5y.pub.sha256
...
```

8.3. Väljundfailid (ühekordne komplekt)

Faili nimi	Kirjeldus	Asukoht
<*nimi*>.crt	Loodud X.509 sertifikaat	/cert/
<*nimi*>.sha256	Sertifikaadi SHA256 sõrmejälg	/cert/
<*nimi*>.crt.sha256	Sertifikaadifaili SHA256 sõrmejälg	/cert/
<nimi>.log	Sertifikaadi loomise logifail	/logs/
Test-<timestamp>.log	Allkirjastamise testi logifail (valikuline)	/logs/
sha256_manifest.txt	Kogu komplekti koondsõrmejälgede fail	/meta/
version.txt	OS, skripti ja teekide versioonid	/meta/

8.4. Väljundite kontroll (PASS/FAIL)

Väljundite koondamise kontroll toimub alljärgneva tabeli alusel:

Kontrollpunkt	PASS - nõue täidetud	FAIL – nõue täitmata
Sertifikaadi fail olemas	<*nimi*>.crt eksisteerib	Puudub või on tühi

Kontrollpunkt	PASS - nõue täidetud	FAIL – nõue täitmata
Sertifikaadi sõrmejälg olemas	`<nimi>.sha256` on olemas	Puudub või on tühi
Sertifikaadifaili sõrmejälg olemas	`<nimi>.crt.sha256` eksisteerib	Puudub või on tühi
Sõrmejäljed korrektsed	SHA256 = arvutatud väärtus	Hash mismatch
Logi olemas	`<nimi>.log` olemas	Puudub
Failistruktuur korrektne	Kataloogid `/cert`, `/logs`, `/meta` on olemas	Vale struktuur
USB-mälupulk loetav	Failisüsteem on korras	Vigane või mitteloetav

Kahe sertifikaadi loomisest peaks USB-pulgal olema:

```
/cert/
    RIA_TL_2025_5y.crt
    RIA_TL_2025_5y.sha256
    RIA_TL_2025_5y.crt.sha256

    RIA_TL_2025_6y.crt
    RIA_TL_2025_6y.sha256
    RIA_TL_2025_6y.crt.sha256
```

Lisaks:
- Mõlemal komplektil oma logi
- Üks ühine `sha256_manifest.txt`, mis sisaldab kõigi failide sõrmejälgi.

9. Turvaümbrikud ja protseduuri lõpetamine

Pärast võtmete edukat loomist ja testimist tuleb kõigi füüsiliste turvaelementide käsitlemine viia lõpule kontrollitud, protokollitud ja auditeeritaval viisil. Selle protsessi eesmärk on tagada, et:

- privaatvõtmed jäävad ainult YubiKey seadmesse,
- võtmed ja PIN-koodid ei satuks ühegi inimese täieliku kontrolli alla,
- üleandmine toimub läbipaistvalt ja dokumenteeritult,
- RIA saab vastu võtta töö tulemused koos täieliku kontrollitavusega.

9.1. Ülevaade turvaelementidest

Turvaobjekt	Kirjeldus	Staatus pärast protsessi
YubiKey #1	Privaatvõti + 5-aastane sertifikaat	Pitseeritud turvaümbrik
YubiKey #2	Privaatvõti + 6-aastane sertifikaat	Pitseeritud turvaümbrik
PIN_A (5-aastane)	Osakond A PIN-i osa	Eraldi pitseeritud ümbrik
PIN_B (5-aastane)	Osakond B PIN-i osa	Eraldi pitseeritud ümbrik
PIN_A (6-aastane)	Osakond A PIN-i osa	Eraldi pitseeritud ümbrik

Turvaobjekt	Kirjeldus	Staatus pärast protsessi
PIN_B (6-aastane)	Osakond B PIN-i osa	Eraldi pitseeritud ümbrik
USB-mälupulk	Avalikud võtmed, sertifikaadid, sõrmejäljed	Eraldi ümbrikus, mitte pitseeritud

PIN-koodide komplekte on kaks (üks iga YubiKey jaoks).

Iga PIN-kood jaguneb kahte füüsiliselt eraldiseisvasse ümbrikusse.

9.2. Pitseerimis- ja pakendamisprotseduur

Alljärgnev protseduur tagab, et ükski inimene ei saa täit kontrolli privaatvõtmete ega PIN-koodide üle.

Samm	Tegevus	Kirjeldus	Vastutav
1	YubiKey eemaldamine RP-st	Pärast testimise lõppu	Operaator
2	YubiKey visuaalne kontroll	Kontrollitakse YubiKey kahjustusi	Operaator + vaatleja
3	YubiKey asetamine turvaümbrikku	Ümbrik ei tohi olla läbipaistev	Operaator
4	Ümbriku pitseerimine	*Tamper-proof* kleebis, allkiri	Operaator + vaatleja
5	PIN_A asetamine eraldi ümbrikku	Iga YubiKey kohta	Osakond A
6	PIN_B asetamine eraldi ümbrikku	Iga YubiKey kohta	Osakond B
7	PIN-ümbrike pitseerimine	*Tamper-proof* kleebised, allkirjad	Osakond A/B + vaatleja
8	USB-pulga pakkimine	Ei pea pitseerima; allkirjast piisab	Operaator
9	Kõigi ümbrike märgistamine	YubiKey #, kehtivusaeg, kuupäev	Operaator
10	Üleandmise protokolli täitmine	Sisaldab ümbrike ID-sid ja SHA256 summasid	Operaator + vastuvõtja

9.3. Turvaümbrikute märgistamine

Iga ümbrik märgistatakse minimaalselt järgmiste andmetega:

Väli	Sisu
Ümbriku sisu tüüp	YubiKey / PIN_A / PIN_B / USB mälupulk
Sertifikaadi tüüp	5 aastat või 6 aastat
Ümbriku ID	Unikaalne ümbriku seerianumber
Kuupäev	Ümbriku pitseerimise kuupäev

Väli	Sisu
Pitseerija allkiri	Füüsiline allkiri
Vaatleja allkiri	Füüsiline allkiri

10. Nõuded dokumenteerimisele

Võtmete loomise tseremoonia peab olema protokollitud viisil, mis tagab turvalise, auditeeritava ja reprodutseeritava protsessi. Käesolevas peatükis sätestatakse miinimumnõuded, millele tseremoonia protokoll peab vastama. RIA võib kasutada oma sisemisi vorme ja protseduure, tingimusel et alljärgnevad nõuded on kaetud.

Kategooria	Kohustuslik sisu	Selgitus/põhjendus
Üldinfo	- Tseremoonia kuupäev ja kellaaeg - Toimumise koht - RP seerianumber / ID - USB-mälupulga ID - SD-kaardi ID	Tagab protsessi ajalis-ruumilise auditeeritavuse ja kasutatud vahendite tuvastatavuse

Osalejad	- Kõik tseremoonias osalevad isikud
	- Rollid:
	  - tseremoonia juht,
	  - tehniline läbiviija (operaator),
	  - PIN_A valdaja 5a/6a,
	  - PIN_B valdaja 5a/6a,
	  - vaatleja(d))
	- Nimed, osakonnad
	- Allkirjad
		Tagab vastutuse, rollide eraldamise ning PIN-i kaheosalisuse nõude

Kasutatud materjalid	- Raspberry Pi mudel + ID - SD-kaart (maht, tootja) - USB-mälupulk (maht, ID) - YubiKey #1 ja #2 seerianumbrid - Ubuntu versioon + SHA256 - libccid .deb + SHA256 - pcscd .deb + SHA256 - YubiKey Manager CLI + SHA256 - PKCS#11 teegid - install.pl skript + SHA256 - gencert.pl skript + SHA256	Tagab, et kõik krüptograafilised ja tarkvaralised komponendid on terviklikud ja identifitseeritavad
Teostatud tegevused	Protokoll peab kinnitama vähemalt: - RP ettevalmistus (offline) - YubiKey #1 kontroll - YubiKey #2 kontroll - PIN_A+PIN_B sisestamine (5a ja 6a) - Sertifikaatide loomine (5a ja 6a) - Testallkirjastamine mõlemale - SHA256 kontroll - Ajutiste failide eemaldamine	Tagab jälgitavuse. Iga kriitiline tegevus peab olema dokumenteeritud kui teostatud/ebaõnnestunud

Kategooria	Kohustuslik sisu	Selgitus/põhjendus
Väljundite kirjeldus	- YubiKey #1 (5a privaatvõti slot 9c) - YubiKey #2 (6a privaatvõti slot 9c) - <nimi>.crt (5a ja 6a) - <nimi>.sha256 (5a ja 6a) - <nimi>.crt.sha256 - sha256_manifest.txt - version.txt	Tagab, et kõik väljundid on protokollis kajastatud ja kontrollitavad
PIN-koodide haldus	PIN_A ja PIN_B eraldi Eraldi ümbrikud iga sertifikaadi jaoks PIN-haldajate rollide dokumenteerimine	Kriitiline nõue: ükski inimene ei tohi teada tervet PIN-i
Allkirjastamine	Tseremoonia juht Tehniline läbiviija Vähemalt üks vaatleja Kuupäevad ja allkirjad	Kinnitab, et protseduur viidi läbi vastavalt juhendile ja tulemused on autentsed

Protseduuri dokumenteerimine protokollina ja allkirjastamine kinnitab, et:
- protseduur viidi läbi vastavalt juhendile;
- tulemused on autentsed;
- väljundid on täielikud ja terviklikud;
- protsess viidi läbi nõuetekohaselt ja turvaliselt.

11. Hävitamise protseduuri nõuded

Alljärgnevad nõuded määravad miinimumi, mida RIA peab täitma pärast võtmete loomise protseduuri lõppu. Hävitamise eesmärk on tagada, et protsessis kasutatud ajutised või potentsiaalselt taastatavad kandjad ei võimaldaks privaatvõtmete või PIN-koodide taastamist.

Kõik hävitamistoimingud tuleb dokumenteerida tseremoonia protokollis.

Kategooria	Kohustuslik nõue	Selgitus/põhjendus
Üldpõhimõtted	Hävitamine on füüsiline ja ühekordne	SD-kaart ja RP ei tohi olla korduvkasutuses; vältida taastamisvõimalusi
	Hävitamine toimub samal päeval ja samas ruumis	Vältida kandjate liigset liikumist ja turvariski
	Protsessi dokumenteerimine on kohustuslik	Auditeeritavuse tagamiseks
SD-kaardi hävitamine	SD-kaart tuleb füüsiliselt hävitada (purustamine, lõikamine, murdmine)	Tarkvaraline kustutamine ei ole lubatud
	SD-kaardi ID tuleb protokollis kajastada	Tagab jälgitavuse
	Hävitaja ja tunnistaja allkirjad protokollis	Tagab vastutuse
Raspberry Pi hävitamine	Raspberry Pi tuleb hävitada RIA sise-eeskirjade kohaselt	RP võib sisaldada tundlikke jääkandmeid/logisid

Kategooria	Kohustuslik nõue	Selgitus/põhjendus
	Hävitatavad komponendid: PCB, kontrollerid, eMMC/flash	Tehniline miinimumnõue taastatavuse vältimiseks
	RP ID ja hävitamismeetodi protokoll	Auditeeritavuse tagamiseks
Ajutiste failide eemaldamine	Kõik ajutised failid (sign-test.txt, .sig) tuleb kustutada enne RP hävitamist	Tehniline hügieen ja riskide maandamine
	Logid kopeeritakse USB-le, seejärel kustutatakse RP-st	USB jääb ainukeseks väljundkandjaks
	Tseremoonia juht kinnitab eemaldamise protokollis	Vastutus ja kontroll
Protokolli nõuded	Protokoll peab sisaldama SD-kaardi ja RP hävitamise kirjet	Kontrollpunkt auditi jaoks
	Protokoll peab sisaldama osalejate nimesid, rolle ja allkirju	Jälgitavus ja rollide eraldatus
	Hävitamise kuupäev, kellaaeg, meetod	Faktipõhine dokumentatsioon
Hävitamise eesmärk	Ainsad säilivad artefaktid: YubiKey #1 ja #2 + USB + protokoll	Kõik muu hävitatakse; privaatvõtmed jäävad ainult YubiKey'dele

Lisa 1: Installatsioonifailide sõrmejäljed

Käesolev lisa sisaldab võtmete loomise protsessis kasutatavate installatsioonifailide ja skriptide krüptograafilisi sõrmejälgi. Sõrmejälgede eesmärk on tagada kasutatavate failide terviklus ja autentsus kogu protseduuri vältel, eriti arvestades, et töö toimub täielikult võrguühenduseta keskkonnas.

Käesolevas lisas loetletud paigalduspaketid (libccid, pcscd) on juhendi koostamise ajal hangitud Ubuntu ametlikust paketiarhiivist tervikluse kontrollimise ja protseduuri valideerimise eesmärgil. Vajaduse korral on võimalik kontrollida nende pakettide uuemaid versioone samast allikast.

Ametlikud allikad:

```
https://archive.ubuntu.com/ubuntu/pool/universe/c/ccid/
https://archive.ubuntu.com/ubuntu/pool/universe/p/pcsc-lite/
```

Kõik tabelis loetletud failid tuleb enne kasutamist kontrollida vastava sõrmejälje alusel. Faili kasutamine on lubatud ainult juhul, kui arvutatud sõrmejälg vastab käesolevas lisas toodud väärtusele. Juhul kui kontroll ebaõnnestub, tuleb fail lugeda kompromiteerituks ning protseduuri ei tohi jätkata enne, kui vastav kõrvalekalle on lahendatud.

Käesolevas lisas toodud tabel moodustab lahutamatu osa tseremoonia protokollist ning viide sellele tuleb vajadusel kajastada ka tseremoonia dokumentatsioonis.

Sõrmejälje arvutamiseks tuleb kasutada käsku: `shasum -a 256 </asukoht/faili nimi>`

Faili nimi	Sõrmejälg
`install.pl`	`60430a459e6a15a4e0e939a32e3e1bcea3cbec4f65f45be6608b65995143f527`
`libccid_1.5.5-1_arm64.deb`	`028a27d13243eec5ee90063acf7111187d185a0adceffa6b7b3c15e96039eea1`
`pcscd_2.0.3-1build1_arm64.deb`	`03ac159caf6b454d4e09c1afb2f01762768a409bcc4b653ec8ae35b4e254072b`
`ykman-users.rules`	`9e14dd17ce6140273f901174e62a59b5a7a05af2431eb350282fb994a92f80e9`
`lib/libykcs11.so.2.7.2`	`b5160b0bbdeab6800371a6c108b82bcd6f7a068e567748ab6f372bf5126c5c8c`
`lib/libykpiv.so.2.7.2`	`76857886a5cc6ac810785214229aa6fd4317413ca24643dfe3c54cd468ba33a4`
`lib/pkcs11.so`	`2ed1c3fc7bc805d2145470b6a772c0bcc628662281cf7487e2990e8002c67a89`
`cert/gencert.pl`	`be91c3105d28378b273cdf4fd887ae1cfa6a12d9c7bd40bcfdd49db9e9e19299`
`cert/openssl.conf`	`4764ea57be8c83e7a79cb2daf96175bfb7097e3b8beb453300be10ce260ab254`
`cert/testcert.pl`	`5934304a408d07219456d035db807e9dea71210c9f74b6fa0dab53ac41f1dfd2`
`cert/yubico-piv-tool`	`bf92f1ee935d2399c543fcb009c4ba760ccb8420ed1eb9a277278403f580561a`

Lisa 2: yubico-piv-tool ja teekide kompileerimine

Ubunutu 24.04 ARM tööjaamas (näiteks Raspberry Pi):

1. Avage terminal
2. Vajalike pakettide installeerimiseks andke käsud:

```
sudo apt-get update

sudo apt -install git cmake libssl-dev pkg-config check libpcsclite-
dev gengetopt help2man libz-dev build-essential
```

3. Laadige alla yubico-piv-tool lähtekood:

```
git clone https://github.com/Yubico/yubico-piv-tool.git
```

4. Kompileerimiseks andke käsud:

```
cd yubico-piv-tool
mkdir build
cd build
cmake ..
make
```

5. Vajalikud failid on:

```
build/tool/yubico-piv-tool
build/lib/libykpiv.so.2.7.2
build/ykcs11/libykcs11.so.2.7.2
```

Lisa 3: OpenSSL PKCS11 mooduli kompileerimine

Ubunutu 24.04 ARM tööjaamas (näiteks Raspberry Pi):

1. Avage terminal
2. Vajalike pakettide installeerimiseks andke käsud:

```
sudo apt install git meson
```

3. Laadige alla pkcs11-provider lähtekood

```
git clone https://github.com/latchset/pkcs11-provider.git
```

   * Vajalik on GitHub-i sisselogimine
4. Kompileerimiseks andke käsud:

```
cd pkcs11-provider/
meson setup build
meson compile -C build
```

5. Vajalik fail on:

```
build/src/pkcs11.so
```