Riigi Valimisteenistus

IVXV: E-hääletamise käsiraamat

Versioon 0.6

Dokument: IVXV-KR-0.6

Kuupäev: 10.05.2019.a.

Sisukord

1. E-hääletamise seadistamine

1.1. Eesmärgid

Käesolev käsiraamat on loodud eesmärgiga kirjeldada tegevusjuhiseid e-hääletamise läbiviimiseks.

Mõnede käsiraamatus kirjeldatud protseduuride täpsem sisu on lahtiseks jäetud eeldades, et protseduuride täitjail on ülevaade kasutatavatest keskkondadest, baassüsteemidest ja infoturbe meetoditest. Eeldatakse ka, et täitjad on tuttavad või tutvuvad vastavalt vajadusele muu e-hääletamise süsteemi dokumentatsiooniga, mida käsiraamatus ei dubleerita.

Süsteemihalduse hõlbustamiseks on mõned keerukamad üldprotseduurid e-hääletamise süsteemi parameetritega käesolevas käsiraamatus detailsemalt lahti kirjutatud.

1.2. Lisanõuded

Käesolev käsiraamat järgib dokumendis „IVXV: Üldkirjeldus“ (IVXV-ÜK-*) toodud raamistikku ning kasutab seal defineeritud mõisteid. Mitmete e-hääletamise protseduurid on detailsemalt kirjeldatud toodud järgmistes materjalides:

1. „Seadistuste koostamise juhend“ (IVXV-JSK-*), edaspidi JSK;
2. „Valijarakenduse pakendamine" (IVXV-JVP-*), edaspidi JVP;
3. „Koguja süsteemihalduri juhend“ (IVXV-JSH-*), edaspidi JSH;
4. „Kogumisteenuse haldusteenuse kasutusjuhend“ (IVXV-JHT-*), edaspidi JHT.

Vastavalt vajadusele lähtutakse Riigikogu Kantselei sisekorraeeskirjast, asjaajamiskorrast, asutusesisese teabe kasutamise korrast ja Toompea lossis viibivate isikute ohuolukorras tegutsemise korrast.

Riigi Valimisteenistus (edaspidi RVT) ja rakkerühma juht kontrollivad kõikide protseduuride täitmist vastavalt vajadusele. Osade protseduuride juures (miksimine ja kõik häälte avamise võtmega seotud protseduurid) on audiitori järelevalve kohustuslik ja vaatlejate järelevalve soovitatav.

2. E-hääletamise seadistamine

2.1. Kogumisteenuse ettevalmistamine

Kogumisteenusesse installeeritakse viimane tarkvaraversioon ning seadistatakse vastavalt juhenditele JSH ja JHT. Seadistamise tulemusena tarnitakse Korraldajale digitaalselt allkirjastatuna tehniline seadistus (*`technical.yaml`) vastavalt JSK punktile 7.3. Samuti antakse Korraldajale üle konfigureerimiseks vajalikud sertifikaadid:

- valikute serveri mikroteenuse sertifikaadid (`choices.pem`);
- DDS mikroteenuse sertifikaadid (`dds.pem`);
- hääletamise mikroteenuse sertifikaadid (`voting.pem`).

Valijarakenduse (parameeter `REG_CERTS`), Kontrollrakenduse konfiguratsiooni (parameeter `tspreg_client_cert`) ja Töötlemisrakenduse jaoks (parameeter `check.tskey`) on vajalik:

- registreerimispäringute tegemise sertifikaat.

Kontrollrakenduse konfiguratsiooni jaoks (parameeter `verification_tls`) on vajalikud

- verifitseerimise mikroteenuse sertifikaadid

2.2. Valimiste konfiguratsiooni loomine

Kogumisteenus vajab järgmisi allkirjastatud konfiguratsioonifaile:

- Kogumisteenuse usaldusjuur (*`trust.yaml`) – vastavalt JSK juhendile.
- Kogumisteenuse tehniline seadistus (*`technical.yaml`) – vastavalt JSK juhendile.
- Valikute (kandidaatide) nimekiri – vastavalt dokumendile „IVXV protokollid“. Valikute nimekirja genereerib RVT Valimiste Infosüsteemis (VIS) ning nimekirja allkirjastab VIS operaator.
- Valijate nimekiri - vastavalt dokumendile „IVXV protokollid“. Valijate nimekirja genereerib rahvastikuregistri vastutav töötleja, kes edastab koos nimekirjaga selle signatuuri ning signatuuri kontrollimiseks vajaliku sertifikaadi. Korraldaja allkirjastab nimekirja ja signatuuri.
- Koguja valimiste konfiguratsioon (`*election.yaml`) vastavalt JSK juhendile.
    - Valijate nimekirjade signeerimise sertifikaadi (sektsioon `voterlist`) saab koos valijate nimekirjaga (vt eelmine punkt)
    - SK sertifikaadid (juur, ESTEID*, TSA) laetakse repositooriumist
    - Tasuta OCSP teenuse () sertifikaate pole vaja konfiguratsioonis näidata, kuna nad sisalduvad OCSP päringute vastuses ja on välja antud konfiguratsioonis näidatud sertifitseerimisasutuse (ESTEID*) poolt.

Kogumisteenuses volitatud isiku poolt allkirjastatud kogumisteenuse usaldusjuure konfiguratsioonifail (`*trust`) tuleb saata Kogujale käsurealt paigaldamiseks; ülejäänud konfiguratsioonifaile (`*technical, *elections`) ning nimekirju saab paigaldada läbi

Koguja Haldusteenuse veebiliidese. Viimane on kasutatav isikutele, kes on loetletud failis `*trust.yaml`, samuti peavad eelnimetatud failid olema allkirjastatud nende isikute poolt.

Lisaks kogumisteenusele on vajalikud allkirjastatud konfiguratsioonifailid ka Korraldaja, Töötleja, Lugeja ja Audiitori funktsioonide täitmiseks, täpsemalt:

- Rakenduste usaldusjuur (`ivxv.properties`) – vastavalt JSK juhendile.
- Võtmerakenduse konfiguratsioon – võib koostada ühe failina, kuid on võimalik ka eraldi vastavalt operatsioonile (vt täpsemalt JSK juhend):
    - Hääle salastamise võtme spetsifikatsioon (`key.groupgen.yaml`). Seda kasutatakse juhul, kui on soov genereerida uued jäägiklassi parameetrid (p,q), mida kasutatakse häälte salastamise võtme loomisel. Tavaliselt on soovitatav kasutada standardis RFC3526 toodud parameetreid.
    - Häälte salastamise võtme loomise konfiguratsioon (`key.init.yaml`).
    - Võtmerakenduse täiendavate tööriistade konfiguratsioon (`util`) ja salajase võtme testimise konfiguratsioon (`testkey`), soovitatavalt kirjutada need sektsioonid samasse faili `init`-failiga (vt eelmine punkt).
    - Häälte salastamise võtme konfiguratsioon (`key.decrypt.yaml`). Võib koostada ka peale töötlemisetappi ning mitmes variandis – tõestuse genereerimisega ja ilma.
- Töötlemisrakendusele vajalikud konfiguratsioonifailid (`processor.*.yaml`). Võib koostada ka vahetult enne töötlemisetappi.
- Auditirakendusele vajalikud konfiguratsioonifailid (`auditor.*.yaml`). Võib koostada ka vahetult enne auditeerimist.
- Valimisjaoskondade ja –ringkondade nimekiri – vastavalt dokumendile „IVXV protokollid“. Nimekirja genereerib RVT Valimiste Infosüsteemis (VIS) ning nimekirja allkirjastab VIS operaator.

Kõik rakenduste konfiguratsioonifailid võivad olla allkirjastatud suvalise isiku poolt. Allkirjastaja nime ja isikukoodi näidatakse rakenduse käivitamisel.

Hääle kontrollimiseks nutiseadme abil on vajalik koostada kontrollrakenduse konfiguratsioon vastavalt JSK juhendile, pidades silmas järgevat:

- PEM-kujulised sertifikaadid esitatakse ühes reas, kusjuures
    - stringi -----BEGIN CERTIFICATE----- **järel** tuleb asetada \n
    - stringi -----END CERTIFICATE----- **ette** tuleb asetada \n

Sama kehtib avaliku võtme esituse kohta (BEGIN/END PUBLIC KEY).

Kontrollrakenduse konfiguratsioon paigaldatakse veebiserverisse varem kontrollrakenduse arendajaga kokkulepitud asukohta.

2.3. Süsteemi võtmepaari genereerimine

Süsteemi võtmepaari genereerimine on auditeeritav protseduur. Süsteemi võtmepaar genereeritakse eraldi võrgust lahti ühendatud arvutis, millel on eemaldatud sisemised salvestusvahendid (v.a. andmete välisele andmekandjale kirjutamist võimaldav seade) ning

mis alglaaditakse väliselt kõvakettalt. Sellisel moel on võimalik audiitoritel ja vaatlejatel veenduda, et süsteemis ei sisaldu pahavara, mis häälte avamise võtit salvestab või kasutab. Kui välist kõvaketast ei kasutata, säilitakse seda turvakleebisega/turvakotis pitseerituna. Mälupulga kasutamine andmevahetuseks selle arvutiga, milles genereeriti süsteemi võtmepaar, on keelatud.

Kõvakettale on installeeritud operatsioonisüsteem (Windows 10) ning järgmised rakendused:

- Java 1.8 või uuem.
- Kaarditootja tööriistad[1]:
    - MyEID Minidriver Utility
    - MyEID Pin Tool
- Mäluketta (RAM disk) utiliit [2]
- OpenSC toolkit kiipkaartidega manipuleerimiseks[3]
- DigiDoc Client konfiguratioonifailide vaatamiseks[4]

Häälte avamise võtme osakud säilitatakse kiipkaartidel, mis iga kasutamise järel pitseeritakse turvakleebisega. Enne kasutust kontrollib audiitor turvakleebise terviklust.

Operatsioonid andmetega viiakse läbi virtuaalsel mälukettal. Toimingu väljund kirjutatakse välisele andmekandjale.

Süsteemi võtmepaari genereerimine koosneb järgmistest etappidest:

1. **Konfiguratsiooni ja võtmerakenduse ettevalmistamine.**
   Veendutakse rakenduste usaldusjuure konfiguratsiooni (`appconf.bdoc/ivxv.properties`) ja võtmerakenduse konfiguratsiooni (`key.init.yaml`) korrektsuses, allkirjastatakse ning kirjutatakse koos võtmerakendusega välisele andmekandjale, olles eelnevalt veendunud võtmerakenduse autentsuses ja tervikluses.

2. **Mäluketta loomine.**
   Vahendiga ImDisk luuakse 2048 MB suurune mäluketas, kusjuures tuleb tähele panna, et oleksid valitud „Quick Format“ ja „Use AWE physical memory“.
   Võtmerakendus paigutatakse mälukettale.

3. **Konfiguratsiooni ja võtmerakenduse import**
   Punktis 1 loodud väliselt andmekandjalt kopeeritakse konfiguratsioon ja võtmerakendus mälukettale.

4. **Kiipkaartide ettevalmistamine.**
   Kaardid valmistatakse ette, kasutades ühte arvutiga ühendatud kaardilugejat. Kasutusele võetavad kiipkaardid tuleb alglähtestada vahendi MyEID Minidriver

[1] Vt http://www.aventra.fi/downloads
[2] Vt http://www.ltr-data.se/opencode.html/#lmDisk
[3] Vt https://osdn.net/projects/sfnet_opensc/downloads/OpenSC/opensc-0.12.2/OpenSC-0.12.2-win64.msi/
[4] Vt https://installer.id.ee/

Utility abil käsuga „Initialize card“. Tuleb veenduda, et oleksid tehtud järgmised valikud:

User PIN – kaardi PIN kood 1111
User PUK – kaard PUK kood 12345678
Challenge/Response - Mitteaktiivne
Administrator PIN – 999999
Administrator PUK – 12345678
Activate applet – mitteaktiivne (vt allpool)
Create msroots file – mitteaktiivne

NB! Activate applet määrab ära, kas kaardi PIN koodi küsimine on aktiivne või mitte.

5. **Kaardilugejate ühendamine ja numereerimine.**
   Arvutiga ühendatakse 9 kaardilugejat ning numereeritakse need 0..8, kasutades ühte kiipkaarti ning veendudes, millises kaardilugejas parajasti kaart on käsuga:
   ```
   key util -c appconf.bdoc --listreaders
   ```

6. **Võtmepaari genereerimine**
   Kiipkaardid sisestatakse kaardilugejatesse ning sisestatakse käsk:
   ```
   key init -c appconf.bdoc -p key.init.bdoc
   ```
   Kaartidele genereeritud identifikaatoreid on näha „lisreaders“ käsuga. Kaartidele kirjutatakse füüsiliselt peale järjekorranumber ning jagatakse protseduuri lõpus laiali võtmeosakute hoidjate vahel. Fikseeritakse kirjalikult, kellele millise numbriga kaart anti.

   Lisaks privaatvõtme osakutele kiipkaartidel, genereeritakse vastavad avalikud võtmed, mida kasutatakse järgnevalt:
   - `*pub.der` – kasutamiseks valijarakenduses
   - `*pub.pem` – kasutamiseks kontrollrakenduse konfiguratsioonis (või valijarakenduses)
   - `*sign.pem` – kasutamiseks valimistulemuse signatuuri kontrollil
   - `*enc.pem` – ei kasutata

7. **PIN-ide valik**
   Soovi korral võib kaartidele omistada PIN-koodid. Selleks tuleb kaardid aktiveerida „Activate applet“ valikuga (vt p.4). PIN-koodide vahetamiseks on kasutatavvahendit MyEID Pin Tool. PIN-i vahetamise järel tuleb käsurealt kasutada käsku:
   ```
   pkcs15-init --finalize
   ```

8. **Kiipaartide testimine**
   Veendumaks kiipkaartide toimivuses, viiakse läbi proovi-dekrüpteerimine mitme erineva kombinatsiooniga üheksast kaardist. Testimisel küsitakse PIN-koode juhul, kui need on määratud. Testimiseks kasutatakse käsku:
   ```
   key testkey -c appconf.bdoc -p key.init.bdoc
   ```

9. **Avalike võtmete varundamine.**
   Genereeritud avalikud võtmed (*.pem, *.der) kirjutatakse välisele andmekandjale.

2.4. Valijarakenduse pakendamine

Valijarakendus pakendatakse vastavalt JVP juhendile, pidades silmas järgnevat:

- Kui ühele väljale läheb mitu sertifikaati, siis tuleb nendest sertifikaatidest eelnevalt moodustada ühine tekstifail. Alternatiiviks on sertifikaatide ükshaaval lisamine, toetatud on nii PEM kui DER vormingus sertifikaadid.
- Sertifikaadiväljadel DDS, CHOICES ja VOTING võib masinate sertifikaatide asemel kasutada neid väljaandva sertifitseerija oma, sellisel juhul pole masinate sertifikaadid vajalikud.
- Valimisringkondade ja –jaoskondade nimekiri lisatakse allkirjastamata kujul.
- Häälte salastamise võti (*pub.{der.pem}) laetakse väliselt andmekandjalt, mis valmendati võtmepaari genereerimise käigus.

RVT vaatab üle kõik valijarakenduse tekstid ning uuendab neid vajadusel. Seejärel rakendatakse konfiguratsioon kõikidele valijarakendustele (Windows, macOS, Linux 32/64-bit).

Windowsi Valijarakendus saadetakse signeerimiseks Arendajale või Kogujale. MacOS rakendus saadetakse pakendamiseks ja signeerimiseks Arendajale. Linuxi valijarakendust ei allkirjastata

Valijarakendused avalikustatakse veebiserveris vahetult enne e-hääletamise algust koos digitaalselt allkirjastatud sõrmejälgede failiga.

2.5. Süsteemi testläbimine

Süsteemi võtmepaari genereerimine on auditeeritav protseduur. Süsteemi testläbimiseks käivitatakse süsteem piiratud moel ja kontrollitakse konfiguratsiooni ja nimekirjade kooskõlalisust. Selleks muudetakse hääletamise algus- ja lõpuaega (fail `election.yaml`) ning luuakse võimalus hääletada piiratud ruumist. Hääletajate valikud protokollitakse. Testläbimise hääled töödeldakse ja loetakse kokku ning võrreldakse protokollitud valikutega.

Testläbimise lõpus taastatakse testläbimise eelne seis.

Koguja lähtestamise ja konfigureerimise järel tuleb Kogujal teha süsteemist tõmmis (*snapshot*), mis võimaldab süsteemi taastet. Esialgsesse konfiguratsiooni pannakse hääletamise ajaks testläbimise aeg.

Et olla valmis olukorraks, kus valikute (kandidaatide) nimekiri peale süsteemi esialgset seadistamist ja testläbimist muutub, tuleb teha kaks süsteemi tõmmist: esimene enne valikute nimekirja laadimist (tõmmis 1) ja teine peale seda (tõmmis 2). Kui peale testläbimist valikute nimekiri muutub, siis lähtestatakse süsteemi peale testläbimist 1. tõmmisest ning laetakse uus nimekiri. Vastasel korral lähtestatakse süsteem 2. tõmmisest. Süsteemi lähtestamisel määratakse ka uued (tegeliku) hääletamise ajad.

3. Häälte töötlemine

Häälte töötlemine toimub andmeside võrku mitte ühendatud arvutis. Kõikide protsesside sisendandmed loetakse sisse väliselt andmekandjalt. Väljund kirjutatakse samuti välisele andmekandjale. Enne väljundi kirjutamist arvutatakse väljundi sõnumilühend, viiakse see mälupulgal mõnesse Internetti ühendatud arvutisse, allkirjastatakse ning viiakse mälupulgaga allkirjastatud sõnumilühend tagasi töötlemiseks kasutatavasse arvutisse.

Häälte töötlemise protsessid viiakse läbi mälukettal. Töötleja rolli kannab VVK otsusega määratud organisatsioon.

3.1. Häälte tervikluse kontroll ja korduvhäälte tühistamine

Protseduur viiakse läbi pärast e-hääletamise perioodi lõppu.

E-hääletamise perioodi lõppemisel annab:

- Koguja Töötlejale üle välisele andmekandjale kirjutatud logid ja e-urni häältega, mille sõnumilühend on allkirjastatud Kogumisteenuse pakkuja esindaja poolt;
- Registreerimisteenus Töötlejale üle ajatemplid, mille sõnumilühend on digitaalselt allkirjastatud teenuseosutaja poolt.

Esmalt kontrollitakse Kogumisteenuses salvestatud e-häälte vastavust Registreerimisteenuses fikseeritud häältega ning häälte digitaalallkirjade terviklust. Soovi korral on võimalik digitaalselt allkirjastatud hääli kõrvutada ka valijate nimekirjaga.

Protseduuriks on vaja järgmisi andmeid:

- Häälte e-urn koos allkirjastatud sõnumilühendiga
- Ajatemplid koos allkirjastatud sõnumilühendiga
- Jaoskondade ja ringkondade nimekiri
- Avalik võti, mille alusel signeeris Koguja oma päringuid Registreerimisteenusele. Üldjuhul tuleb see tekitada vastavast sertifikaadist käsuga:
  `openssl x509 -in TEST2017tspreg.pem  -noout -pubkey -out tspkey.pem`
- Valijate nimekirjad (algne nimekiri ning nimekirja uuendused) koos signatuuridega
- Avalik võti valijate nimekirja signatuuride kontrollimiseks
- Rakenduste usaldusjuure konfiguratsioon (`ivxv.properties`)

Protseduur koosneb järgmisest sammudest:

1. Veendutakse rakenduse konfiguratsioonifaili (`processor.yaml`) korrektsuses (vt JSK juhend). Korrektsust tuleb kontrollida vähemalt faili kahe esimese osa kohta („`check`“ ja „`squash`“). Konfiguratsioonifail allkirjastatakse digitaalselt.
2. Mäluketas luuakse käskudega:
   ```
   sudo mkdir /mnt/ramdisk
   sudo mount –t tmpfs –o size=3000m tmpfs /mnt/ramdisk
   ```
3. Kõik vajalikud andmed koos töötlemisrakendusega kantakse üle töötlemiseks kasutatava arvuti vastloodud mälukettale.
4. Kontroll viiakse läbi käsuga:

```
processor check -c appconf.bdoc -p processor.bdoc
```

5. Väljundkataloogi tekkinud kontrollsumma allkirjastatakse
6. Korduvad hääled tühistatakse käsuga
```
processor squash -c appconf.bdoc -p processor.bdoc
```
7. Väljundkataloogi tekkinud kontrollsumma allkirjastatakse
8. Mõlemate protsesside väljundkataloogid kirjutatakse välisele andmekandjale. Teise, squash protsessi väljund sisaldab ka e-hääletanute nimekirja trükkimiseks mõeldud PDF-kujul ja VIS-i sisestamiseks mõeldud JSON-kujul.

3.2. Topelthäälte tühistamine ja anonüümistamine

Lisaks korduvalt antud e-häältele tuleb tühistada ka nende valijate e-hääled, kes hääletasid nii elektrooniliselt kui valimisjaoskonnas eelhääletamise ajal. Tühistusnimekirja väljastab VIS operaator ning allkirjastab selle digitaalselt.

Tühistusnimekiri väljastatakse valimispäeval vahetult enne häälte kokkulugemist. Topelthäälte tühistamise protsessiks vajalikud sisendid on:

- Korduvhäältest puhastatud e-urn koos allkirjastatud sõnumilühendiga
- Jaoskondade ja ringkondade nimekiri
- Allkirjastatud tühistusnimekiri
- Häälte salastamise võti
- Rakenduste usaldusjuure konfiguratsioon (`appconf.bdoc/ivxv.properties`)

Tühistamise ja anonüümistamise protseduur koosneb järgmisest sammudest:

1. Veendutakse rakenduse konfiguratsioonifaili (`processor.yaml`) korrektsuses (vt JSK juhend). Kontrollida tuleb faili osasid „`revoke`“ ja „`anonymize`“. Konfiguratsioonifail allkirjastatakse digitaalselt.
2. Kõik vajalikud andmed koos töötlemisrakendusega kantakse üle töötlemiseks kasutatava arvuti mälukettale.
3. Tühistamine viiakse läbi käsuga:
```
processor revoke -c appconf.bdoc -p processor.bdoc
```
4. Väljundkataloogi tekkinud kontrollsumma allkirjastatakse.
5. Hääled anonüümistatakse käsuga
```
processor anonymize -c appconf.bdoc -p processor.bdoc
```
6. Väljundkataloogi tekkinud kontrollsumma allkirjastatakse. Kui hääled loetakse koheselt kokku, pole allkirjastamine vajalik (vt ptk 4).
7. Mõlema protseduuri väljundkataloogid koos sisuga kirjutatakse välisele andmekandjale.

4. Häälte kokku lugemine

Häälte kokkulugemine on auditeeritav protseduur.

Hääli loetakse kokku kahel viisil:

1. Kui muud valimisprotseduurid ei võimalda piisavat miksimiseks vajalikku ajaperioodi, loetakse miksimata hääled kokku valimispäeva õhtul ilma lugemistõendit väljastamata. Sel juhul loetakse miksitud hääled kokku valimispäevale järgneval päeval ja väljastatakse ka lugemistõend.
2. Miksitud hääled loetakse kokku valimispäeval. Miksitud häälte kokkulugemise järel väljastatakse ka lugemistõend.

Hääled loetakse kokku samas keskkonnas, kus genereeriti süsteemi võtmepaar (vt p.2.3). Kokkulugemisel kasutatakse ainult ühte kiipkaardilugejat.

4.1. Häälte kokkulugemine ilma miksimiseta

Häälte kokkulugemiseks on vajalikud järgmised sisendid:

- Anonüümistatud (miksimata või miksitud) häältega e-urn koos allkirjastatud sõnumilühendiga
- Jaoskondade ja ringkondade nimekiri
- Valikute nimekiri
- Konfiguratsioonifailid (rakenduste usaldusjuur, võtmerakenduse konfiguratsioon)

Protseduur koosneb järgmisest sammudest:

Ettevalmistav osa (mitte-auditeeritav):

1. Veendutakse rakenduse konfiguratsioonifaili (`*key.yaml`) korrektsuses (vt JSK juhend). Parameeter „`provable`“ peab olema kas „`false`“ (tõestust ei genereerita, kasutatakse esmasel ilma miksimata lugemisel) või „`true`“ (tõestus genereeritakse, kasutatakse miksitud lugemisel).
2. Konfiguratsioonifail allkirjastatakse digitaalselt.
3. Veendutakse võtmerakenduse autentsuses ja tervikluses.
4. Kõik vajalikud andmed koos võtmerakenduse ja vajalike utiliitidega kirjutatakse välisele andmekandjale.

Lugemisosa

Häälte kokkulugemine viiakse läbi võtmeprotseduurideks mõeldud arvutis ning on auditeeritav protseduur järgmiste elementidega:

5. Punktis 4 loodud väliselt andmekandjalt kantakse andmed ja programmid üle mälukettale.
6. Ühendatakse kaardilugeja ja kontrollitakse selle tööd käsuga:
   ```
   key util -c *_ivxv.asice --listreaders
   ```
7. Hääte kokkulugemiseks sisestatakse käsk:
   ```
   key decrypt -c *_ivxv.asice -p *_key_f.asice
   ```
   Protsess käigus küsitakse viite erinevat võtmeosakutega kiipkaarti.

8. Kontrollitakse valimistulemuste sisu ja selle signatuuri.
9. Protsessi väljund (`decout`) ja logid (`log`) kirjutatakse välisele andmekandjale.

Valimistulemuse autentsust ja terviklust on võimalik kontrollida häälte lugemise käigus loodud signatuurfaili abil, kasutades lisaks genereerimisprotsessis loodud faili „*`sign.pem`“ (vt p.3.3). Kontrollida saab utiliidiga `openssl`, mille versioon peab olema vähemalt 1.0. Kontrollimiseks:

a) eraldatakse avalik võti:
   `openssl x509 -in sign.pem -noout -pubkey > sign.pub`
b) Kontrollitakse signatuuri:
   `openssl dgst -sha256 -sigopt rsa_padding_mode:pss -sigopt rsa_pss_saltlen:32 -sigopt rsa_mgf1_md:sha256 -verify sign.pub -signature *.tally.signature *.tally`

Häälte kokkulugemisel saadud failid (hääletustulemused, signatuur, avalik võti) koos LOEMIND-failiga pakendatakse ZIP-konteinerisse ja avaldatakse veebilehel.

4.2. Miksimine

Miksimine on auditeeritav protseduur. Protsess viiakse läbi arvutis, mida kasutati häälte töötlemiseks. Eelnevalt tuleb miksimisrakendus ette valmistada vastavalt JSK juhendile.

Miksimise sisendiks on :

- Anonüümistatud häältega e-urn
- Häälte salastamise võti (*`pub.pem`)

Protseduur koosneb järgmisest sammudest:

1. Sisendid kantakse üle töötlemiseks kasutatava arvuti mälukettale.
2. Miksimine viiakse läbi käsuga:
   `mix.py --pubkey pub.pem --ballotbox bb-4.json --shuffled` *`shuffled.json`* `--proof-zipfile proof.zip shuffle`
3. Miksitud e-urni SHA256 kontrollsumma allkirjastatakse digitaalselt ning kirjutatakse koos väljundite enestega (`shuffled.json` ja `proof.zip`) välisele andmekandjale. Kirjutamine on otstarbekas läbi viia pärast häälte teisendamise ja miksimise korrektsuse kontrollide (punktid 5.2 ja 5.3) läbimist.

NB! Miksimisrakendus ei toimi, kui miksimiseks kasutatava arvuti nimes sisalduvad suured tähed.

Vajadusel on võimlik muuta miksimisrakenduse logi detailsemaks asendades mix.py failis

`logging-basicConfig(level=logging.INFO…`

asemel

`logging-basicConfig(level=logging.DEBUG…`

4.3. Häälte teisendamise kontrollimine

Teisendust kontrollitakse keskkonnas, kus tagatakse sisendi (miksimata krüptogrammide) konfidentsiaalsus.

Häälte teisendamise kontrolli sisendid on:

- miksimise-eelne e-urn (miksimata hääled)
- miksimise-järgne e-urn (miksitud hääled)
- Häälte salastamise võti (*`pub.pem`)
- Miksimisrakenduse väljund ehk miksimise tõestus (*`proof.zip` fail)
- Rakenduste usaldusjuure konfiguratsioon (`appconf.bdoc/ivxv.properties`)

Protseduur koosneb järgmisest sammudest:

1. Kõik sisendandmed koos auditirakendusega kantakse üle töötlemiseks kasutatava arvuti mälukettale.
2. Miksimisrakenduse väljund zip pakitakse lahti audiitorrakenduse bin/ kataloogi
3. Veendutakse rakenduse konfiguratsioonifaili (auditor.yaml) korrektsuses (vt JSK juhend). Konfiguratsioonifail allkirjastatakse digitaalselt.
4. Kontroll viiakse läbi käsuga:

```
auditor convert -c appconf.bdoc -p auditor.bdoc
```

4.4. Miksimise korrektsuse kontrollimine

Miksimise korrektsust kontrollitakse keskkonnas, kus tagatakse sisendi (miksimata krüptogrammide) konfidentsiaalsus.

Protsessi läbiviimiseks on kaks võimalust – IVXV auditirakendus või Verificatumi vahend. Mõlema vahendi kasutamine peab andma sama tulemuse, tuleb aga silmas pidada, et Verificatumi vahend on ca 18x kiirem.

Kontrollimiseks **Verificatumi vahendiga** tuleb ette valmistada keskkond vastavalt JSK juhendile. Kasutatakse käsku:

```
mix.py verify --proof-zipfile proof.zip
```

NB! Kontroll tuleb läbi viia kataloogis, mis erineb miksimisel kasutatud kataloogist.

Kontrollimiseks **IVXV auditirakendusega** tuleb ette valmistada auditirakenduse konfiguratsioonifail (`auditor.yaml`) sektsioon „`mixer`“ vastavalt JSK juhendile.

Protseduur koosneb järgmisest sammudest:

- Miksimistõend (`proof.zip`) tuleb lahti pakkida
- Konfiguratsioonis (`auditor.yaml`) tuleb näidata „`protinfo`“ väljal faili protokollifaili `prot.xml` asukoht ning „`proofdir`“ väljal kataloogi `mixnet/`.
- Konfiguratsioonifail allkirjastatakse digitaalselt.

Lisaks on vajalik rakenduste usaldusjuure konfiguratsioon (`appconf.bdoc/ivxv.properties`).

Verifitseerimine viiakse läbi käsuga:

```
auditor mixer -c appconf.bdoc -p auditor.bdoc
```

4.5. Miksitud häälte kokkulugemine

Häälte kokkulugemine on auditeeritav protseduur. Miksitud hääled (`shuffled.json`) koos allkirjastamise kontrollsummaga loetakse kokku koos lugemistõendi genereerimisega. Kui varem loeti hääled kokku miksimata kujul, siis võrreldakse miksitud häälte kokkulugemise tulemust miksimata häälte kokkulugemise tulemustega. Tulemused peavad olema identsed.

4.6. Kokkulugemise korrektsuse kontroll

Kokkulugemise kontrolliks kasutatakse auditirakendust. Protsessi võib läbi viia suvalises arvutis kartmata andmekadu, kuna sisendiks kasutatavad krüptogrammid on miksitud ja väljund avalik.

Kokkulugemise korrektsuse kontrolli sisendid on:

- Häälte teistkordsel lugemisel genereeritud lugemistõend (fail „`proof`“)
- Häälte salastamise võti (`pub.pem`)
- Rakenduste usaldusjuure konfiguratsioon (`appconf.bdoc/ivxv.properties`)

Protseduur koosneb järgmisest sammudest:

1. Veendutakse rakenduse konfiguratsioonifaili (`auditor.yaml`) korrektsuses (vt JSK juhend). Konfiguratsioonifail allkirjastatakse digitaalselt.
2. Kõik vajalikud andmed koos auditirakendusega kantakse üle töötlemiseks kasutatavasse arvutisse
3. Kontroll viiakse läbi käsuga:
   ```
   auditor decrypt -c appconf.bdoc -p auditor.bdoc
   ```

Vigased tõestused kirjutatakse konfiguratsioonifailis määratud faili.

5. Valimispäeva järgsed protseduurid

Kui muud valimisprotseduurid ei võimalda piisavat miksimiseks vajalikku ajaperioodi ning hääled on miksimata kokku loetud valimispäeval, loetakse hääled üle valimispäevale järgneval päeval. Sel juhul hääled eelnevalt miksitakse ning loetakse kokku koos lugemistõendi genereerimisega. Miksimis- ja lugemistõendeid verifitseeritakse audiitorirakendusega. Pärast lugemist ja kõikide korrektsuskontrollide läbimist kinnitab RVT juht e-hääletamise tulemused.

Protseduurid on otstarbekas läbi viia järgmistes seadmetes:

- Miksimine, häälte teisendamise korrektsuse kontroll ja miksimise korrektsuse kontroll(id) ühtse protseduurina samas arvutis: kõik vajalikud sisendfailid nendeks protseduurideks komplekteerida korraga.
- Häälte teistkordne kokku lugemine viiakse läbi võtmerakenduse käitamiseks mõeldud arvutis.
- Häälte kokku lugemise korrektsust võib kontrollida suvalises arvutis.

5.1. Süsteemi võtmepaari hävitamine

Süsteemi võtmepaari hävitamine on auditeeritav protseduur.

RVT säilitab võimekust elektroonilisi hääli uuesti kokku lugeda ühe kuu jooksul valimispäevast arvates. Pärast nimetatud tähtaja möödumist, kuid mitte enne, kui esitatud kaebuste kohta on tehtud lõplikud otsused, hävitab elektrooniliste häälte avamise võtme, mis muudab elektrooniliste häälte kokku lugemise võimatuks.

Häälte avamise võtme osakud on salvestatud kiipkaartidele. Kiipkaardid, mis on Vabariigi Valimiskomisjoni liikmete ja RVT töötajate valduses, kogutakse kokku ja hävitatakse füüsiliselt.

Samuti tuleb hävitada füüsiliselt väline kõvaketas, mida on kasutatud võtmetoimingute käigus

Redaktsioonide ajalugu

Kuupäev	Versioon	Kirjeldus ja muudatused	Autor
18.09.2017	0.1	Valimiste seadistamise osa	Tarvi Martens
22.09.2017	0.2	Esimene kompilatsioon	Tarvi Martens
26.09.2017	0.3	Läbivad täpsustused	Tarvi Martens
21.01.2019	0.4	Läbivad täpsustused vastavalt DEMO2018 kogemustele. Uus protsess: häälte teisendamise kontroll (8.2)	Tarvi Martens
30.01.2019	0.5	Arendaja dokumentatsiooni ja märkuste poolt põhjendatud muudatused	Tarvi Martens
09.05.2019	0.6	Läbivad täpsustused, hääletamisele eelneva perioodi tegevuste eemaldamine (varasem 2.ptk).	Epp Maaten, Tarvi Martens