Elektroonilise hääletamise protokollistik IV: Kesksüsteemi protokollid

Dokumendi liik

Redaktsioon: 1.2

18.04.2014

20 lk

Kuupäev	Nr	Kirjeldus	Autor
01.12.2012	0.1	Dokumendi algversioon	Cybernetica AS (Sven Heiberg)
18.07.2013	1.0	Muudatused metaandmetes	Cybernetica AS (Sven Heiberg)
18.09.2013	1.1	Räsi arvutamine	Cybernetica AS (Sven Heiberg)
18.04.2014	1.2	SHA-256 räsi arvutamine, BDOC2.1, uuenenud DDS protokoll, suletud nimekirjade eemaldamine	Cybernetica AS (Sven Heiberg)

Annotatsioon

Dokument spetsifitseerib elektroonilise hääletamise kesksüsteemi komponentide vahelised protokollid ning liidestumise väliste infosüsteemidega.

Sisukord

1 Sissejuhatus

1.1 Töö eesmärk

Dokument spetsifitseerib elektroonilise hääletamise kesksüsteemi komponentide vahelised protokollid ning liidesed väliste infosüsteemidega. Defineeritakse järgmised andmestruktuurid:

1. valimisjaoskondade ja -ringkondade nimekiri,
2. tühistus- ja ennistusnimekiri,
3. tühistamise tulemuste aruanne,
4. valikute nimekiri,
5. valijate nimekiri,
6. muudatuste nimekirja rakendamise tulemus,
7. e-hääletanute nimekiri,
8. logikirjete nimekiri,
9. loendamisele minevate häälte nimekiri,
10. hääletamistulemus.

Spetsifitseeritakse järgmised protokollid:

1. HES – HTS,
2. HES – Mobiil-ID teenus,
3. HTS – kehtivuskinnituse teenus.

1.2 Viited

1. [EHI] – Elektroonilise hääletamise protokollistik I: Üldkirjeldus.
2. [BDOC2.1] – BDOC. Digitaalallkirja vorming. Versioon 2.1:2013. http://www.id.ee/public/bdoc-spec21-est.pdf
3. [HTTP] – Hypertext Transfer Protocol – HTTP/1.1. RFC 2616
4. [Mobiil-ID] AS Sertifitseerimiskeskus. DigiDocService spetsifikatsioon. Dokumendi versioon: 2.127. Viimati uuendatud 01.04.2014. Kirjeldatav teenuse versioon: 3.5.*. http://www.sk.ee/files/DigiDocService_spec_est.pdf
5. [OCSP] – Online Certificate Status Protocol (RFC2560)

2 Andmestruktuurid

2.1 Üldpõhimõtted

Peatükk kehtestab üldpõhimõtteid, mida järgitakse kõigi andmestruktuuride kavandamisel.

1. Tagamaks täpitähtede ühest esitust kõigil platvormidel kasutatakse mistahes tekstiandmete kodeerimiseks UTF-8 kodeeringut.
2. Kõik andmestruktuurid sisaldavad versiooninumbrit. Antud spetsifikatsioonile vastavate andmestruktuuride korral on versiooninumber 1. Kõik käesoleva spetsifikatsiooni alusel loodud süsteemi komponendid peavad keelduma töötlemast muude versiooninumbritega andmeid.
3. Kõik süsteemi tööd mõjutavad andmed, mis on loodud konkreetse isiku poolt, on digitaalselt allkirjastatud. Kõik need andmed logitakse süsteemi poolt.
4. Digitaalselt allkirjastamata andmed võivad olla varustatud kontrollsummaga, mis võimaldab tuvastada andmete ülekandmisel tekkinud andmevigu.

2.1.1 Digitaalallkirjastatud andmed

Digitaalselt allkirjastatud andmete esitamiseks näeb spetsifikatsioon ette Eesti Vabariigi Standardi kavandis [BDOC2.1] defineeritud BDOC 2.1 allkirjavormingu kasutamise. Digitaalsel allkirjastamisel kehtivad järgmised kitsendused:

- Digitaalselt allkirjastatud dokument tohib sisaldada täpselt ühte `text/plain` tüüpi andmefaili.
- Digitaalselt allkirjastatud dokument esitatakse BDOC-TM vormingus ning peab sisaldama kehtivuskinnitust standardis RFC2560 defineeritud andmestruktuuri `OCSPResponse` kujul [OCSP].
- Andmefaili ja teiste signeeritavate andmeobjektide räsimiseks enne allkirjastamist kasutatakse räsifunktsiooni SHA-256.
- Dokument sisaldab kõiki standardi poolt vormingute BES/TM jaoks märgendiga M tähistatud elemente;
- Dokument ei sisalda mitte ühtegi standardi poolt vormingute BES/TM jaoks märgendiga C tähistatud elementi;
- Dokument ei sisalda mitte ühtegi standardi poolt vormingute BES/TM jaoks märgendiga N/A tähistatud elementi;
- Standardi poolt vormingute BES/TM jaoks märgendiga O märgitud elementidest võib kasutada elementi `SigningTime` kontekstis `SignedSignatureProperties`.

Juhul kui kasvõi üht neist tingimustest on rikutud, loetakse dokument rikutuks ja seda dokumenti ei võeta arvesse.

2.1.2 Kontrollsummaga kaitstud andmed

Andmevigade tuvastamist võimaldavate kontrollsummade arvutamine toimub järgmisel viisil:

1. Andmestruktuuri koostav rakendus valmistab ette ja salvestab ülekantava andmestruktuuri.
2. Rakendus arvutab salvestatud andmestruktuuri krüptograafilise kontrollsumma kasutades selleks SHA-1 või SHA-256 räsifunktsiooni. SHA-1 räsifunktsiooni väljundi pikkus binaarkujul on 20 baiti. Räsifunktsiooni väljund HEX-kodeertitakse ehk viiakse kuueteistkümnendkujule, kus iga bait on kujutatud kahekohalise kuueteistkümnendearvuna. Sellisel kujul oleva kontrollsumma pikkus on 40 baiti. SHA-256 räsifunktsiooni väljundi pikkus binaarkujul on 32 baiti. Räsifunktsiooni väljund HEX-kodeertitakse. Sellisel kujul oleva kontrollsumma pikkus on 64 baiti.
3. SHA-1 vormingus krüptograafiline kontrollsumma viiakse BASE64 kodeeringusse ning salvestatakse eraldi andmefaili, mille nimi langeb kokku algse andmefaili nimega, kuid millele on lisatud failinime laiend ".sha1". BASE64 kodeering ei tohi sisaldada muid märke (näiteks reavahetust) peale 40 baidi, mis kujutavad SHA1 räsi kuueteistkümnendsüsteemis.
4. SHA-256 vormingus krüptograafiline kontrollsumma salvestatakse HEX-kodeerituna eraldi andmefaili, mille nimi langeb kokku algse andmefaili nimega, kuid millele on lisatud failinime laiend ".sha256". BASE64 kodeeringut SHA-256 kontrollsumma jaoks ei rakendata.

OpenSSL käsurearakendust ning standardset Unixi utiliiti *tr* kasutades toimub faili SHA-1 kontrollsumma arvutamine järgmiselt:

```
openssl dgst -sha1 < fail | cut -d' ' -f2 | tr -d "\012" | openssl base64 >
fail.sha1
```

OpenSSL käsurearakendust ning standardset Unixi utiliiti *tr* kasutades toimub faili SHA-256 kontrollsumma arvutamine järgmiselt:

```
openssl dgst -sha256 < fail | cut -d' ' -f2 | tr -d "\012" > fail.sha256
```

Kontrollsumma kontrollimine toimub järgmisel viisil:

1. Enne andmete kasutamist tuvastab andmeid laadiv rakendus kasutatava kontrollsumma algoritmi.
2. Kui kontrollsumma algoritm on SHA-1, arvutab andmeid laadiv rakendus kontrollsumma üle sisendandmeid sisaldava faili, kasutades SHA-1 räsifunktsiooni. Seejärel loeb rakendus andmete koostaja poolt salvestatud kontrollsummat sisaldavas failis sisalduva BASE64 kodeeritud HEX kodeeritud kontrollsumma, dekodeerib selle ning võrdleb äsja arvutatud kontrollsummaga.
3. Kui kontrollsumma algoritm on SHA-256, arvutab andmeid laadiv rakendus kontrollsumma üle sisendandmeid sisaldava faili, kasutades SHA-256 räsifunktsiooni. Seejärel loeb rakendus andmete koostaja poolt salvestatud kontrollsummat sisaldavas failis sisalduva HEX kodeeritud kontrollsumma, dekodeerib selle ning võrdleb äsja arvutatud kontrollsummaga.
4. Juhul, kui arvutatud ja failist loetud kontrollsumma klapivad võib andmed laadida ning neid kasutada.
5. Juhul, kui arvutatud ja failist loetud kontrollsumma ei klapi tuleb väljastada veateade ning andmeid kasutada ei tohi.

2.1.3 Üldkirjelduses defineeritud andmestruktuurid

Dokument viitab mitmetele andmestruktuuridele, mis on defineeritud protokollistiku üldkirjelduses [EHI]:

```
versiooninumber

valimise-identifikaator

valiku-kood

ringkond

ringkonna-nimi

jaoskond

jaoskonna-ehak-kood

jaoskonna-number-omavalitsuses

jaoskonna-nimi

maakonna-nimi

valiku-id

valiku-nimi

valimisnimekirja-nimi

kandidaadi-nimi
```

Täiendavalt kasutatakse dokumendis järgmiseid andmestruktuure:

```
isikukood = 11*11DIGIT

nimi = 1*100UTF-8-CHAR

aeg = 14*14DIGIT
```

2.2 Valimisjaoskondade ja -ringkondade nimekiri

Nimekiri ringkondadest ja valimisjaoskondadest, koos nende nimede, numbrite ja valimisjaoskondade ringkonnakuuluvusega. Rahvahääletuse ja Euroopa Parlamendi valimiste korral on kirjeldatud üks ringkond, kuhu kuuluvad kõik valimisjaoskonnad.

Nimekiri laetakse süsteemi digitaalselt allkirjastatud dokumendina, mille andmefaili vorming on järgmine:

```
valimisjaoskondade-nimekiri = versiooninumber LF valimise-identifikaator LF
*ringkonna-kirjeldus *valimisjaoskonna-kirjeldus

ringkonna-kirjeldus = "ringkond" TAB ringkond TAB ringkonna-nimi LF

valimisjaoskonna-kirjeldus = "valimisjaoskond" TAB jaoskond TAB jaoskonna-nimi
TAB maakonna-nimi LF
```

2.3 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 laetakse süsteemi digitaalselt allkirjastatud dokumendina, mille andmefaili vorming on järgmine:

```
tühistus-ennistus-nimekiri = versiooninumber LF valimise-identifikaator LF
nimekirja-tüüp LF *tühistus-ennistuskirje

nimekirja tüüp = "tühistus" | "ennistus"

tühistus-ennistuskirje = isikukood TAB nimi TAB põhjus LF

põhjus = 1*100UTF-8-CHAR
```

2.4 Tühistamiste ja ennistamiste aruanne

Süsteem genereerib kontrollsummaga kaitstud tühistamiste ja ennistamiste aruande, mille vorming on järgmine:

```
tühistamiste-aruanne = versiooninumber LF valimise-identifikaator LF *kirje

kirje = tegevus TAB isikukood TAB nimi TAB hääle-laekumise-aeg TAB tegevuse-aeg
TAB *1operaatori-isikukood TAB *1põhjus LF

tegevus = "korduv e-hääl" |  "ennistatud" | "tühistatud"

hääle-laekumise-aeg = aeg

tegevuse-aeg = aeg

operaatori-isikukood = isikukood
```

Aeg on antud kujul YYYYMMDDhhmmss, kus YYYY on aastanumber, MM – kuu (01 – jaanuar, 12 – detsember), DD – kuupäev, hh – tunnid, mm – minutid, ss – sekundid. Automaatselt sooritatavatel tegevustel nagu korduvate e-häälte tühistamine jäetakse operaatori isikukoodi ning põhjuse lahtrid tühjaks.

2.5 Valikute nimekiri

Valikute nimekiri sisaldab andmeid kandidaatide (valimistel) või vastusevariantide (rahvahääletusel) kohta. Valimiste korral on lisaks kandidaadi andmetele nimekirjas ka tema valimisnimekirja nimi (sel juhul peavad valikute koodid olema organiseeritud nii, et sama valimisnimekirja kandidaadid oleksid ringkonniti grupeeritud).

Nimekiri laetakse süsteemi digitaalselt allkirjastatud dokumendina, mille vorming on järgmine:

```
valikute-nimekiri-kesksüsteemis = versiooninumber LF valimise-identifikaator LF
*kirje

kirje = valiku-id TAB valiku-nimi TAB *1valimisnimekirja-nimi TAB ringkond LF
```

Valiku nimi on kas reaalse isiku nimi (KOV, riigikogu, Europarlament) või tekst küsimusega (rahvahääletus).

2.6 Valijate nimekiri

Valijate nimekiri sisaldab valijate nimesid, isikukoode, valimisjaoskonda ning rea numbrit valimisjaoskonna valijate nimekirjas, milles valija hääletab. Valijate nimekiri laetakse süsteemi digitaalselt allkirjastamata dokumendina, mille vorming on järgmine:

```
valijate-nimekiri = versiooninumber LF valimise-identifikaator LF tüüp LF *valija

tüüp = "algne" | "muudatused"

valija = isikukood TAB nimi TAB tegevus TAB jaoskond TAB rea-number-voi-tyhi TAB
pohjus-voi-tyhi LF

tegevus = "lisamine" | "kustutamine"

rea-number-voi-tyhi = "" | rea-number

rea-number = 1*11DIGIT

pohjus-voi-tyhi = "" | pohjus

pohjus = "tokend" | "jaoskonna vahetus" | "muu"
```

Andmete sisu on järgmine.

1. Tüüp – "algne" tähistab seda esialgset suurt nimekirja, mis laetakse süsteemi enne e-hääletamise algust ja "muudatused" hilisemaid kumulatiivseid uuendusi.
2. Tegevus – "lisamine" tähendab uue valija lisamist nimetatud valimisjaoskonda ja "kustutamine" eemaldamist. Kui valija liigub ühest jaoskonnast teise, siis kantakse valijate nimekirja muudatuste hulka üks kustutamise kirje, millega valija oma eelmisest valimisjaoskonnast kustutatakse ja üks lisamise kirje, millega valija uues valimisjaoskonnas valijate nimekirja kantakse. Algses nimekirjas on kõik kirjed "lisamine" tüüpi.
3. Jaoskond – identifitseerib jaoskonna, ringkonna ja omavalitsuse, kus valija hääletab.
4. Rea-number – inimese rea number valimisjaoskonna nimekirjas. Täidetud ainult algse nimekirja puhul, muudatuste korral on see väli tühi.
5. Põhjus – kasutatakse kustutamiskirjete juures märkimaks kustutamise põhjust. Lisamiskirjete korral peab põhjus tühi olema. Kui põhjuseks on “tokend” tähendab see, et muudatuse rakendumisest alates ei tohi vastava isikukoodiga valija enam hääletada. Kui põhjuseks on “jaoskonna vahetus” tähendab see, et valija kustutatakse ühest jaoskonnast, kuna ta lisatakse teise jaoskonda. Sellisel juhul peab kaasnema kustutamiskirjega ka lisamiskirje (seda kontrollitakse). Kui kasutaja eemaldatakse nimekirjast mingil muul põhjusel (surm, mujale (piirkonda, mis ei

osale valimistel) elama kolimine), siis peab põhjuseks olema “muu” või võib põhjus tühi olema.

Peale “muudatused”-tüüpi nimekirja laadimist genereerib e-hääletussüsteem nimekirja tõkendi rakendamistest, mille korral valija oli juba enne tõkendi rakendumist hääletamas käinud.

```
tokendi-nimekiri = versiooninumber LF valimise-identifikaator LF *valija

valija = isikukood TAB nimi TAB jaoskond LF
```

2.7 E-hääletanute nimekiri

E-hääletanute nimekiri on pärast e-hääletamise lõppu väljastatav nimekiri e-hääletanud isikutest, sorteerituna valimisjaoskondade kaupa. Dokument genereeritakse süsteemi poolt ning on digitaalselt allkirjastamata. Kirjed on sorteeritud valimisjaoskonna numbri järgi ning seejärel valija nime järgi. Juhul, kui rea number valimisjaoskonna nimekirjas pole teada (see on võimalik kui valija lisati pärast valimiste algust), siis on see puudu. Dokumendi vorming on järgmine:

```
e-hääletanute-nimekiri = versiooninumber LF valimise-identifikaator LF *kirje

kirje = jaoskonna-ehak-kood TAB jaoskonna-number-omavalitsuses TAB *1rea-number
TAB isikukood TAB nimi LF
```

2.8 Logikirjete nimekiri

Süsteemi komponentide töö käigus tekib viis süsteemset logifaili, kuhu komponendid salvestavad andmed töödeldud häälte kohta. Süsteemis on kokku viis logi:

1. LOG1: vastvõetud hääled kujul: *aeg, hash(hääl), ringkond, valimisjaoskond, IK*
2. LOG2: tühistatud hääled kujul: *aeg, hash(hääl), ringkond, valimisjaoskond, IK, põhjus*
3. LOG3: lugemisesse läinud hääled kujul: *aeg, hash(hääl), ringkond, valimisjaoskond, IK*
4. LOG4: kehtetud sedelid – vale kandidaadi nr. kujul: *aeg, hash(hääl), ringkond*
5. LOG5: arvestatud hääled kujul: *aeg, hash(hääl), ringkond*

Dokument genereeritakse süsteemi poolt ning on digitaalselt allkirjastamata. Dokumendi vorming on järgmine:

```
logi = versiooninumber LF valimise-identifikaator LF logi-tüüp LF *kirje

logi-tüüp = "1" | "2" | "3" | "4" | "5"

kirje = aeg TAB hääle-räsi TAB jaoskond TAB *1valija-andmed LF

valija-andmed = isikukood TAB *1põhjus

hääle-räsi = 28*28BASE64-CHAR

põhjus = 1*100UTF-8-CHAR
```

Aeg on antud kujul YYYYMMDDhhmmss, kus YYYY on aastanumber, MM – kuu (01 – jaanuar, 12 – detsember), DD – kuupäev, hh – tunnid, mm – minutid, ss – sekundid.

Hääle räsi on saadud sisemise ümbriku binaarkujul räsimisel räsifunktsiooniga SHA-1 ning saadud tulemuse kodeerimisel BASE64 kujule.

2.9 Loendamisele minevate häälte nimekiri

Häältetalletusserveri poolt koostatud nimekiri jaoskonna numbriga varustatud krüpteeritud häältest, mis saadetakse häältelugemisrakendusele häälte kokkulugemiseks. Krüpteeritud hääl on viidud BASE64-kodeeritud kujule, millest on reavahetused eemaldatud. Andmevorming on kasutatav selliste krüpteeritud häälte esitamiseks, mille krüpteerimiseks kasutatud RSA võtme pikkus on vahemikus 1024-6144 bitti (turvaanalüüs nägi ette 2048-bitise võtme kasutamise).

Loendamisele minevate häälte nimekiri on allkirjastamata ning selle vorming on järgmine:

```
häälte-nimekiri = versiooninumber LF valimise-identifikaator LF *hääl

hääl = jaoskond TAB krüpteeritud-hääl LF

krüpteeritud-hääl = 172*1024BASE64-CHAR
```

2.10 Hääletamistulemus

Häältelugemisrakenduse poolt dešifreeritud ning summeeritud hääled sorteerituna valimisringkondade ja jaoskondade kaupa.

Hääletamistulemus on allkirjastamata ning nende vorming on järgmine:

```
hääletamistulemus = versiooninumber LF valimise-identifikaator LF
*hääletamistulemuse-rida

hääletamistulemuse-rida = jaoskond TAB häälte-arv TAB kelle-poolt LF

kelle-poolt = "kehtetu" | valiku-kood

häälte-arv = 1*11DIGIT
```

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.

3 Sõnumivahetus HES ja HTS vahel

3.1 Sõnumite struktuur

HES ja HTS vaheline transpordiprotokoll kasutab sõnumivahetuseks HTTP protokolli [HTTP]. HES sõnumid esitatakse POST-päringutena etteantud URLile. POST-päringu parameetrite binaarsetele väärtustele rakendatakse BASE64 kodeeringut. HTS vastussõnumid on text/plain tüüpi dokumendid, mille ülesehitus on kolmeosaline. Sõnum sisaldab protokolli versiooninumbrit ja protokolli sammust sõltuvaid oleku koodi ning teadet.

```
hts-sõnum = versiooninumber LF oleku-kood LF teade

oleku-kood = 0 | 1 | 2 | 3

teade = *1000UTF-8-CHAR
```

3.2 Kesksüsteemi kooskõlalisuse kontroll

Valijanimekirjade räsi arvutamine

Valijanimekirjasid laaditakse HES-i ja HTS-i. Selleks, et oleks võimalik kontrollida, kas mõlemas serveris on laetud samad sisendfailid, leitakse mõlemas serveris sisendfailide SHA1-räsi, mida hiljem saab kontrollimiseks kasutada. Valijanimekirjade sisendfailide räsi arvutatakse serveris järgmiselt:

- leitakse iga vastava sisendfaili SHA1-räsi, räsid sorteeritakse HEX-kodeeringus alfabeetiliselt ning võetakse SHA1 räsi sorteeritud räside konkatenatsioonist.

3.2.1 HES päring

Kooskõlalisuse kontroll käivitatakse, kui kesksüsteemi operaator käivitab vastava funktsionaalsuse HES kasutajaliidesest.

Kooskõlalisuse kontrolliks saadab HES POST päringu:

- `hash = valijanimekirjade-räsi`

3.2.2 HTS vastus

Kui HES ja HTS nimekirjad on kooskõlalised, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `1`. Väli `teade` on tühi.

Kui HES ja HTS nimekirjad ei ole kooskõlalised, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `0`. Välja `teade` sisu logitakse HES vealogisse.

Kui päringu teenindamisel tekkis viga, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `2`. Välja `teade` sisu logitakse HES vealogisse.

3.3 Korduvhääletamise kontroll

3.3.1 HES päring

Korduvhääletuse kontroll käivitatakse, kui valijarakendus pöördub HES poole valikute nimekirja saamiseks.

Korduvhääletamise kontrolliks saadab HES POST päringu:

- `hash = valijanimekirjade-räsi`
- `ik = isikukood`
- `session = seansiidentifikaator`

Väli `isikukood` sisaldab autenditud valija isikukoodi. Väli `seansiidentifikaator` sisaldab HES poolt antud valija jaoks genereeritud seansi identifikaatorit, mis tuleb logidesse kirjutada.

3.3.2 HTS vastus

Enne päringule vastamist kontrollib HTS kooskõlalisust. Seejärel kontrollib HTS, kas isikukoodiga identifitseeritud valija kohta on mõnel käimasolevatest valimistest talletatud hääl.

Kui HES ja HTS nimekirjad on kooskõlalised ning isikukoodiga identifitseeritud valija ei ole ühelgi käimasolevatest valimistest hääletanud, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `0`. Väli `teade` on tühi.

Kui HES ja HTS nimekirjad on kooskõlalised ning isikukoodiga identifitseeritud valija on mõnel käimasolevatest valimistest hääletanud, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `1`. Väli `teade` sisaldab kõigi valimiste identifikaatoreid, kus valija on osalenud.

Kui HES ja HTS nimekirjad ei ole kooskõlalised, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `2`. Välja `teade` sisu logitakse HES vealogisse.

Kui päringu teenindamisel tekkis viga, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `3`. Välja `teade` sisu logitakse HES vealogisse.

3.4 Hääle talletamine

3.4.1 HES päring

Hääle talletamine käivitatakse, kui HES on saanud digitaalselt allkirjastatud hääle.

Hääle talletamiseks saadab HES POST päringu:

- `hash = valijanimekirjade-räsi`
- `vote = BASE64(hääl-bdoc-idcard) | BASE64(hääl-bdoc-mobid)`
- `session = seansiidentifikaator`

Väli `vote` sisaldab digitaalselt allkirjastatud häält. Väli `seansiidentifikaator` sisaldab HES poolt antud valija jaoks genereeritud seansi identifikaatorit, mis tuleb logidesse kirjutada.

3.4.2 HTS vastus

Enne päringule vastamist kontrollib HTS kooskõlalisust. Kui digitaalallkirjal puudub kehtivuskinnitus, siis hangib HTS kehtivuskinnituse, kinnitab hääle omapoolse digitaalallkirjaga ning seejärel talletab hääle. Ühtlasi genereerib HTS kontrollprotokollis kasutatava unikaalse identifikaatori ning seostab talletatud hääle sellega.

Kui HES ja HTS nimekirjad on kooskõlalised ning hääle talletamine õnnestub, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `0`. Väli `teade` sisaldab unikaalset identifikaatorit `hääle-identifikaator`, mis tuleb valijarakendusele edastada.

Kui HES ja HTS nimekirjad ei ole kooskõlalised või hääle talletamisel tekib tehniline viga, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `1`. Välja `teade` sisu logitakse HES vealogisse.

Kui HES ja HTS nimekirjad on kooskõlalised kuid kehtivuskinnituse hankimine ei õnnestu, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `0`. Välja `teade` sisu logitakse HES vealogisse.

3.5 Hääle kontrollimiseks väljastamine

3.5.1 HES päring

Hääle kontrollimiseks väljastamine käivitatakse, kui HES on saanud kontrollrakenduselt valija hääle identifikaatori.

Hääle kontrollimiseks saadab HES POST päringu:

- `hash = valijanimekirjade-räsi`
- `verify = hääle-identifikaator`

Väli `verify` sisaldab talletamise käigus genereeritud unikaalset identifikaatorit.

3.5.2 HTS vastus

Enne päringule vastamist kontrollib HTS kooskõlalisust. Seejärel kontrollib HTS, et unikaalsele identifikaatorile vastav hääl on olemas ning selle väljastamine on lubatud nii ajalises kui kordade mõttes.

Kui HES ja HTS nimekirjad on kooskõlalised ning hääle väljastamine on võimalik, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `0`. Väli `teade` sisaldab andmestruktuuri `BASE64(ZIP(hääl-bdoc, bdoc-tempel, valikute-nimekiri))`, mis tuleb kontrollrakendusele edastada.

Kui HES ja HTS nimekirjad on kooskõlalised, aga konkreetse hääle kontrollimine ei ole enam võimalik, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `1`. Väli `teade` on tühi.

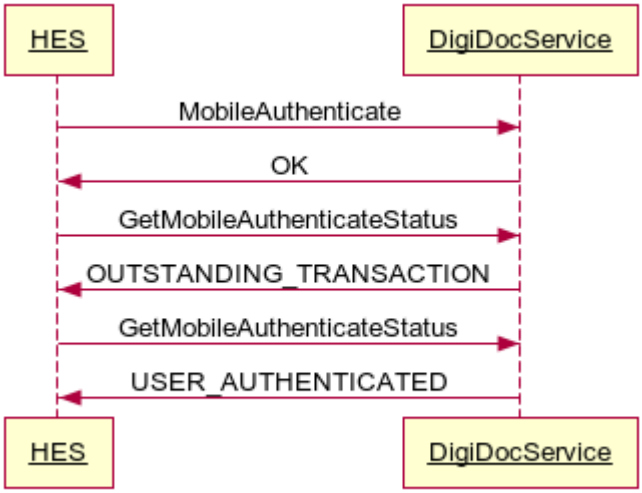
Kui HES ja HTS nimekirjad ei ole kooskõlalised, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `2`. Välja `teade` sisu logitakse HES vealogisse.

Kui päringu teenindamisel tekkis viga, siis on vastussõnumis `hts-sõnum` välja `oleku-kood` väärtus `3`. Välja `teade` sisu logitakse HES vealogisse.

4 Sõnumivahetus HES ja Mobiil-ID teenuse vahel

4.1.1 Valija autentimine

Mobiil-ID protokolli kasutamise korral suhtleb HES valija autentimiseks SOAP-põhist asünkroonset protokolli kasutades Mobiil-ID teenusega DigiDocService [Mobiil-ID]. Sõnumivahetus on kirjeldatud joonisel 1.

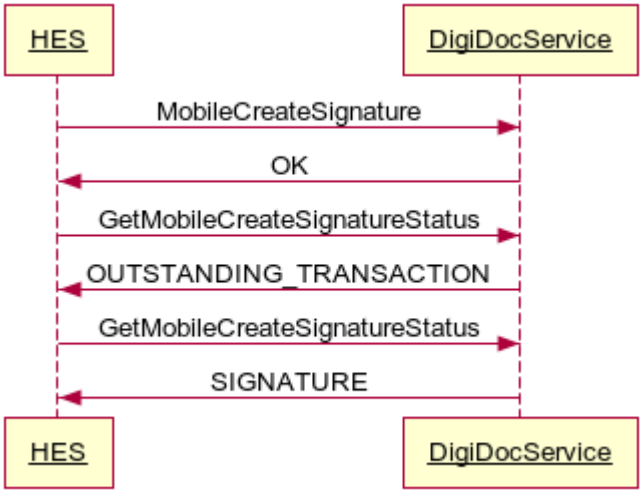


Joonis 1: Sõnumivahetus autentimiseks

1. HES edastatab valija poolt sisestatud telefoninumbri koos omapoolse `challenge` komponendiga päringuga `MobileAuthenticate` Mobiil-ID teenusele.
2. Mobiil-ID teenus vastab päringule valija autentimissertifikaadi ja omapoolse `challenge` komponendiga ning algatab suhtluse mobiiltelefoniga.
3. Mõne aja möödudes pollib HES Mobiil-ID teenust autentimispäringu õnnestumise suhtes päringuga `GetMobileAuthenticateStatus`.
4. Kui autentimispäring ei ole veel lõpule viidud vastab Mobiil-ID teenus teatega, mis nõuab uut pollimist mõne aja möödudes.
5. Mõne aja möödudes pollib HES Mobiil-ID teenust autentimispäringu õnnestumise suhtes päringuga `GetMobileAuthenticateStatus`.
6. Kui autentimine on lõpule viidud, siis vastab Mobiil-ID teenus Mobiil-ID autentimisvõtmega allkirjastatud ühise `challenge` parameetriga.
7. HES verifitseerib allkirja autentimissertifikaadi vastu, kui verifitseerimine õnnestub, siis oli autentimine edukas.

4.1.2 Hääle allkirjastamine

Mobiil-ID protokolli kasutamise korral suhtleb HES hääle allkirjastamiseks SOAP-põhist asünkroonset protokolli kasutades Mobiil-ID teenusega (DigiDocService). Sõnumivahetus on kirjeldatud joonisel 2.



Joonis 2: Sõnumivahetus allkirjastamiseks

1. HES koostab allkirjastatavate andmefailide räside komplekti ning algatab päringuga `MobileCreateSignature` asünkroonse allkirjastamisseansi Mobiil-ID teenusega.
2. Mobiil-ID teenus vastab päringule `challenge` komponendiga ning algatab suhtluse mobiiltelefoniga.
3. Mõne aja möödudes pollib HES Mobiil-ID teenust allkirjastamispäringu õnnestumise suhtes päringuga `GetMobileCreateSignatureStatus`.
4. Kui allkirjastamispäring ei ole veel lõpule viidud vastab Mobiil-ID teenus teatega, mis nõuab uut pollimist mõne aja möödudes.
5. Mõne aja möödudes pollib HES Mobiil-ID teenust allkirjastamispäringu õnnestumise suhtes päringuga `GetMobileCreateSignatureStatus`.
6. Kui allkirjastamine on lõpule viidud, tagastab Mobiil-ID allkirjafaili.

4.1.3 Mobiil-ID veakoodid

HES peab Mobiil-ID teenusega suheldes olema valmis töötlema vähemalt järgmiseid vigu:

SOAP vead

- 301, 201 – telefoninumber ei kuulu Mobiil-ID kasutajale
- 302 – sertifikaat on kas tühistatud või peatatud
- 303 – Mobiil-ID teenus on aktiveerimata

Mobiil-ID protokolli vead

- USER_CANCEL – PIN sisestus katkestati

- PHONE_ABSENT – telefon ei ole levis
- SENDING_ERROR – viga sõnumi saatmisel
- SIM_ERROR – SIM kaardi viga
- INTERNAL_ERROR – tehniline viga
- REVOKED – sertifikaat on tühistatud
- SUSPENDED – sertifikaat on peatatud
- NOT_ACTIVATED – Mobiil-ID teenus on aktiveerimata

5 Sõnumivahetus HTS ja kehtivuskinnitusteenuse vahel

HTS kasutab digitaalselt allkirjastatud häälele kehtivuskinnituse hankimiseks OCSP protokolli [OCSP]. Protokoll on kahe päringuline:

1. HTS koostab ASN.1 andmestruktuuri `OCSPRequest` vastavalt RFC2560 reeglitele. Päringu koostamisel kasutatakse laiendust `Nonce`, mille väärtuseks on signatuuri räsi, mis on kodeeritud spetsifikatsiooni [BDOC2.1] nõuetele vastavalt.
2. Kehtivuskinnitusteenus vastab päringule digitaalselt allkirjastatud andmestruktuuriga `OCSPResponse`, mis sisaldab eelmises päringus kasutatud nonssi ning kinnitab sertifikaadi olekut. Positiivne vastus talletatakse HTSis koos häälega kehtivuskinnitusena.