IVXV protokollide kirjeldus

Spetsifikatsioon

Versioon 1.7.6

27.09.2021

53 lk

Dok IVXV-PR-1.7.6

Sisukord

PEATÜKK 1

Annotatsioon

Käesolev dokument kirjeldab elektroonilise hääletamise infosüsteemi IVXV protokollistikku.

Dokument annab üldise ülevaate elektroonilise hääletamise süsteemi tehnilisest ülesehitusest ja kasutatavatest protokollidest. Dokumendis defineeritakse protokollides kasutatavad ühised mõisted ja andmestruktuurid.

PEATÜKK 2

Ülevaade

Elektroonilise hääletamise protokollistik (edaspidi protokollistik) defineerib elektroonilise hääletamise süsteemi komponentide vahelise sõnumivahetuse, kasutatavad andmestruktuurid, algoritmid ning liidesed väliste süsteemidega. Sõnumivahetus esitatakse UML suhtlusskeemidena, mis üheselt defineerivad sõnumite järgnevuse. Andmestruktuuride kirjeldused on varustatud Backus-Naur või JSON-schema notatsioonis spetsifikatsioonidega. Andmestruktuuride väljade eraldajateks kasutatakse reavahetusmärki `LF` (ASCII-kood `0x0A`) ja tabeldusmärki `TAB` (ASCII-kood `0x09`). Algoritmid esitatakse pseudokoodina.

NB! Kõigis protokollistiku andmestruktuuride väljades tuleb rangelt kinni pidada lubatud märkidest ning väljade minimaalsetest ja maksimaalsetest pikkustest. Täiendavate tühikute, tabulaatorite jms. kasutamine on keelatud ning spetsifikatsiooni realiseerivad rakendused peavad vorminguga mitte-vastavate andmete töötlemisest keelduma.

Protokollistik defineerib elektroonilise hääletamise protokolli ning selle protokolli realiseerimiseks vajalikud tugistruktuurid.

2.1 Elektroonilise hääletamise protokoll

Elektroonilise hääletamise protokoll spetsifitseerib:

1. elektroonilise hääle vormingu, mis võimaldab üheselt määratleda valija tahte konkreetsel valimisel;
2. elektroonilise hääle krüpteerimise hääle salajasuse tagamiseks;
3. elektroonilise hääle digitaalse allkirjastamise tervikluse ja valija identifitseerimise tagamiseks;

4. elektroonilise hääle kvalifitseerimise kogumisteenuse poolt, hääle vastuvõtmise tähistamiseks;

Protokoll eeldab, et valimise korraldaja on defineerinud valimise ning genereerinud häälte salastamise võtmepaari, mille avalik komponent on tehtud valijarakendusele kättesaadavaks.

Protokolli vahendusel liigub valija tahe kogumisteenuses talletatavasse e-valimiskasti ning võetakse tulemuse kujunemisel arvesse järgmist sündmusterida pidi:

1. Valija kasutab valijarakendust oma tahteavalduse elektrooniliseks vormistamiseks:
    1. tahteavaldus vormistatakse elektroonilise häälena;
    2. vormistatud hääl krüpteeritakse;
    3. krüpteeritud hääl allkirjastatakse digitaalselt.
2. Kogumisteenus talletab elektroonilise hääle:
    1. digitaalselt allkirjastatud häälele võetakse valija sertifikaadi kehtivust kinnitavad elemendid;
    2. elektrooniline hääl registreeritakse välises registreerimisteenuses;
    3. valijale võimaldatakse kvalifitseeritud elektroonilise hääle kontrollimine kontrollrakenduse abil.
3. Valija võib kasutada kontrollrakendust veendumaks oma hääle korrektses käitlemises kogumisteenuse poolt;
4. Hääletamisperioodi lõppedes väljastab kogumisteenus valimise korraldajale e-valimiskasti ning registreerimisteenus loendi kogumisteenuse poolt registreeritud häältest;
5. Valimise korraldaja arvutab hääletamistulemuse:
    1. veendutakse, et kõik registreerimisteenuses registreeritud hääled on e-valimiskasti koosseisus üle antud;
    2. eraldatakse krüpteeritud hääled ja digitaalallkirjad;
    3. dekrüpteeritakse krüpteeritud hääled;
    4. dekrüpteeritud häälte põhjal arvutatakse hääletamistulemus.

Protokoll on analoogne paberil posti teel hääletamise protokolliga, kus valija tahe liigub valimiskomisjonini kahes ümbrikus – välimise ümbriku sees on sisemine ümbrik, mis omakorda sisaldab valija tahteavaldusega hääletussedelit. Välimine ümbrik kannab valijat identifitseerivat infot ning võimaldab mh. kontrollida valija õigust hääletada. Sisemine ümbrik on anonüümne ning kaitseb hääle salajasust. Enne häälte kokkulugemist eraldatakse sisemised ümbrikud välimistest.

Elektroonilise hääletamise kontekstis on sisemine ümbrik vormistatud krüpteeritud häälena ning välimine ümbrik digitaalselt allkirjastatud dokumendina.

PEATÜKK 3

Valimise definitsioon

Valimise defineerib valimise korraldaja. Eesti riiklikel valimistel jagunevad kõik hääleõiguslikud isikud ühte või mitmesse valimisringkonda. Valijal on võimalik hääletamisel valida ainult selle ringkonna kandidaatide vahel, kuhu ta kuulub.

Valimise defineerimiseks tuleb määratleda vähemalt

1. valimise unikaalne identifikaator ning küsimuste unikaalsed identifikaatorid;
2. täielik loend valimisringkondadest ja -jaoskondadest;
3. hääleõiguslike isikute nimekiri ja jagunemine valimisringkondadesse;
4. kandidaatide nimekiri ja jagunemine valimisringkondadesse.

Valimise sisendandmed koostatakse Valimise Infosüsteemis (VIS), vormingukirjeldused on spetsifitseeritud VIS ja EHS ühisspetsifikatsioonis[1].

[1] https://github.com/e-gov/VIS3-EHS/

PEATÜKK 4

Elektrooniline hääl

IVXV hääletamisprotokoll põhineb topeltümbrikuskeemil, mis tähendab, et valija avakujul tahteavaldus krüpteeritakse valimise korraldaja poolt levitatud avaliku võtmega. Krüpteeritud tahteavaldus allkirjastatakse digitaalselt valija käsutuses oleva allkirjastamisvahendiga ning edastatakse kogumisteenusesse mingis kokkulepitud konteinervormingus. Kogumisteenus võib valija poolt allkirjastatud häält täiendavalt kvalifitseerida, veendudes näiteks allkirjastamissertifikaadi kehtivuses. IVXV protokollistik näeb mh. ette kogumisteenuse poolt vastuvõetud häälte registreerimise välises registreerimisteenuses.

Kogumisteenuse poolt talletamisele võetud hääl koos kvalifitseerivate elementidega tehakse kättesaadavaks nii valijarakendusele kui kontrollrakendusele, mis teostavad üksiku hääle peal samad kontrollid, mida hilisem valimise korraldaja töötlemisrakendus teostab kõigi häälte peal. Kvalifitseerivate elementide kontrollimise võimalus annab valijale kindluse, et tema häält on hilisemates protsessides korrektselt menetletud.

4.1 Valija tahteavaldus avakujul

Valija tahteavaldus avakujul eksisteerib valijarakenduses ning hiljem ka kontrollrakenduses. Tahteavaldus sisaldab nii valiku koodi ringkonnas, ringkonna EHAK-koodi kui ka valiku nimekirja nime ning konkreetse valiku nime nimekirjas.

```
choice-name = 1*100UTF-8-CHAR
choicelist-name = 1*100UTF-8-CHAR

ballot = district-choice '\x1F' choicelist-name '\x1F' choice-name
```

4.2 Krüpteeritud sedel

Valija tahteavaldus avakujul `ballot` krüpteeritakse valijarakenduse poolt valimise korraldaja genereeritud avaliku võtmega. IVXV vajab krüpteerimiseks mittedeterministlikku, homomorfset avaliku võtme krüptosüsteemi. Selliseks süsteemiks sobib ElGamal krüptosüsteem, mida täna rakendatakse IVXV kontekstis jäägiklassi rühmal.

ElGamal avalik võti kodeeritakse koos ElGamal krüptosüsteemi parameetritega ning konkreetset valimist iseloomustava identifikaatoriga. Krüptosüsteemi parameetrid on osaks algoritmi identifikaatori struktuurist, avalik võti on kodeeritud `SubjectPublicKeyInfo` struktuuri.

```
elGamalEncryption OBJECT IDENTIFIER ::= {
    {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1)␣
↪dds(3029) asymmetric-encryption(2) 1}
}

elGamal-Params-IVXV ::= SEQUENCE {
    p                   INTEGER,
    g                   INTEGER,
    election-identifier GeneralString
}

elGamalPublicKey ::= SEQUENCE {
    y           INTEGER,
}

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm   AlgorithmIdentifier,
    subjectPublicKey    BIT STRING
}
```

Valija tahteavalduse krüpteerimiseks võetakse UTF-8 kodeeringus struktuur `ballot` ning teisendatakse see ElGamal parameetrite poolt kirjeldatud rühma elemendiks. Eeldame, et parameeter `p` on 256 baiti. Sellisel juhul võib struktuuri `ballot` pikkus olla 253 baiti. Avakujul tahteavaldus pikendatakse parameetri `p` pikkuseni.

```
padded-ballot = ballot '\x00' '\x01' *'\xff' '\x00'
```

Pikendatud tahteavaldust interpreteeritakse kui täisarvu, mis kodeeritakse ruutjäägina parameetri `p` poolt kirjeldatud rühmas. Kodeerimine on üksühene ning oluline krüptogrammi edasise miksimise jaoks.

Tahteavaldus krüpteeritakse vastavalt ElGamal meetodile avaliku võtmega.

```
elGamalEncryptedMessage ::= SEQUENCE {
    a           INTEGER,
    b           INTEGER
}
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
encryptedBallot ::= SEQUENCE {
    algorithm   AlgorithmIdentifier,
    cipher  ANY
}
```

Andmestruktuuri `encryptedBallot` DER-kodeering on krüpteeritud sedel ehk sisemine ümbrik topeltümbriku skeemis.

Tahteavalduse krüpteerimise käigus genereeritakse valijarakenduses juhuarv, mida ElGamal krüpteerimisel kasutab. Sama juhuarv avalikustatakse hiljem kontrollrakendusele. Tulenevalt ElGamal krüptosüsteemi eripärast funktsioneerib see juhuarv nö. teise võtmena ning võimaldab krüptogrammi dekodeerimist kontrollrakenduses.

4.3 Valija poolt allkirjastatud hääl

Krüpteeritud sedel tuleb enne kogumisteenusesse talletamisele saatmist digitaalselt allkirjastada, milleks on võimalik kasutada kõiki Eesti Vabariigis kehtivaid digitaalallkirjavahendeid – ID-kaart, Digi-ID, Mobiil-ID.

Käesolev spetsifikatsioon näeb ette Eesti Vabariigi Standardikavandis [BDOC2.1] defineeritud BDOC allkirjavormingu kasutamise. BDOC allkirjavorming koosneb ETSI standardi TS 101 903 (XadES) profiilist ning OpenDocument konteineri vormingust. IVXV protokollistik võimaldab ka alternatiivsete allkirja- ning konteinervormingute kasutamist.

Olenevalt käimasoleval valimisel esitatud küsimuste arvust võib digitaalselt allkirjastatud hääl sisaldada ühte või mitut andmefaili MIME-tüübiga `application/octet-stream`. Iga andmefaili sisuks on krüpteeritud sedel. Andmefaili ja teiste signeeritavate andmeobjektide räsimiseks enne allkirjastamist kasutatakse räsifunktsiooni SHA-256. Andmefaili nimi moodustatakse laiendist ‚`ballot`‘ ning valimise ja küsimuse identifikaatorist. Kõik viidatud andmefailid peavad sisalduma allkirjakonteineris. Digitaalselt allkirjastatud hääl ei tohi sisaldada muid andmefaile kui neid, mis sisaldavad hääli mõne käimasoleva valimise kontekstis. Seadistusele mittevastavate häälte vastuvõtmisest, talletamisest ja töötlemisest peab kogumisteenus keelduma.

```
extension = "ballot"

encrypted-ballot-name = election-identifier '.' question-identifier
↪'.' extension
```

Valija poolt valijarakenduses allkirjastatud hääl moodustatakse nii, et on võimalik selle edasine kvalifitseerimine kogumisteenuses. Käesolev septsifikatsioon näeb ette hääle kvalifitseerimiseks nii OCSP kehtivuskinnituse kui PKIX ajatempli võtmise. Sellisena on lõplik, kvalfitseeritud hääl, BDOC-TS vormingus.

Kui hääl allkirjastatakse ID-kaardi või Digi-ID’ga, siis toimub algse allkirjastatud konteineri moodustamine valijarakenduses. Kui hääl allkirjastatakse Mobiil-ID’ga, siis toi-

mub konteineri moodustamine valijarakenduse ning kogumisteenuse poolt vahendatava Mobiil-ID teenuse koostöös. Mobiil-ID juhtumil kasutab kogumisteenus Mobiil-ID teenust ainult signatuuri saamiseks krüpteeritud sedelile. Kõik hääle kvalifitseerimiseks vajalikud elemendid hangitakse vastavatelt teenustelt alles siis kui valijarakendus on saatnud signeeritud hääle talletamiseks. Kvalifitseeritud hääl esitatakse kogumisteenuse poolt valijarakendusele verifitseerimiseks, ainult kvalifitseeritud hääl peab vastama BDOC 2.1 standardi tingimustele – valijarakenduse poolt moodustatud hääl on vaheetapp kvalifitseeritud hääleni jõudmiseks.

Valijarakenduses signeeritud häälel peab olema üks ja ainult üks allkiri, mida hoitakse signatuurifailis `META-INF/signature0.xml`. Häält ja allkirja sisaldav konteiner (edaspidi viidatud kui `SignedVote`) moodustatakse BDOC 2.1 standardis kirjeldatud meetodit kasutades.

Spetsifitseerime valijarakenduses allkirjastatud hääle vormingu ühe küsimuse korral.

Räsialgoritmina `DIGEST_ALG` on kasutusel SHA-256 (http://www.w3.org/2001/04/xmlenc#sha256). XML kanoniseerimiseks (`CANON_ALG`) kasutatakse meetodit c14n11 (http://www.w3.org/2006/12/xml-c14n11).

RSA võtmete korral (ID-kaart, Digi-ID) on allkirjastamismeetodiks http://www.w3.org/2001/04/xmldsig-more#rsa-sha256. ECC võtmete korral (ID-kaart, Mobiil-ID) http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256.

Identifikaatorite *VOTE_REF*, *SP_REF*, *SP_URI* ning *SV_URI* täpne väärtus ei ole oluline.

4.3.1 Element *SignedProperties*

Element `SignedProperties` moodustatakse kooskõlas BDOC 2.1 standardiga. Kui kvalifitseerimisel kasutatakse ajatemplit, siis elementi `SignaturePolicyIdentifier` ei kasutata. Ühtegi mitte-kohustuslikku elementi ei kasutata. Allkirjastamise kellaaja fikseerib andmestruktuuri täitev arvuti ning valija X509-sertifikaat saadakse kas ID-kaardilt või Mobiil-ID teenuse vahendusel.

```
1  <xades:SignedProperties xmlns:asic="http://uri.etsi.org/02918/v1.2.1
   ↪#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xades=
   ↪"http://uri.etsi.org/01903/v1.3.2#" Id="%SP_URI%">
2  <xades:SignedSignatureProperties>
3    <xades:SigningTime>%SIGNING_TIME%</xades:SigningTime>
4    <xades:SigningCertificate>
5      <xades:Cert>
6        <xades:CertDigest>
7          <ds:DigestMethod Algorithm="%DIGEST_ALG%"></ds:DigestMethod>
8          <ds:DigestValue>%CERT_DIGEST%</ds:DigestValue>
9        </xades:CertDigest>
10       <xades:IssuerSerial>
11         <ds:X509IssuerName>%ISSUER_NAME%</ds:X509IssuerName>
12         <ds:X509SerialNumber>%ISSUER_SERIAL%</ds:X509SerialNumber>
13       </xades:IssuerSerial>
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
    </xades:Cert>
  </xades:SigningCertificate>
</xades:SignedSignatureProperties>
<xades:SignedDataObjectProperties>
  <xades:DataObjectFormat ObjectReference="#%VOTE_REF%">
    <xades:MimeType>application/octet-stream</xades:MimeType>
  </xades:DataObjectFormat>
</xades:SignedDataObjectProperties>
</xades:SignedProperties>
```

4.3.2 Element *SignedInfo*

Element `SignedInfo` moodustatakse kooskõlas BDOC 2.1 standardiga, viidates nii krüpteeritud sedelile (`VOTE_DIGEST`) kui elemendile `SignedProperties` (`SP_DIGEST`).

```
<ds:SignedInfo xmlns:asic="http://uri.etsi.org/02918/v1.2.1#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xades="http://uri.etsi.org/01903/v1.3.2#">
  <ds:CanonicalizationMethod Algorithm="%CANON_ALG%"></ds:CanonicalizationMethod>
  <ds:SignatureMethod Algorithm="%SIG_ALG%"></ds:SignatureMethod>
  <ds:Reference Id="%VOTE_REF%" URI="%VOTE_URI%">
    <ds:DigestMethod Algorithm="%DIGEST_ALG%"></ds:DigestMethod>
    <ds:DigestValue>%VOTE_DIGEST%</ds:DigestValue>
  </ds:Reference>
  <ds:Reference Id="%SP_REF%" Type="http://uri.etsi.org/01903#SignedProperties" URI="#%SP_URI%">
    <ds:Transforms>
      <ds:Transform Algorithm="%CANON_ALG%"></ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="%DIGEST_ALG%"></ds:DigestMethod>
    <ds:DigestValue>%SP_DIGEST%</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
```

4.3.3 Element *SignatureValue*

Element `SignatureValue` moodustatakse kooskõlas BDOC 2.1 standardiga. Kanoniseeritud elemendist `SignedInfo` arvutatakse räsi, mis allkirjastatakse PKCS1-meetodiga.

```
<ds:SignatureValue xmlns:asic="http://uri.etsi.org/02918/v1.2.1#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Id="%SV_URI%">%SIG_VALUE%</ds:SignatureValue>
```

4.3.4 Element *XAdESSignatures*

Element `XAdESSignatures` sisaldab ühte `Signature` elementi, mis on koostatud lähtudes kõigist eelmistest elementidest ning valija X509 sertifikaadist. Elementi `UnsignedProperties` ei kasutata.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<asic:XAdESSignatures xmlns:asic="http://uri.etsi.org/02918/v1.2.1#">
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="S0">
        %SI_XML%
        %SV_XML%
    <ds:KeyInfo>
      <ds:X509Data>
          <ds:X509Certificate>%X509_CERT%</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
    <ds:Object>
        <xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Target="#S0">
            %SP_XML%
      </xades:QualifyingProperties>
    </ds:Object>
  </ds:Signature>
</asic:XAdESSignatures>
```

PEATÜKK 5

Elektroonilise hääle kvalifitseerimine talletamiseks

5.1 Kvalifitseeritud hääl

Valijarakenduse töö tulemusena saadetakse kogumisteenusesse talletamiseks topeltümbrik, mis sisaldab endas valija tahteavaldust krüpteeritud kujul, valija allkirja krüpteeritud tahteavaldusel kooskõlastatud allkirja- ja konteinervormingus ning valija allkirjastamissertifikaati X509-vormingus.

Hääle edukaks talletamiseks näeb IVXV protokoll ette hääle registreerimise välise registreerimisteenuse osutaja juures ning registreerimistõendi valijarakendusele kättesaadavaks tegemise. Valimise korraldaja võib hääle kvalifitseerimiseks näha ette täiendavaid samme lisaks registreerimisele – näiteks kehtivuskinnituse hankimist hääle allkirjastanud sertifikaadi kohta.

Kõik kogumisteenuse poolt hangitavad kvalifitseerivad elemendid, mis määravad hääle staatuse hilisemates töötlusetappides, tuleb esitada valijarakendusele ning nõudmise korral ka kontrollrakendusele tagamaks, et valija saab oma hääle korrektse menetlemise võimalikkusest õigeaegselt teada.

5.1.1 OCSP kehtivuskinnitus

OCSP (*Online Certificate Status Protocol*) on standartne protokoll X509-sertifikaatide kehtivusinfo pärimiseks. Kogumisteenus võib seda protokolli kasutada hääle allkirjastanud sertifikaadi kehtivuse teadasaamiseks. OCSP vastus ütleb, et sertifikaat kehtis päringu tegemise ajahetkel, kuid ei seosta OCSP vastust konkreetse allkirjaga.

5.1.2 OCSP-TM kehtivuskinnitus

BDOC 2.1 standard kirjeldab BDOC-TM profiili, kus OCSP protokolliga hangitud kehtivuskinnitus toimib ka ajamärgendina, mis kinnitab, et konkreetne allkiri eksisteeris enne OCSP kehtivuskinnituse võtmist.

5.1.3 RFC3161 ajatempel

RFC3161 ajatempli protokolliga saadakse usaldusteenuse pakkujalt kinnitus, et mingi andmekogum eksisteeris enne teatud ajahetke. BDOC-TS kontekstis ajatembeldatakse allkirja element `SignatureValue` kanoniseeritud kujul. Klassikaline OCSP vastus koos RFC 3161 vormingus ajatempliga kvalifitseerivad BDOC-TS allkirja.

5.2 Talletamine

Elektroonilise hääle talletamine kogumisteenuses tähendab:

1. hääle vastuvõtmist valijarakenduselt ning hääletaja allkirja verifitseerimist;
2. hääle võimalikku kvalifitseerimist – näiteks sertifikaadi kehtivuse tõendamist hääle allkirjastamisele lähedasel ajahetkel;
3. hääle registreerimist sõltumatus registreerimisteenuses;
4. häält kvalifitseerivate elementide vahendamist valijarakendusele.

Erinevad kombinatsioonid allkirjavormingust ning hääli kvalifitseerivatest teenustest võivad tekitada erinevaid IVXV-profiile. Konkreetse dokumendi raames on IVXV profiil:

1. Allkirjastatud hääle vorming on BDOC-TS;
2. Kehtivuskinnitusprotokolliks on standartne OCSP;
3. BDOC-TS kvalifitseerimiseks kasutatav RFC3161 ajatempel on kasutusel ka registreerimistõendina.

PEATÜKK 6

Elektroonilise hääle registreerimine

Hääle edukaks talletamiseks näeb IVXV protokoll ette hääle registreerimise välise registreerimisteenuse osutaja juures ning registreerimistõendi valijarakendusele kättesaadavaks tegemise. Registreerimisteenus toimub RFC3161 ajatempli protokolli baasil. Protokolli on laiendatud selliselt, et kogumisteenus saab ajatempli päringule anda oma signatuuri, mis teeb võimalikuks hilisema võrdleva väljavõtte registreerimisteenusest. Sõltumatu registreerimisteenuse olemasolu vähendab häälte kogumisteenuse poolt „kaotamise“ riski.

6.1 Registreerimisteenus

Registreerimisteenus on teenus, mille abil Kogumisteenus registreerib kõik Hääletajatelt saadud hääled. Pärast hääletamisperioodi lõppu edastab Kogumisteenus talletatud hääled Töötlejale ning Registreerimisteenus edastab registreeritud hääled Töötlejale.

Registreerimisteenus aitab tagada e-valimiskasti terviklust. Eeldame et Kogumisteenusel puudub võimalus võltsida digitaalallkirja ning sellisel moel tekitada juurde hääli või muuta juba talletatud hääli. Riskina säilib võimalus, et Kogumisteenus ei anna kõiki talletatud hääli Töötlejale üle. Häälte valikulise üleandmise riski maandamiseks kasutab IVXV protokoll täiendavat Registreerimisteenust, kuhu Kogumisteenus iga talletatud hääle registreerib. Hääletajal on protokollikohaselt võimalus korrektses registreerimises veenduda – Registreerimisteenuse digitaalselt allkirjastatud kinnitus konkreetse hääle registreerimise kohta esitatakse ka konkreetsele hääletajale.

6.1.1 Registreerimisteenus protokollis

Kogumisteenus saadab Registreerimisteenusele enda poolt allkirjastatud registreerimiskorralduse (edaspidi KORRALDUS), mis sisaldab hääle identifikaatorit ja allkirjastatud hääle räsi:

```
KORRALDUS = Sign_K(V_id, Hash(VOTE))
```

Registreerimisteenus talletab KORRALDUSE hilisemaks väljastamiseks ning vastab Kogumisteenusele oma poolt allkirjastatud registreerimiskinnitusega (edaspidi KINNITUS), mis allkirjastab algset KORRALDUST ning KINNITUSE väljastamise aega:

```
KINNITUS = Sign_R(Hash(KORRALDUS), t)
```

Kogumisteenus tagastab nii KINNITUSE kui KORRALDUSE valijarakendusele, Hääletaja saab teate hääle positiivsest talletamisest ainult KINNITUSE ja selle aluseks olnud KORRALDUSE edukal verifitseerimisel.

Hääletamisperioodi lõppedes edastab Registreerimisteenus Kogumisteenuse poolt tehtud KORRALDUSED Töötlejale.

Registreerimisteenus peab Töötlejale algselt andma üle vähemalt loendi:

```
(V_id, Hash(VOTE))
```

Kui Registreerimisteenus annab üle ainult ülalkirjeldatud loendi, siis peab ta hiljem oleme võimeline andma loendi nõutud elemendile vastava korralduse:

```
(V_id, Hash(VOTE)), Sign_K(V_id, Hash(VOTE))
```

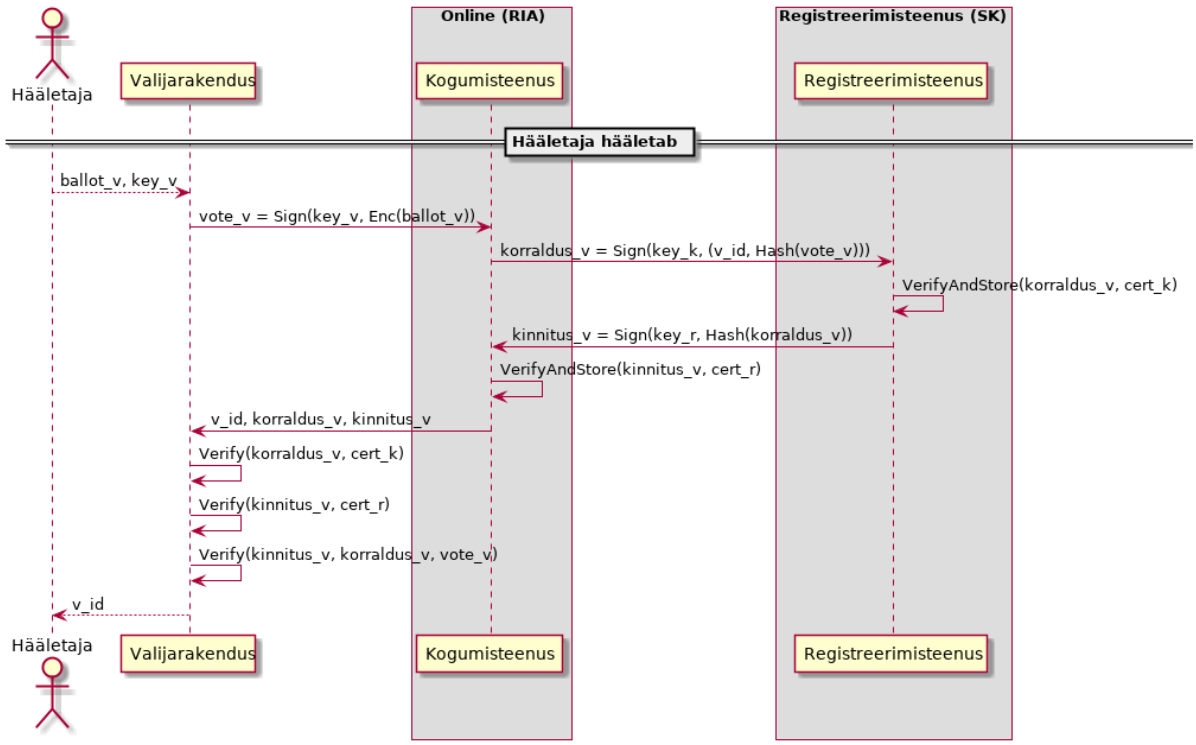
6.1.2 Registreerimisteenuse liidesed

Registreerimisteenusel on kaks liidest

1. KORRALDUSTE vastuvõtmise ja KINNITUSTE väljastamise liides;
2. KORRALDUSE alusel väljastatud KINNITUSTE loetelu ja nende aluseks olnud KORRALDUSTE väljastamise liides.

Registreerimisteenuse funktsionaalsuseks on Kogumisteenusele KINNITUSTE väljastamine, väljastatud KINNITUSTE ja neile aluseks olevate KORRALDUSTE säilitamine ja hilisem Töötlejale üleandmine.

Kui Registreerimisteenus osutab teenust mitmele erinevale osapoolele, siis peab olema garanteeritud, et konkreetse valimisega seotud Kogumisteenuse KORRALDUSED ja neile vastavad KINNITUSED on Valijarakenduse ja Kontrollrakenduse poolt kontrollitavalt eristatavad teiste osapoolte KORRALDUSTEST ning neile vastavatest KINNITUSTEST. Vastasel juhul võib tekkida olukord, kus Kogumisteenus küll registreerib hääle, kuid info sellest ei jõua Töötlejani.



Joonis 6.1. Registreerimisteenuse roll hääletamisel

Registreerimisteenus peab olema võimeline kõiki Kogumisteenuse poolt tulnud KORRALDUSI üle andma.

6.1.3 Osapoolte nõuded registreerimisteenusele

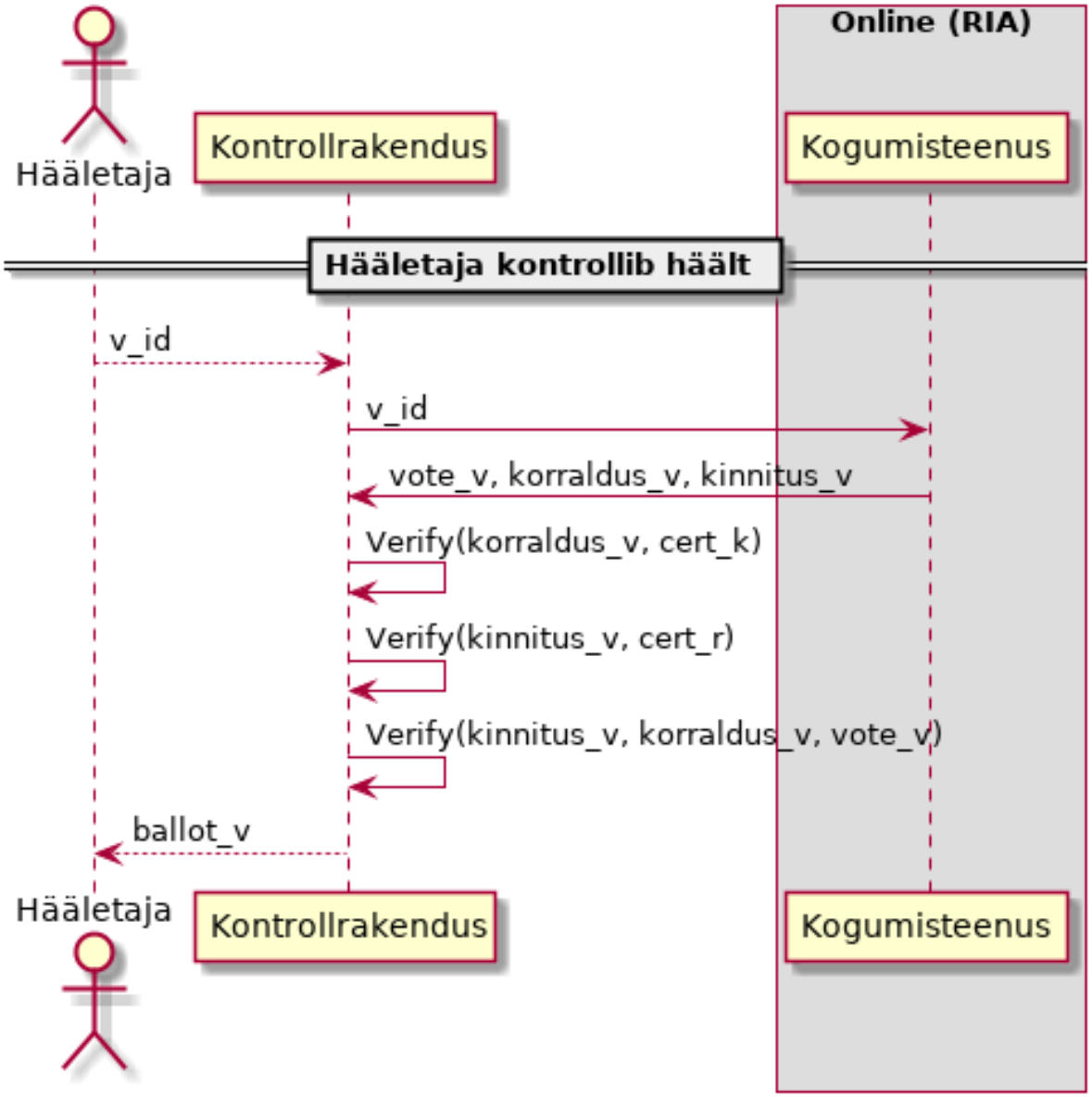
6.1.3.1 Töötleja

Töötleja ülesanne on muuhulgas tuvastada,

1. millised Kogumisteenuse poolt üle antud hääled lähevad lugemisele ja
2. kas Kogumisteenus on jätnud hääli üle andmata.

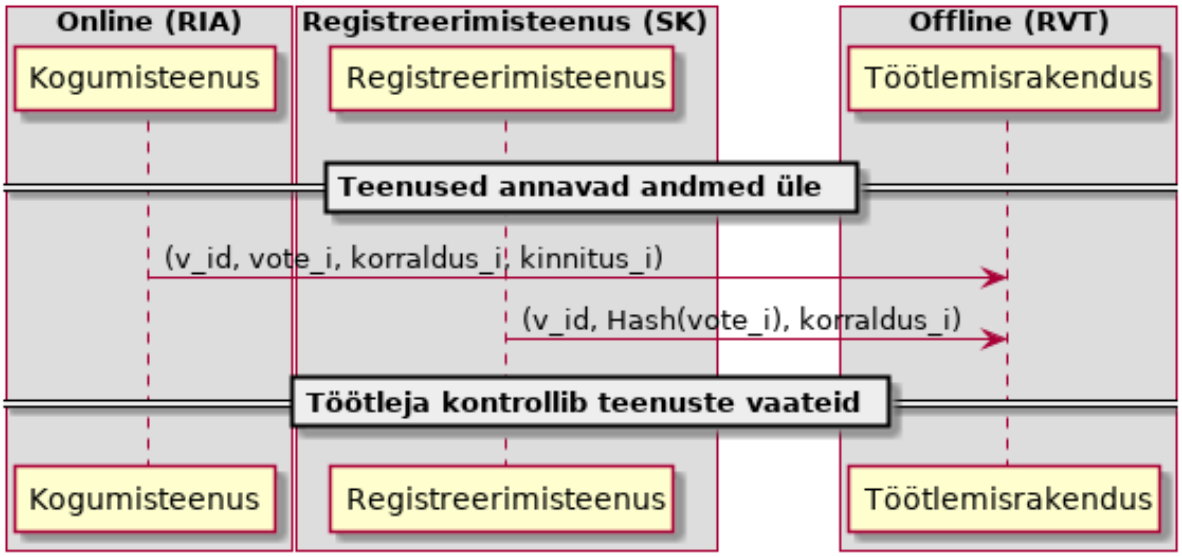
Töötleja töö tulemusena selguvaid erisusi tuleb lahendada ning siin on järgmised 3 juhtumit:

1. Kogumisteenus annab Töötlejale üle allkirjastatud hääle koos KINNITUSega, Registreerimisteenus annab Töötlejale üle Kogumisteenuse KORRALDUSE, mille alusel KINNITUS anti - vaidlust ei ole, kui konkreetne hääl oli antud Hääletaja jaoks viimane, siis suunatakse ta lugemisele.
2. Kogumisteenus annab Töötlejale üle allkirjastatud hääle koos KINNITUSega, Registreerimisteenus ei anna Töötlejale selle hääle kohta midagi üle. Kuna KINNITUS on Registreerimisteenuse poolt allkirjastatud, siis on viga Registreerimisteenuse poolel. Kui konkreetne hääl oli antud Hääletaja jaoks viimane, siis suunatakse ta lugemisele.



Joonis 6.2. Registreerimisteenuse roll hääle kontrollimisel

3. Registreerimisteenus annab Töötlejale üle Kogumisteenuse KORRALDUSE, Kogumisteenus ei anna Töötlejale vastava räsiga häält üle. Kuna KORRALDUS on Kogumisteenuse poolt allkirjastatud, siis on viga Kogumisteenuse poolel ning antud hääl tuleb üles otsida.



Joonis 6.3. Registreerimisteenuse roll häälte üleandmisel

6.1.3.2 Hääletaja

Hääletaja jaoks on oht, et Kogumisteenus võib tema hääle ‚unustada'. Nõuetekohase KINNITUSE nägemine annab Hääletajale kindluse, et väline osapool garanteerib tema hääle Töötlejani jõudmist. Kindluse jaoks on oluline:

1. KINNITUS on Registreerimisteenuse poolt allkirjastatud;
2. KINNITUSES sisalduv algne KORRALDUS on Kogumisteenuse poolt allkirjastatud;
3. usaldus, et Registreerimisteenus on võimeline KINNITUSE andmise fakti meeles pidama;
4. usaldus, et Registreerimisteenus on võimeline KINNITUSE andmise kohasust Töötlejale tõestama;
5. usaldus, et Kogumisteenusel ei ole võimalik hankida alternatiivset valekinnitust, mis Valijarakenduses verifitseerub, kuid Töötlejani ei jõua.

6.1.3.3 Kogumisteenus

Kogumisteenuse jaoks on oht et, Registreerimisteenuse poolt üleantud KINNITUSTE ja talletatud häälte vaated erinevad. Kogumisteenuse poolt KORRALDUSELE antud allkiri on Kogumisteenuse garantii, et Registreerimisteenuse poolt ei saa tekkida fiktiivseid KINNITUSI, mida Kogumisteenus tegelikult nõudnud pole.

Kogumisteenus talletab kõiki Registreerimisteenuse vastuseid. Kuna need on allkirjastatud, siis on täiendav info oluline vaid siis kui Kogumisteenus väidab, et mingit KORRALDUST ei ole antud, kuigi Registreerimisteenus on (v_id, Hash(VOTE)) esitanud. Sellisel juhul saab Registreerimisteenus esitada terve Kogumisteenuse KORRALDUSE (või vähemalt selle allkirjastatud komponendi)

6.1.3.4 Registreerimisteenus

Registreerimisteenus on huvitatud, et vaidlusolukordades, kus Kogumisteenus jätab midagi üle andmata, oleks tal võimalik oma tegevuse korrektsust tõestada. Oluline on tagada:

1. Kogumisteenuse poolt konkreetse valimise raames antavad KORRALDUSED on teiste klientide poolt esitatud KORRALDUSTEST kontrollitavalt eristatavad.
2. Kogumisteenus ei saa juba esitatud KORRALDUSTE kohta väita, et ta neid ei esitanud.

6.1.4 Registreerimisteenuse realiseerimine RFC 3161 protokolli raamistikus

PKIX on ajatembeldusprotokoll, kus usaldatav kolmas osapool (ajatempliteenuse osutaja ehk ATO) kinnitab oma allkirjaga andmete eksisteerimist konkreetsel ajahetkel. Protokoll koosneb ühest päringust ja vastusest.

Ajatemplipäring:

```
TimeStampReq ::= SEQUENCE  {
  version                  INTEGER  { v1(1) },
  messageImprint           MessageImprint,
    --a hash algorithm OID and the hash value of the data to be
    --time-stamped
  reqPolicy                TSAPolicyId              OPTIONAL,
  nonce                    INTEGER                  OPTIONAL,
  certReq                  BOOLEAN                  DEFAULT FALSE,
  extensions               [0] IMPLICIT Extensions  OPTIONAL  }
```

Ajatembeldatavad andmed esitatakse teenusele messageImprint koosseisus räsina. TimeStampReq ei sisalda endas päringu esitaja allkirja.

ATO vastus ajatemplipäringule:

```
TimeStampResp ::= SEQUENCE  {
  status                   PKIStatusInfo,
  timeStampToken           TimeStampToken           OPTIONAL  }

TimeStampToken ::= ContentInfo
  -- contentType is id-signedData ([CMS])
  -- content is SignedData ([CMS])

TSTInfo ::= SEQUENCE  {
  version                  INTEGER  { v1(1) },
  policy                   TSAPolicyId,
  messageImprint           MessageImprint,
    -- MUST have the same value as the similar field in
    -- TimeStampReq
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
  serialNumber          INTEGER,
    -- Time-Stamping users MUST be ready to accommodate integers
    -- up to 160 bits.
  genTime               GeneralizedTime,
  accuracy              Accuracy                OPTIONAL,
  ordering              BOOLEAN                 DEFAULT FALSE,
  nonce                 INTEGER                 OPTIONAL,
    -- MUST be present if the similar field was present
    -- in TimeStampReq.  In that case it MUST have the same value.
  tsa                   [0] GeneralName         OPTIONAL,
  extensions            [1] IMPLICIT Extensions  OPTIONAL }
```

TimeStampResp on ATO poolt digitaalselt allkirjastatud konteiner, mis sisaldab endas päringu koosseisus saadud messageImprint'i ning nonssi.

Registreerimisteenuse huvides on, et Kogumisteenuse päring oleks signeeritud. Kuna RFC 3161 ei toeta allkirjastatud päringuid on alternatiiviks kasutada mõnda laiendust, mis võimaldab Kogumisteenuse signatuuri edastamist. See laiendus tuleks teenuse poolt ajatempli koosseisus ka tagasi saata. Kuna RFC 3161 ei sõnasta laienduste tagasipeegeldamise nõuet ühemõtteliselt on reaalne võimalus kaustada protokolli laiendamiseks ajatemplipäringu nonssi. Nonss on ASN.1 INTEGER andmetüüp kuhu saab kodeerida suvalise struktuuriga andmeid, mis teeb võimalikuks järgmise skeemi:

Enne hääletamist:

1. Kogumisteenus genereerib allkirjastamisvõtme ja sertifikaadi.
2. Kogumisteenus annab sertifikaadi Korraldajale üle.
3. Kogumisteenus seadistab ennast ATO'd kasutama.

Hääletamise ajal:

1. Valija saadab hääle talletamiseks.
2. Kogumisteenus räsib hääle, allkirjastab räsi ning võtab räsile ajatempli, kasutades ajatemplipäringu TimeStampReq nonssina oma allkirja sellel räsil.
3. ATO töötleb ajatemplipäringut kooskõlas RFC 3161 nõuetega ning väljastab allkirjastatud ajatempli.
4. Kogumisteenus vahendab ajatempli Valijarakendusele, mis teostab järgmised kontrollid:
   a) ajatempel on ATO poolt allkirjastatud,
   b) ajatempel sisaldab nonssi,
   c) ajatemple sisaldab tema hääle räsi,
   d) nonss on Kogumisteenuse poolt allkirjastatud valija hääle räsi.

Peale hääletamist:

1. Korraldaja annab ATO'le ajavahemiku ja Kogumisteenuse sertifikaadi
2. ATO otsib kõigi selle ajavahemiku ajatemplipäringute ja vastuste hulgast neid, millel

a) on nonss,
b) nonss dekodeerub kokkuleppeliseks andmestruktuuriks,
c) andmestruktuur verifitseerub Kogumisteenuse sertifikaadiga.

3. ATO annab üle kõik leitud ajatemplipäringud ja ajatemplid.
4. Kogumisteenus annab üle kõik ajatemplipäringud, ajatemplid ja hääled.
5. Töötleja analüüsib andmeid vastavalt protokollile
6. KINNITUS on Registreerimisteenuse poolt allkirjastatud;
7. KINNITUSES sisalduv algne KORRALDUS on Kogumisteenuse poolt allkirjastatud;
8. Usaldus, et Registreerimisteenus on võimeline KINNITUSE andmise fakti meeles pidama;
9. Usaldus, et Kogumisteenusel ei ole võimalik hankida alternatiivset valekinnitust, mis Valijarakenduses verifitseerub, kuid Töötlejani ei jõua
10. Usaldus, et Registreerimisteenus on võimeline KINNITUSE andmise kohasust Töötlejale tõestama;
11. Registreerimisteenuse poolt ei saa tekkida fiktiivseid KINNITUSI, mida Kogumisteenus tegelikult nõudnud pole
12. Kogumisteenus ei saa juba esitatud KORRALDUSTE kohta väita, et ta neid ei esitanud

Nonssi vorming:

```
Signature ::= SEQUENCE {
  signingAlgorithm AlgorithmIdentifier,
  signature        ANY DEFINED BY signingAlgorithm
}

AlgorithmIdentifier ::= SEQUENCE {
  algorithm  OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm OPTIONAL
}
```

Sõnumiks on TimeStampReq.messageImprint DER-kodeering:

```
MessageImprint ::= SEQUENCE {
  hashAlgorithm AlgorithmIdentifier,
  hashedMessage OCTET STRING
}
```

RSA kasutamisel allkirjastamiseks. Signature.signingAlgorithm.algorithm sõltub sõnumi hashAlgorithmist:

```
pkcs-1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) US(840)␣
↪rsadsi(113549) pkcs(1) 1 }

sha-1WithRSAEncryption   OBJECT IDENTIFIER  ::=  { pkcs-1  5 }
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
sha224WithRSAEncryption  OBJECT IDENTIFIER  ::=  { pkcs-1 14 }
sha256WithRSAEncryption  OBJECT IDENTIFIER  ::=  { pkcs-1 11 }
sha384WithRSAEncryption  OBJECT IDENTIFIER  ::=  { pkcs-1 12 }
sha512WithRSAEncryption  OBJECT IDENTIFIER  ::=  { pkcs-1 13 }
```

Signature.signingAlgorithm.parameters puudub või on NULL.

Signature.signature on OCTET STRING, mis sisaldab RSA signatuuri sõnumil.

6.1.5 ATO väljavõte

ATO väljavõte Kogumisteenuse poolt esitatud TimeStampReq vormingus päringutest esitatakse ZIP failina, kus iga päring on salvestatud ühte faili, mis vastab järgmistele tingimustele:

- Kaustastruktuur puudub, ükski fail ei paikne kaustas.
- Failinimed on unikaalsed (failinimedele samas tähendust ei omistata).

Üleantav andmekomplekt on:

- Andmefail: *<andmefailinimi>.zip*
- Kontrollsummafail: *<andmefailinimi>.zip.sha256sum.asice*

Ühe realise kontrollsummafaili sisu on HEX kodeeringus SHA256 räsi andmefailist.

PEATÜKK 7

Elektroonilise hääle kontrollimine

Elektroonilist häält kontrollitakse töötlemisrakenduses, kogumisteenuses, valijarakenduses ja kontrollrakenduses. Kõige põhjalikuma kontrolli läbib elektrooniline hääl e-valimiskasti koosseisus töötlemisrakenduses, kus otsustatakse konkreetse hääle lugemisele saatmine või mittesaatmine. Iga üksiku hääle kohta läbitakse töötlemisrakendusega analoogsel tasemel kontroll valijarakenduses, kus veendutakse, et kogumisteenus on hääle kvalifitseerinud selliselt, et töötlemisrakenduses tehtavad kontrollid õnnestuvad. Valijarakendusega analoogsed kontrollid viib läbi kontrollrakendus.

7.1 Kontrollid kogumisteenuses

Valijarakendus saadab kogumisteenusele allkirjastatud hääle koosseisus:

1. krüpteeritud sedeli;
2. valija allkirja krüpteeritud sedelil;
3. valija allkirjastamissertifikaadi.

Kogumisteenus viib läbi minimaalselt järgmised kontrollid:

1. hääle allkirjastaja on valijate nimekirjas;
2. allkirjastatud hääl on esitatud korrektses konteinervormingus;
3. digitaalallkiri krüpteeritud sedelil on korrektne;
4. hääle allkirjastaja sertifikaat on kehtiv hääle vastuvõtmise ajahetkel.

Hääle allkirjastaja sertifikaadi kehtivuse kontrolliks teeb kogumisteenus päringu kehtivuskinnitusteenusele. Kogumisteenus verifitseerib kehtivuskinnitusteenuse vastust sertifikaadi oleku kohta ning lisab selle vastuse häält kvalifitseerivate elementide hulka.

Kogumisteenus registreerib hääle talletamise fakti välises registreerimisteenuses, allkirjastades registreerimispäringu ning talletades registreerimisteenuse poolt allkirjastatud registreerimistõendi häält kvalifitseerivate elementide hulka.

Kogumisteenus tagastab kõik tema poolt hangitud häält kvalifitseerivad elemendid valijarakendusele koos hääle unikaalse identifikaatoriga.

7.2 Kontrollid valijarakenduses

Valijarakendus moodustab valija avakujul tahteavalduse põhjal krüpteeritud sedeli ning allkirjastab selle valija allkirja andmise vahendiga.

Valijarakenduse rolliks peale hääle allkirjastamist on veenduda, et kogumisteenus käitus häält kvaliftseerivate elementide võtmisel protokollikohaselt ning et hääl on talletatud selliselt, et ta saab töötlemisrakenduse poolt arvesse võetud.

Valijarakendus viib läbi minimaalselt järgmised kontrollid:

1. Kogumisteenus võttis kehtivuskinnituse valija sertifikaadile volitatud kehtivuskinnitusteenuselt. Valijarakendus kontrollib allkirja kehtivuskinnitusteenuse vastusel.
2. Kogumisteenus registreeris valija poolt allkirjastatud hääle volitatud registreerimisteenuses. Valijarakendus kontrollib, et kogumisteenuse poolt moodustatud päring oli kogumisteenuse poolt signeeritud ning viitas korrektselt allkirjastatud häälele. Valijarakendus kontrollib, et registreerimisteenuse vastus on allkirjastatud õige registreerimisteenuse osutaja poolt ning sisaldab kogumisteenuse poolt allkirjastatud päringut.

Kui hääle kvalifitseerimiseks vajalike elementide kontroll ei õnnestu, siis teavitab valijarakendus sellest kasutajat.

7.3 Kontrollid kontrollrakenduses

Kontrollrakendus saab valijarakendusest järgmise info:

1. Krüpteeritud sedeli moodustamisel kasutatud juhuslikkuse;
2. Allkirjastatud hääle unikaalse identifikaatori kogumisteenuses.

Kontrollrakendus kasutab hääle unikaalset identifikaatorit kogumisteenusest järgmise info saamiseks:

1. krüpteeritud sedel;
2. valija allkiri krüpteeritud sedelil;
3. valija allkirjastamissertifikaat;
4. häält kvalifitseerivad elemendid, kaasa arvatud kehtivuskinnitus ja registreerimistõend.

Kontrollrakendus teostab järgmised kontrollid:

1. allkirjastatud hääl on esitatud korrektses konteinervormingus;
2. digitaalallkiri krüpteeritud sedelil on korrektne;
3. hääle allkirjastaja sertifikaat on kehtiv hääle vastuvõtmise ajahetkel, mida kinnitab korrektne kehtivuskinnitus;
4. hääl on korrektselt registreeritud õiges registreerimisteenuses.

Nende kontrollide teostamise järel kuvab kontrollrakendus hääle allkirjastanud isiku andmeid.

Täiendavalt kasutab kontrollrakendus krüpteeritud sedeli moodustamisel kasutatud juhuslikkust krüpteeritud sedeli dekrüpteerimiseks.

> **Tähelepanu:** Ühe hääle krüpteerimisel kasutatud juhuslikkust saab kasutada ainult selle hääle dekrüpteerimiseks. Mitme erineva hääle dekrüpteerimiseks läheb vaja häälte salastamise võtme privaatkomponenti.

Kontrollrakendus veendub, et dekrüpteerimisel saadud avatekst vastab avakujul tahteavalduse vorminõuetele.

Kontrollrakendus kuvab vorminõuetele vastava tahteavalduse võimaldamaks kontrollijal veenduda selle tahteavalduse korrektsuses.

7.4 Kontrollid töötlemisrakenduses

Töötlemisrakendus kontrollib iga üksikut häält eraldi, veendudes muuhulgas, et iga kogumisteenuse ning registreerimisteenuse poolt esitatud vaated e-valimiskasti sisu kohta on konsistentsed. Seejärel otsustab töötlemisrakendus iga valija häälte kohta, milline neist on ajaliselt viimane ning suunatakse töötlemise järgmisesse etappi, mille tulemusena hääl võib jõuda lugemisele.

Töötlemisrakenduse sisendiks on:

1. Loend registreerimisteenuse poolt vastuvõetud registreerimispäringutest;
2. Loend kogumisteenuses rakendatud valijanimekirjadest;
3. Kogumisteenuse poolt üle antud e-valimiskast, mis sisaldab iga hääle kohta krüpteeritud sedelit, valija allkirja krüpteeritud sedelil, valija allkirjastamissertifikaati, sertifikaadi kehtivuskinnitust ning registreerimistõendit.

Töötlemisrakendus kontrollib registreerimisteenuse ja kogumisteenuse kooskõla ning väljastab erinevused:

1. Hääled, mille kohta on olemas registreerimispäring kogumisteenuses, kuid vastus ei ole jõudnud kogumisteenusesse;
2. Hääled, mille kohta on olemas registreerimispäring registreerimisteenuses, kuid mida kogumisteenus ei ole üle andnud.

Töötlemisrakendus kontrollib iga üksikut häält:

1. hääle allkirjastaja oli valijate nimekirjas;
2. allkirjastatud hääl on esitatud korrektses konteinervormingus;
3. digitaalallkiri krüpteeritud sedelil on korrektne;
4. hääle allkirjastaja sertifikaat on kehtiv hääle vastuvõtmise ajahetkel, mida kinnitab korrektne kehtivuskinnitus;
5. hääl on korrektselt registreeritud õiges registreerimisteenuses.

Töötlemisrakendus otsustab, milline valija häältest oli viimane ning liigub töötlemise järgmisesse etappi. S.t. üks häält kvalifitseerivatest elementidest täidab hääle talletamise aja fikseerimise rolli ning selle elemendi põhjal moodustatakse üksikute häälte ajaline järgnevus. Olenevalt IVXV profiilist võib see element olla kehtivuskinnituse koosseisus (BDOC-TM), eraldi ajatemplina (BDOC-TS) või registreerimistõendi koosseisus (BDOC-TS).

PEATÜKK 8

Suhtlusprotokollid

8.1 Liides

Kogumisteenuse valijale suunatud mikroteenused suhtlevad valijarakendusega ja kontrollrakendusega JSON-RPC protokolli vahendusel.

id JSON-RPC päringuidentifikaator
method RPC-meetod
params Konkreetse RPC-meetodi parameetrid

```
{
    "id": 0.0,
    "method": "RPC.Method",
    "params": [
        {
            "MethodParam": "value",
            "SessionID": "ec3a0cab353d552952289f2c7ad52e27"
        }
    ]
}
```

error Võimalik veainfo või `null` vea puudumisel
id JSON-RPC päringuidentifikaator, peab ühtima päringus kasutatud id-ga
result Meetodipõhine vastusandmestruktuur

```
{
    "error": null,
    "id": 0.0,
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
    "result": {
        "ResultParam": "value",
        "SessionID": "ec3a0cab353d552952289f2c7ad52e27"
    }
}
```

Esimese päringuvahetuse käigus mõne IVXV mikroteenusega väljastatakse suhtlevale rakendusele HEX-kodeeritud unikaalne seansiidentifkaator (`result.SessionID`), mida rakendus kasutab edaspidi kõigis kogumisteenuse suunalistes päringutes (`params.SessionID`). Seansiidentifikaatori abil seostatakse hääletamisega seotud RPC-päringud üheks seansiks. Seostamine on informatiivne ning selle eesmärk on logianalüüsi lihtsustamine, hääle ringkonnakuuluvust jm. sisulisi aspekte puudutavad otsused tehakse digiallkirjastatud andmete põhjal.

Transpordiprotokollina on kasutusel TLS. Krüpteeritud kanali termineerimine toimub konkreetses mikroteenuses. Võimaldamaks koormuse jaotamist ning mikroteenuste paindlikku evitamist kasutatakse TLS-i SNI laiendust, mis lubab vahendusteenusel TLS voogu termineerimata õigesse mikroteenusinstantsi suunata. Vahendusteenus on tüüpiliselt kättesaadav kogumisteenuse välise liidese pordis 443.

8.2 Valikute nimekirja hankimine

Valikute nimekirja hankimine tähendab valijarakenduse suhtlemist nimekirjateenusega (SNI `choices.ivxv.invalid`). Valikute nimekirja hankimine eeldab valija autentimist ning tema ringkonnakuuluvuse tuvastamist.

Valijarakendus teeb päringu `RPC.VoterChoices` nimekirjade hankimiseks.

params.AuthMethod Toetatud valikud on meetodid `tls` ja `ticket`.

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

Päring `RPC.VoterChoices` ID-kaardiga autentimise korral - autentimine toimub TLS-protokolli tasemel päringu töötlemise ajal kasutades ID-kaardi autentimissertifikaati.

```
{
    "id": 0.0,
    "method": "RPC.VoterChoices",
    "params": [
        {
            "AuthMethod": "tls",
            "OS": "Operating System,2,0"
        }
    ]
}
```

Päring `RPC.VoterChoices` Mobiil-ID'ga autentimise korral - päringu sooritamiseks tuleb eelnevalt kasutada Mobiil-ID vahendusteenuse (SNI `mid.ivxv.invalid`) abi allkirjastatud autentimistõendi saamiseks.

params.AuthToken Autentimisteenuse vahendusel allkirjastatud tõend, mis sisaldab endas valija unikaalset identifikaatorit.

params.SessionID Kuna Mobiil-ID korral on nimekirja hankimisele eelnenud interaktsioon autentimistõendi saamiseks, on olemas seansiidentifikaator, mida tuleb kasutada.

```
{
    "id": 0.0,
    "method": "RPC.VoterChoices",
    "params": [
        {
            "AuthMethod": "ticket",
            "AuthToken":
↪"G1RTZqBSBKrzqReuKYrmFUFXWFPvaxhJjdiZi6zqAnaK3OvrT...",
            "OS": "Operating System,2,0",
            "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
        }
    ]
}
```

Nimekirjateenuse vastus päringule `RPC.VoterChoices`.

result.Choices Valija ringkonnakuuluvuse identifikaator `VoterDistrict`

result.List BASE64-kodeeritud ringkonna valikute nimekiri `DistrictChoices`

result.Voted Kui valija on juba hääletanud, siis `true`, vastasel juhul seda välja vastuses ei ole.

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "Choices": "0140.1",
        "List":
↪"ew0KICAgICAgICAgICAgIkVyYWtvbmQgMSI6IHsNCiAgICAgICAgICAgIC...",
        "SessionID": "ec3a0cab353d552952289f2c7ad52e27",
        "Voted": true
    }
}
```

Võimalikud veateated päringu `RPC.VoterChoices` korral.

BAD_CERTIFICATE Viga valija isikutuvastussertifikaadiga.

BAD_REQUEST Vigane päring.

INELIGIBLE_VOTER Valijal pole õigust hääletada.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

UNAUTHENTICATED Autentimata päring.

VOTER_TOO_YOUNG Valija on liiga noor.

VOTING_END Hääletusperiood on lõppenud.

8.3 Allkirjastatud hääle saatmine talletamiseks

Allkirjastatud hääle saatmine talletamiseks tähendab valijarakenduse suhtlemist hääletamisteenusega (SNI `voting.ivxv.invalid`).

Valijarakendus teeb päringu `RPC.Vote` allkirjastatud hääle talletamiseks saatmiseks.

params.AuthMethod Toetatud valikud on meetodid `tls` ja `ticket`.

params.Choices Valija ringkonnakuuluvuse identifikaator `VoterDistrict` mis kehtis valikute nimekirja hankimise ajal. Parameetri korrektne kasutamine võimaldab kogumisteenusel valijat hoiatada kui tema ringkonnakuuluvus on võrreldes hääletamise algushetkega muutunud.

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

params.Type Allkirjastatud hääle vorming. Hetkel on ainus toetatud väärtus `bdoc`.

params.Vote BASE64-kodeeritud hääl `SignedVote` eelpoolmääratud vormingus (*Valija poolt allkirjastatud hääl*).

Päring `RPC.Vote` ID-kaardiga autentimise korral.

```
{
    "id": 0.0,
    "method": "RPC.Vote",
    "params": [
        {
            "AuthMethod": "tls",
            "Choices": "0140.1",
            "SessionID": "ec3a0cab353d552952289f2c7ad52e27",
            "OS": "Operating System,2,0",
            "Type": "bdoc",
            "Vote": "UEsDBAoABgAAAAIAAAAbWltZXR5cGVhcHBsaWNhdGlv\
↪nbi92bmQuZX..."
        }
    ]
}
```

Päring `RPC.Vote` Mobiil-ID'ga autentimise korral.

```
{
    "id": 0.0,
    "method": "RPC.Vote",
    "params": [
        {
            "AuthMethod": "ticket",
            "AuthToken":
↪"G1RTZqBSBKrzqReuKYrmFUFXWFPvaxhJjdiZi6zqAnaK3OvrT...",
            "Choices": "0919.1",
            "SessionID": "057229fdfa2df7d3c7f4ced81b02760b",
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
            "OS": "Operating System,2,0",
            "Type": "bdoc",
            "Vote":
↪"UEsDBAoAAAAAAAAAACKIflFHwAAAB8AAAAIAAAAbWltZXR5cGVhcHB..."
        }
    ]
}
```

Hääletamisteenuse vastus päringule `RPC.Vote`.

result.Qualification.ocsp

result.Qualification.tspreg Kogumisteenuse poolt hangitud täiendavad tõendid valijarakenduse poolt loodud hääle `SignedVote` (*Valija poolt allkirjastatud hääl*) kvalifitseerimiseks ning korrektseks registreerimiseks. Vastuse koosseis sõltub kogumisteenuse konkreetsest seadistusest, antud juhul kasutatakse standardset OCSP protokolli valija allkirjasertifikaadi kehtivuse kontrolliks ning PKIX ajatempliprotokolli põhist registreerimisteenust nii hääle andmise aja fikseerimiseks kui elektroonilise hääle registreerimiseks välises sõltumatus teenuses. Valijarakendusele kontrollimiseks edastatakse nii OCSP vastus kui PKIX vormingus ajatempel koos registreerimisteenusele vajalike täiendustega.

result.TestVote Kui hääl esitati enne hääletamise algust ning läks arvesse proovihäälena, siis `true`, vastasel juhul seda välja vastuses ei ole. Valijarakendus kuvab valijale proovihääle korral sellekohase hoiatuse.

result.VoteID Hääle identifikaator talletusteenuses, mille alusel on kontrollrakendusel võimalik häält hilisemaks analüüsiks välja nõuda.

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "Qualification": {
            "ocsp":
↪"MIIFTAoBAKCCBUUwggVBBgkrBgEFBQcwAQEEggUyMIIFLjCB5qFMME...",
            "tspreg":
↪"MIIDsAYJKoZIhvcNAQcCoIIDoTCCA50CAQMxCzAJBgUrDgMCGgQS..."
        },
        "SessionID": "ec3a0cab353d552952289f2c7ad52e27",
        "TestVote": true,
        "VoteID": "VM/cUIU4n7VjxpUx1fC00Q=="
    }
}
```

Võimalikud veateated päringu `RPC.Vote` korral.

BAD_CERTIFICATE Viga valija isikutuvastus- või allkirjastamissertifikaadiga.

BAD_REQUEST Vigane päring.

IDENTITY_MISMATCH Isikutuvastus- ning allkirjastamissertifikaadi isikukoodid ei kattu.

INELIGIBLE_VOTER Valijal pole õigust hääletada.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

OUTDATED_CHOICES Valija ringkonnakuuluvus on nimekirja hankimisest muutunud.

UNAUTHENTICATED Autentimata päring.

VOTER_TOO_YOUNG Valija on liiga noor.

VOTING_END Hääletusperiood on lõppenud.

8.4 Hääletamine Mobiil-ID'ga

Mobiil-ID kasutamine allkirjastamis- ning autentimisvahendina tingib Mobiil-ID teenusega liidestuva abiteenuse (SNI `mid.ivxv.invalid`) kasutamise autentimistõendi hankimiseks enne valikute nimekirja hankimist ning hääle allkirjastamiseks enne talletamist.

8.4.1 Autentimistõendi hankimine

Valijarakendus teeb päringu `RPC.Authenticate` Mobiil-ID autentimise algatamiseks.

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

params.IDCode Mobiil-ID kasutaja isikukood.

params.PhoneNo Mobiil-ID kasutaja telefoninumber.

```
{
    "id": 0.0,
    "method": "RPC.Authenticate",
    "params": [
        {
            "OS": "Operating System,2,0",
            "IDCode": "60001019906",
            "PhoneNo": "+37200000766"
        }
    ]
}
```

result.Challenge Räsi, millest arvutada Mobiil-ID kontrollkood valijarakenduses kuvamiseks

result.SessionCode Mobiil-ID seansiidentifikaator edasiste poll-päringute jaoks

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "Challenge": "EtsTur4XV7xEGS9LBjHSfF9Cc5PQxtYW+YAOysRIt2r...
↪",
        "SessionCode": "2127729011",
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
    }
}
```

Võimalikud veateated päringu `RPC.Authenticate` korral.

BAD_REQUEST Vigane päring.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

VOTING_END Hääletusperiood on lõppenud.

Valijarakendus teeb päringu `RPC.AuthenticateStatus` autentimisprotsessi oleku hindamiseks.

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

params.SessionCode Autentimisseansi identifikaator

```
{
    "id": 0.0,
    "method": "RPC.AuthenticateStatus",
    "params": [
        {
            "OS": "Operating System,2,0",
            "SessionCode": "2127729011",
            "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
        }
    ]
}
```

result.AuthToken Autentimistõend teistele IVXV teenustele esitamiseks või `null`, kui päringu töötlemine alles käib.

result.GivenName Eduka autentimise korral valija eesnimi

result.PersonalCode Eduka autentimise korral valija isikukood

result.Status Päringu staatus - `POLL` viitab vajadusele päringut korrata, `OK` viitab edukale autentimisele. Vastuse muud väljad sisaldavad infot vaid siis kui väärtus on `OK`.

result.Surname Eduka autentimise korral valija perekonnanimi

```
{
    "error": null,
    "id": 0.0,
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)


```
    "result": {
        "AuthToken": null,
        "GivenName": "",
        "PersonalCode": "",
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b",
        "Status": "POLL",
        "Surname": ""
    }
}
```



```
{
    "error": null,
    "id": 0.0,
    "result": {
        "AuthToken":
↪"G1RTZqBSBKrzqReuKYrmFUFXWFPvaxhJjdiZi6zqAnaK3OvrT2Qu6...",
        "GivenName": "MARY \u00c4NN",
        "PersonalCode": "60001019906",
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b",
        "Status": "OK",
        "Surname": "O\u2019CONNE\u017d-\u0160USLIK"
    }
}
```


Võimalikud veateated päringu `RPC.AuthenticateStatus` korral.

BAD_REQUEST Vigane päring.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

MID_BAD_CERTIFICATE Viga valija Mobiil-ID isikutuvastussertifikaadiga.

MID_NOT_USER Telefoninumber ei kuulu Mobiil-ID kliendile.

MID_OPERATOR Probleem valija mobiiltelefoni SIM kaardiga, mille lahendamiseks tuleb pöörduda mobiilioperaatori poole.

MID_ABSENT Valija mobiiltelefon ei ole kättesaadav.

MID_CANCELED Valija katkestas Mobiil-ID seansi.

MID_EXPIRED Mobiil-ID seanss on aegunud.

MID_GENERAL Viga Mobiil-ID teenuse töös.

VOTING_END Hääletusperiood on lõppenud.

8.4.2 Hääle allkirjastamine

Valijarakendus teeb päringu `RPC.GetCertificate` allkirjastamissertifikaadi hankimiseks.

params.AuthMethod Toetatud ainult autentimismeetod `ticket`.

params.AuthToken Mobiil-ID autentimistõend.

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

params.PhoneNo Hääle allkirjastaja telefoninumber

```
{
    "id": 0.0,
    "method": "RPC.GetCertificate",
    "params": [
        {
            "AuthMethod": "ticket",
            "AuthToken":
↪"G1RTZqBSBKrzqReuKYrmFUFXWFPvaxhJjdiZi6zqAnaK3OvrT...",
            "OS": "Operating System,2,0",
            "PhoneNo": "+37200000766",
            "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
        }
    ]
}
```

result.Certificate Allkirjastamissertifikaat X509-vormingus

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "Certificate":
↪"MIIEVjCCAz6gAwIBAgIQRfmbsIcpkQ9UhxScCwG6VDANBgkqhki...",
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
    }
}
```

Võimalikud veateated päringu `RPC.GetCertificate` korral.

BAD_REQUEST Vigane päring.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

MID_BAD_CERTIFICATE Viga valija Mobiil-ID allkirjastamissertifikaadiga.

MID_GENERAL Viga Mobiil-ID teenuse töös.

MID_NOT_USER Telefoninumber ei kuulu Mobiil-ID kliendile.

VOTING_END Hääletusperiood on lõppenud.

Valijarakendus teeb päringu `RPC.Sign` hääle allkirjastamise algatamiseks. Mobiil-ID kontrollkoodi arvutab valijarakendus andmevälja `Hash` väärtusest.

params.AuthMethod Toetatud ainult autentimismeetod `ticket`.

params.AuthToken Mobiil-ID autentimistõend.

params.Hash BASE64-kodeeritud elektroonilise hääle räsi

params.HashType Räsifunktsiooni nimi Mobiil-ID teenusele edastamiseks, kas `SHA256`, `SHA384` või `SHA512`

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

params.PhoneNo Hääle allkirjastaja telefoninumber

```
{
    "id": 0.0,
    "method": "RPC.Sign",
    "params": [
        {
            "AuthMethod": "ticket",
            "AuthToken":
↪"G1RTZqBSBKrzqReuKYrmFUFXWFPvaxhJjdiZi6zqAnaK3OvrT...",
            "Hash": "9IBrA05ylt2StdjxKkSTYMW/rQXY3Vub4upzShdfEzo=",
            "HashType": "SHA256",
            "OS": "Operating System,2,0",
            "PhoneNo": "+37200000766",
            "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
        }
    ]
}
```

result.SessionCode Mobiil-ID seansiidentifikaator edasiste poll-päringute jaoks.

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "SessionCode": "E663A711BB9447EAD82491F9372F4CA",
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
    }
}
```

Võimalikud veateated päringu `RPC.Sign` korral.

BAD_REQUEST Vigane päring.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

VOTING_END Hääletusperiood on lõppenud.

Valijarakendus teeb päringu `RPC.SignStatus` allkirjastamisprotsessi seisundi hindamiseks.

params.OS Operatsioonisüsteem, millel valijarakendust kasutatakse.

params.SessionCode Mobiil-ID seansiidentifikaator

```
{
    "id": 0.0,
    "method": "RPC.SignStatus",
    "params": [
        {
            "OS": "Operating System,2,0",
            "SessionCode": "E663A711BB9447EAD82491F9372F4CA",
            "SessionID": "057229fdfa2df7d3c7f4ced81b02760b"
        }
    ]
}
```

result.Signature Juhul kui vastuse `Status` väli on `OK`, BASE-64 kodeeritud PKCS1-vormingus signatuur, vastasel juhul `null`.

result.Algorithm Juhul kui vastuse `Status` väli on `OK`, Mobiil-ID teenuse poolt tagastatud signatuuri algoritm. Võimalikud väärtused on `SHA256WithECEncryption`, `SHA256WithRSAEncryption`, `SHA384WithECEncryption`, `SHA384WithRSAEncryption`, `SHA512WithECEncryption` ja `SHA512WithRSAEncryption`.

result.Status Päringu staatus - `POLL` viitab vajadusele päringut korrata, `OK` viitab edukale allkirjastamisele. Vastuse muud väljad sisaldavad infot vaid siis kui väärtus on `OK`.

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b",
        "Signature": null,
        "Status": "POLL"
    }
}
```

```
{
    "error": null,
    "id": 0.0,
    "result": {
        "SessionID": "057229fdfa2df7d3c7f4ced81b02760b",
        "Signature": "MOj+8xQ9DmZPr/
↪ItHlm0tHNMCuTgn6dT9jcXjPLf0+2sVjsS11jRI...",
        "Algorithm": "SHA256WithECEncryption",
        "Status": "OK"
    }
}
```

Võimalikud veateated päringu `RPC.SignStatus` korral.

BAD_REQUEST Vigane päring.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

MID_ABSENT Valija mobiiltelefon ei ole kättesaadav.

MID_BAD_CERTIFICATE Viga valija Mobiil-ID allkirjastamissertifikaadiga.

MID_OPERATOR Probleem valija mobiiltelefoni SIM kaardiga, mille lahendamiseks tuleb pöörduda mobiilioperaatori poole.

MID_CANCELED Valija katkestas Mobiil-ID seansi.

MID_EXPIRED Mobiil-ID seanss on aegunud.

MID_GENERAL Viga Mobiil-ID teenuse töös.

VOTING_END Hääletusperiood on lõppenud.

8.5 Hääle kontrollimine

Kontrollrakendus teeb päringu `RPC.Verify` allkirjastatud hääle ning häält kvalifitseerivate tõendite allalaadimiseks kogumisteenusest.

params.OS Operatsioonisüsteem, millel kontrollrakendust kasutatakse.

params.VoteID QR-koodi vahendusel valijarakendusest saadud hääle identifikaator talletusteenuses.

```
{
    "id": 1,
    "method": "RPC.Verify",
    "params": [
        {
            "OS": "Operating System,2,0",
            "SessionID": "ec3a0cab353d552952289f2c7ad52e27",
            "VoteID": "VM/cUIU4n7VjxpUx1fC00Q=="
        }
    ]
}
```

result.Qualification.ocsp

result.Qualification.tspreg Vaata peatükki hääle verifitseerimisest

result.Type Allkirjastatud hääle vorming. Hetkel on ainus toetatud väärtus `bdoc`.

result.Vote BASE64-kodeeritud hääl `SignedVote` eelpoolmääratud vormingus (*Valija poolt allkirjastatud hääl*).

```
{
    "error": null,
    "id": 1,
    "result": {
        "Qualification": {
            "ocsp":
↪"MIIG8woBAKCCBuwwggboBgkrBgEFBQcwAQEEggbZMIIG1TCCASehgY...",
            "tspreg":
↪"MIIE0QYJKoZIhvcNAQcCoIIEwjCCBL4CAQMxDzANBglghkgBDQEJE..."
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
        },
        "SessionID": "027ab451969d9d3f044ea2cb2675b503",
        "Type": "bdoc",
        "Vote":
↪"UEsDBAoAAAAAAAAAACKIflFHwAAAB8AAAAIAAAAbWltZXR5cGVhcHB..."
    }
}
```

Võimalikud veateated päringu `RPC.Verify` korral.

BAD_REQUEST Vigane päring.

INTERNAL_SERVER_ERROR Viga serveri sisemises töös.

VOTING_END Hääletusperiood on lõppenud.

PEATÜKK 9

E-valimiskasti töötlemine

9.1 Tühistus- ja ennistusnimekiri

Tühistus- ja ennistusnimekiri sisaldab andmeid isikute kohta, kelle e-hääl tuleb tühistada (ei lähe arvesse valimistulemuste kokkulugemisel) või ennistada (s.t. tühistatakse eelnev tühistamine ning häälte uuesti üle lugemisel võetakse ennistatud e-hääl arvesse). Nimekiri laaditakse süsteemi digitaalselt allkirjastatud dokumendina, mille andmefaili vorming on järgmine:

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "definitions": {
        "rev_entry": {
            "type": "string",
            "pattern": "^[0-9]{11}$",
            "description": "Personal code of onlinevoter to be revoked"
        }
    },
    "type": "object",
    "properties": {
        "election": {
            "type": "string",
            "pattern": "^[ \\-,\\+\\.:;=!?&%#<>_/\\'\\*()\\[\\]{}|^A-Za-z0-9]{1,28}$"
        },
        "type": {"enum": ["revoke", "restore"]},
        "persons": {
            "type": "array",
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
            "items": {
                "$ref": "#/definitions/rev_entry"
            }
        }
    },
    "required": [
        "election",
        "persons",
        "type"
    ],
    "additionalProperties": false
}
```

Näide:

```
{
    "election": "TESTKOV",
    "persons": [
        "11412090004",
        "11412090005",
        "11412090006"
    ],
    "type": "revoke"
}
```

9.2 E-hääletanute nimekiri

E-hääletanute nimekiri on pärast e-hääletamise lõppu väljastatav nimekiri e-hääletanud isikutest, sorditуna valimisjaoskondade kaupa. Dokument genereeritakse töötlemisrakenduse poolt.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "onlinevoters_entry": {
            "type": "string",
            "pattern": "^[0-9]{11}$",
            "additionalItems": false
        },

        "onlinevoters": {
            "type": "array",
            "items": {
                "$ref": "#/definitions/onlinevoters_entry"
            },
            "additionalItems": false
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
        },

        "parish": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}$": {
                    "$ref": "#/definitions/onlinevoters"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        },

        "districts": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}\\.[0-9]{1,2}$": {
                    "$ref": "#/definitions/parish"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        }
    },
    "type": "object",
    "properties": {
        "election": {
            "type": "string",
            "pattern": "^[ \\-,\\+\\.:;=!?&%#<>_/\\'\\*()\\[\\]{}|^
↪A-Za-z0-9]{1,28}$"
        },
        "onlinevoters": {
            "$ref": "#/definitions/districts"
        }
    },
    "required": [
        "election",
        "onlinevoters"
    ],
    "additionalProperties": false
}
```

Näide:

```
{
  "election": "RK2030",
  "onlinevoters": {
    "0000.1": {
      "0176": [
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
        "11412090001"
      ],
      "0339": [
        "11412090002",
        "11412090003"
      ],
      "0614": [
        "11412090004"
      ],
      "0000": [
        "11412090005"
      ]
    },
    "0000.10": {
      "0793": [
        "11412090006",
        "11412090007",
        "11412090008"
      ],
      "0000": [
        "11412090009"
      ]
    }
  }
}
```

9.3 Hääletamistulemus

Võtmerakenduse poolt dekrüpteeritud ning summeeritud hääled jagatud valimisringkondade ja jaoskondade kaupa.

Hääletamistulemuste failis peavad iga jaoskonna kohta olema järgmised andmed.

1. Rikutud ja kehtetute häälte arvu näitav kirje. Seda ka juhul, kui valimisjaoskonnas polnud ühtki rikutud või kehtetut häält: sellisel juhul on häälte arv null.
2. Iga valiku poolt antud häälte arvu näitav kirje. Seda ka juhul, kui valimisjaoskonnas ei antud selle valiku poolt ühtki häält: sellisel juhul on häälte arv null.

```
1  {
2      "$schema": "http://json-schema.org/draft-04/schema#",
3      "definitions": {
4          "results": {
5              "type": "object",
6              "properties": {
7                  "invalid": {
8                      "type": "integer",
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
                    "description": "Number of invalid votes"
                }
            },
            "patternProperties": {
                "^[0-9]{4}\\.[0-9]{3,4}$": {
                    "type": "integer"
                }
            },
            "additionalProperties": false,
            "required": [
                "invalid"
            ]
        },
        "parish": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}$": {
                    "$ref": "#/definitions/results"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        },
        "district_dict": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}\\.[0-9]{1,2}$": {
                    "$ref": "#/definitions/results"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        },
        "parish_dict": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}\\.[0-9]{1,2}$": {
                    "$ref": "#/definitions/parish"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        }
    },
    "type": "object",
    "properties": {
        "election": {
            "type": "string",
            "pattern": "^[ \\-,\\+\\.:;=!?&%#<>_/\\'\\*()\\[\\]{}|^
↪A-Za-z0-9]{1,28}$"
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)

```
        },
        "bydistrict": {
            "$ref": "#/definitions/district_dict"
        },
        "byparish": {
            "$ref": "#/definitions/parish_dict"
        }
    },
    "required": ["election", "bydistrict", "byparish"],
    "additionalProperties": false
}
```

Näide:

```
{
    "bydistrict": {
        "0164.1": {
            "0164.126": 0,
            "0164.127": 0,
            "invalid": 0
        },
        "0296.1": {
            "0296.101": 0,
            "0296.102": 0,
            "0296.115": 0,
            "0296.116": 0,
            "0296.117": 0,
            "0296.198": 0,
            "0296.199": 0,
            "0296.200": 0,
            "invalid": 0
        }
    },
    "byparish": {
        "0164.1": {
            "0164": {
                "0164.126": 0,
                "0164.127": 0,
                "invalid": 0
            }
        },
        "0296.1": {
            "0296": {
                "0296.101": 0,
                "0296.102": 0,
                "0296.115": 0,
                "0296.116": 0,
                "0296.117": 0,
                "0296.198": 0,
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)


```
                "0296.199": 0,
                "0296.200": 0,
                "invalid": 0
            },
            "0296": {
                "0296.101": 0,
                "0296.102": 0,
                "0296.115": 0,
                "0296.116": 0,
                "0296.117": 0,
                "0296.198": 0,
                "0296.199": 0,
                "0296.200": 0,
                "invalid": 0
            }
        }
    },
    "election": "TESTKOV"
}
```


9.4 E-valimiskast

Fail sisaldab kogumisteenuse poolt vastu võetud hääli koos häälte juurde kuuluvate andmetega.

Faili vorming on Zip64 konteiner.

Valija-spetsiifilised kaustad asuvad vahetult juurkausta *votes* all.

Faili sisu:

- `votes/<voter id>/`
- `<timestamp>.version`
- `<timestamp>.<vote type>`
- `<timestamp>.<qualifier>*`

kus:

- `<voter id>` on valija identifikaator, Eesti puhul isikukood;
- `<timestamp>` on hääle esitamise kellaaeg vormingus `yyyymmddhhmmssmmm±zzzz`;
  - see kellaaeg kajastab hetke, mil päring kogumisteenusesse tehti, ja on antud lihtsalt valimiskasti inimloetavuse parandamiseks; hääle tegelik ajamärk või -tempel on mõne kvalifitseeriva vastuse sees;
- `<vote type>` on valikute konteineri tüüp, Eesti puhul BDOC;

  - kusjuures BDOC ise on lihtsalt põhiprofiiliga ja ei sisalda kvalifitseerivad parameetreid (kehtivuskinnitusi, ajamärgendeid, ajatempleid),
- `<qualifier>` on häält kvalifitseeriva protokolli tüüp, millest hetkel võimalikud on:
  - `ocsp` - *Online Certificate Status Protocol* (kehtivuskinnitus, RFC 6960[2]) kinnitab valija allkirjastamissertifikaadi kehtivust hääle andmise hetkel,
  - `ocsptm` - sama, mis `ocsp`, aga kasutab BDOC[3] spetsifikatsiooni jaotises 6.1 kirjeldatud laiendust, kus nonsiks pannakse hääle allkirja räsi, et häält ajamärgendada,
  - `tsp` - *Time-Stamp Protocol* (ajatempel, RFC 3161[4]) kinnitab, et päringu tegemise hetkeks oli hääl olemas,
  - `tspreg` - sama, mis `tsp`, aga nonsiks pannakse kogumisteenuse allkiri päringu `MessageImprint` elemendil, et häält registreerida.
- Iga hääle kohta esinevad failid on:

- `<timestamp>.version` - hääle andmise ajal kehtinud valijate nimekirja versioon;
- `<timestamp>.<vote type>` - valikute konteiner, mille sees on valiku identifikaator kujul `<valimise id>.<küsimuse id>.ballot`. Eesti puhul BDOC-konteineris olev vastava nimega fail;
- `<timestamp>.<qualifier>` - häält kvalifitseeriva protokolli päringu vastus; neid võib esineda mitu, aga iga protokolli kohta maksimaalselt üks.

9.5 Anonüümistatud e-valimiskast

Valimisringkondade ja jaoskondade järgi grupeeritud krüpteeritud hääled. Anonüümistatud e-valimiskastis puudub informatsioon valijate kohta.

Anonüümistatud e-valimiskast on töötlemisrakenduse väljund ning võtmerakenduse dekrüpteerimise tööriista sisend.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "results": {
            "type": "array",
            "items": {
                "type": "string"
            },
            "additionalItems": false
        },
        "questions": {
```

(jätkub järgmisel leheküljel)

[2] https://tools.ietf.org/html/rfc6960
[3] http://www.id.ee/public/bdoc-spec212-est.pdf
[4] https://tools.ietf.org/html/rfc3161

(jätk eelmisele leheküljele)

```
            "type": "object",
            "additionalProperties": {
                "$ref": "#/definitions/results"
            },
            "minProperties": 1
        },
        "parish": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}$|^(FOREIGN)$": {
                    "$ref": "#/definitions/questions"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        },
        "districts": {
            "type": "object",
            "patternProperties": {
                "^[0-9]{4}\\.[0-9]{1,2}$": {
                    "$ref": "#/definitions/parish"
                }
            },
            "additionalProperties": false,
            "minProperties": 1
        }
    },
    "type": "object",
    "properties": {
        "election": {
            "type": "string"
        },
        "districts": {
            "$ref": "#/definitions/districts"
        }
    },
    "required": [
        "election",
        "districts"
    ],
    "additionalProperties": false
}
```

Näide:

```
{
    "election": "TESTKOV",
    "districts": {
        "0164.1": {
```

(jätkub järgmisel leheküljel)

(jätk eelmisele leheküljele)


```
            "0164": {
                "TESTKOV.1": [
                    "MDkxOS4xMDUK",
                    "MDkxOS4xMDQK",
                    "MDkxOS4xMDEK",
                    "MDkxOS4xMDMK"
                ]
            }
        },
        "0296.1": {
            "0296": {
                "TESTKOV.1": [
                    "MDkxOS4xMDQK",
                    "MDkxOS4xMDQK"
                ]
            }
        }
    }
}
```

PEATÜKK 10

Hääletamistulemuse audit

10.1 Miksimistõendi kontroll

Miksimistõendi kontrollimiseks kasutatakse algoritmi nagu on defineeritud Verificatumi verifitseerija implementeerimise manuaalis[5].

Märgime, et miksimistõendi koostamisel lisatakse krüptogrammile andmed valimiste, ringkonna, jaoskonna ja küsimuse identifikaatori kohta. Lisamiseks kodeeritakse vastav väli rühma elemendina, kasutades pimendamiseks juhuslikkust 0. Näitena, kui esialgu on krüptogramm $c_0 = (c_{00}, c_{01})$, kasutades avalikku võtit $pk = (g, y)$, siis Verificatumi sisendina kasutatakse laia krüptogrammi $C = (c_{id}, c_d, c_s, c_q, c_0)$, kus:

- valimiste identifikaatori pseudokrüptogramm on antud kujul $c_{id} = (1, encode(id))$, kus funktsioon $encode$ kodeerib sõne vastava rühma elemendina ja *id* on valimiste identifikaatori sõne.
- ringkonna identifikaatori pseudokrüptogramm on antud kujul $c_d = (1, encode(d))$, kus *d* on ringkonna identifikaatori sõne.
- jaoskonna identifikaatori pseudokrüptogramm on antud kujul $c_s = (1, encode(s))$, kus *s* on jaoskonna identifikaatori sõne.
- küsimuse identifikaatori pseudokrüptogramm on antud kujul $c_q = (1, encode(q))$, kus *q* on küsimuse identifikaatori sõne.

Sellisel juhul defineeritakse laia krüptogrammile vastava avaliku võtmena $((g, 1), (g, 1), (g, 1), (g, 1), (g, y))$.

[5] https://www.verificatum.org/files/vmnv-3.0.3.pdf

10.2 Korrektse dekrüpteerimise tõendi kontroll

Olgu antud krüptogramm $c = (c_0, c_1)$, mis deküpteeritakse väärtuseks d antud avaliku võtmega pk üle parameetrite (p, g) ja dekrüpteerimistõendiga (a, b, s).

Korrektse dekrüpteerimise kontrollimise jaoks on tarvis arvutada mitte-interaktiivne kontrollija väljakutse. Selle jaoks kodeeritakse $"DECRYPTION"||pk||c||d||a||b$ DER-kodeeringus. Baidijada kasutatakse deterministliku juhuarvugeneraatori initsialiseerimiseks ja selle väljundist loetakse rühma järgu pikkune täisarv k.

Korrektse dekrüpteerimsie tõendi kontrolliks tuleb kontrollida, et $c_0^s = a * (c_1/d)^k$ ja $g^s = b * y^k$.

10.3 Korrektse teisendamise kontroll

Kontrollimaks, et teisendus IVXV e-valimiskasti ja Verificatumi krüptogrammide vahel on tehtud korrektselt, tuleb korrata teisendust sõltumatult. Pärast sõltumatut teisendust tuleb võrrelda saadud väljundeid. Kuna teisendamine on deterministlik protseduur, siis garanteerib kordamine tegevuse õigsuse.