



Euroopa Liit  
Euroopa  
Regionaalarengu Fond



Eesti  
tuleviku heaks

## SPOF2.3 – eID infrastruktuuri usaldusmudel

Uuringuaruanne

Versioon: 1.0

28. oktoober 2022

127 lehekülge

Dokumendi nr: D-16-158



Projektijuhid: Tõnis Reimo (Riigi Infosüsteemi Amet)  
Sandhra-Mirella Valdma (Cybernetica)

Autorid: Aleksander Kamenik (Cybernetica)  
Peeter Laud (Cybernetica)  
Alisa Pankova (Cybernetica)  
Triin Siil (Cybernetica)  
Nikita Snetkov (Cybernetica)

Riigi Infosüsteemi Amet, Pärnu maantee 139a, 15169 Tallinn, Eesti.

Email: [ria@ria.ee](mailto:ria@ria.ee), Web: <https://www.ria.ee>, Telefon: +372 663 0200.

Cybernetica AS, Mäéaluse 2/1, 12618 Tallinn, Eesti.

E-mail: [info@cyber.ee](mailto:info@cyber.ee), Web: <https://www.cyber.ee>, Telefon: +372 639 7991.

© Riigi Infosüsteemi Amet, 2022

# Kommenteeritud kokkuvõte

Kaks aastat tagasi ilmunud, Riigi Infosüsteemi ameti tellimisel valminud uuring "eID infrastruktuuri tõrkekindluse analüüs" tuvastas, et üheks eID taristu sõlmkomponendiks on sertifitseerimistaristu: enam-vähem kõik meie elektroonilise isikutuvastuse vahendid sõltuvad ühestainsast sertifitseerimiskeskusest. Sertifitseerimiskeskuse roll eID taristus on teatavasti kodanike identiteetide sidumine krüptograafiliste võtmepaaridega ja nende seoste töendamine. Tegemist on *usaldusteenusega*, see tähendab, et serifitseerimiskeskuse loodud töendite vastavust reaalsele eluga ei ole võimalik lõpuni kontrollida, vaid nende õigsust tuleb uskuda keskuse reputatsioonile tuginedes. Kui selle keskuse ja selle pakutavate sidumis- ja töendamisteenustega midagi juhtub, siis võib takistatud olla enam-vähem kõigi teiste e-teenuste kasutamine. Tegemist on seega väga olulise nõrkusega eID taristus. Eelmainitud uuring hindab ohuallikate võimekust seda nõrkust ära kasutada küll väga madalaks, kuid tegemist on siiski eID taristu piisavalt olulise riskiga, et uurida, kuidas vähendada sõltuvust ühestainsast sertifitseerimiskeskusest. Seega tellis Riigi Infosüsteemi amet uuringu, mille tulemusena on valminud käesolev uuringuaruanne.

Käesolevas aruandes me kirjeldame kodanike identiteetide krüptograafiliste võtmepaaridega sidumise teoreetiliste aluste, tehnoloogiate ja teenuste hetkeseisu. Me analüüsime, kuidas uskumine mingi identiteedi ja võtmepaari seotusse saab edasi kanduda ühelt olemilt teisele. Kirjeldame erinevaid teoreetilisi, teadus- ja tehnilises kirjanduses väljapakutud võimalusi usaldustaristu ülesseadmiseks ning analüüsime, kuidas seal usaldus edasi kandub ning kui tõrkekindlad, kuid samas kulukad need on. Selgitame, kuivõrd on need ülesseadmisiidid toetatud standardsete andmestruktuuride ja protokollidega. Anname ka ülevaate tõrgetest ja intsidentitest, mis on tegelikult sertifitseerimiskustega juhtunud, ning kategoriseerime nende intsidentide põhjused. Need üldised kirjeldused on vormistatud käesoleva aruande ingliskeelsete lisadena, pidades silmas ka tõenäolist väljastpoolt Eestit lähtuvat huvi nende vastu.

See olemasoleva olukorra kirjeldus on aluseks meie uuringu põhitulemusele: soovitusele, kuidas Eestis muuta sertifitseerimistaristut nõnda, et üksainus sertifitseerimiskeskus ei oleks enam eID taristu nõrk lüli. Soovitust andes pidasime silmas, et muudatused ühilduksid kasutuses olevate standarditega, et väljapakutav lahendus mahuks olemasolevasse usaldusteenuse õiguslikku raamistikku ja ei oleks liialt kallis ülal pidada.

Me soovitame, et ühe sertifitseerimiskeskuse asemel oleks taristus kaks teineteisest sõltumatut keskust, s.t. riik peab looma tingimused, mis motiveeriks kahel erineval teenusepakkujal Eestis sertifitseerimisteenust pakkuda. Siiski ei tee me ettepanekut sertifitseerimistaristu keerukust ja kulusid kahekordistada. Me nimelt soovitame, et ainult üks neist kahest sertifitseerimiskeskusest peaks olema *aktiivne*, mis väljastab kõik sertifikaadid ja vastab sertifikaatide kehtivuspäringutele. Teine sertifitseerimiskeskus on *passiivne*, mis peaks tema jooksvaid kulusid palju väiksemana hoidma. Teine keskus peab küll enda juures hoidma koopiat kõigist esimese sertifitseerimiskeskuse väljastatud sertifikaatide olekutest, seda pidevalt ajakohastades. Kui esimese keskusega midagi juhtub, siis toimub *ümberlülitumine* teisele keskusele, mis muutub nüüd aktiivseks. Ümberlülitamise käigus peab teine sertifitseerimiskeskus väljastama koopiad esimese keskuse väljastatud sertifikaatidest, mis hetkel kehtivad.

Käideldavuse poolest on sertifitseerimiskeskus meie lahenduses seega dubleeritud: kui üks keskus enam päringutele ei vasta, töötavad teenused edasi. Usaldama peame aga mõlemaid keskusi (mis peavad seega mõlemad olema kantud ka Euroopa Liidu usaldusteenuse pakkujate registrisse): meie

lahendus ei paku välja täiendavaid võimalusi ühe keskuse väljastatud sertifikaatide ja kehtivuskinnituste võrdlemiseks mingi muu informatsiooniga. Teataval määral parandab meie lahendus kyll väljastatud sertifikaatide registri terviklust, sest sellest tuleb nüüd teha rohkem koopiaid.

Meie väljapakutud lahenduse hea külg on, et kuni ümberlülitumiseni jäääb lõppkasutaja kogemus täpselt samasuguseks nagu praegu. Ümberlülitumisel tuleb lõppkasutajate sertifikaadid asendada. Teenustel nagu Smart-ID ja Mobiil-ID, kus kasutaja sertifikaate hoitakse teenusepakkija serveris, võib see vahetamine jääda lõppkasutaja jaoks vähemärgatavaks, sest uued sertifikaadid on võimalik sinna serverisse laadida otse teisest sertifitseerimiskeskusest. ID-kaardi puhul hoitakse sertifikaate aga kaardil. Nende uuendamiseks peab lõppkasutaja kasutama DigiDoc-rakendust.

# Sisukord

Kommenteeritud kokkuvõte .....	3
1. Sissejuhatus .....	12
1.1. Töö eesmärk .....	12
1.2. Usaldusteenused, -mudelid ja -taristu .....	13
1.3. Käesoleva aruande sisu .....	13
1.4. Kasutatud metoodika .....	14
1.5. Määratlused ja lühendid .....	15
1.6. Viited .....	20
2. Järeldused analüüsist .....	32
3. Soovituslik eID usaldusmuodel .....	35
3.1. Ülesandepüstitus .....	35
3.2. Arhitektuurivalikute arutelu .....	36
3.2.1. Võimalikud nõrgad lülid eID süsteemides .....	36
3.2.2. Nõrkade lülide võimalikud mõjud ja nende leevendamine .....	37
3.2.3. Lävi-PKI kasutuselevõtt ja sellega seotud kitsaskohad .....	38
3.2.3.1. Stabiilsus .....	38
3.2.3.2. Usaldusväärssus .....	38
3.2.3.3. Paindlikkus .....	38
3.3. Pakutud arhitektuur .....	39
3.3.1. Osapooled .....	39
3.3.2. TSP-de eri usaldustasemed .....	39
3.3.3. Usaldusteenuse pakkujate andmebaas .....	40
3.3.4. Usaldusmuodel .....	41
3.3.5. Protsessid .....	41
3.3.5.1. Uue TSP saabumine .....	41
3.3.5.2. TSP asub pakkuma teatud usaldusteenust .....	41
3.3.5.3. TSP lõpetab teatud usaldusteenuse osutamise .....	42
3.3.5.4. TSP lahkub turult (vabatahtlikult) .....	42
3.3.5.5. TSP sunnitakse lahkuma .....	42
3.3.5.6. Olem küsib ja saab sertifikaadi teatud teenusest .....	43
3.3.5.7. Olem kontrollib sertifikaati .....	44
3.3.5.8. Olem autendib ennast sõltuvale osapoolele .....	44
3.3.5.9. Olem kontrollib signatuuri .....	44
3.3.5.10. Ennustamine, kas kolmas osapool aktsepteerib signatuuri .....	44
3.3.5.11. Olem kasutab digiallkirjateenust .....	45
3.3.5.12. Väljaandja tühistab sertifikaadi .....	45
3.3.6. Aktiivsed ja passiivsed TSP-d .....	45
3.4. Väljapakutud arhitektuuri kitsaskohad .....	47

3.4.1. Mitme sertifikaadi kontrollimine .....	47
3.4.1.1. Mitme sertifikaadiga avalikud võtmehad .....	47
3.4.1.2. Lävisigneeritud sertifikaadiga avalikud võtmehad .....	47
3.4.2. Kas ühe või teise sertifikaadi kontrollimine .....	48
3.4.2.1. Mitu erinevat võtit, igaühel üks sertifikaat .....	48
3.4.2.2. Üks võti ja üks sertifikaat .....	49
3.5. Meie soovitatav lahendus .....	49
3.5.1. Ajatembedamiseks .....	49
3.5.2. Sertifitseerimiseks .....	50
3.6. Kvantitatiivne analüüs .....	53
4. Õiguslik hinnang .....	54
4.1. Sissejuhatus .....	54
4.2. Asjaolude kirjeldus .....	55
4.3. Õiguslik analüüs .....	56
4.3.1. Õiguslikud küsimused .....	56
4.3.2. Ülevaade kehtiva õiguse raamistikust .....	56
4.3.2.1. Euroopa Liidu õigus .....	56
4.3.2.2. Eesti õigus .....	57
4.3.3. Üldised õiguslikud nõuded soovitusliku eID usaldusmudeli toimimiseks .....	57
4.3.4. Hinnang üldiste õiguslike nõuete teostatavusele kehtiva õiguse alusel .....	59
5. Kokkuvõte .....	61
6. Summary in English .....	63
A. Appendix: Incidents through failing CAs .....	65
A.1. Single point of failure cases of PKI .....	65
A.1.1. VeriSign and Microsoft (2001) .....	65
A.1.1.1. Description of the incident .....	65
A.1.1.2. The reasons and mitigation .....	65
A.1.2. Thawte (2008) .....	66
A.1.2.1. Description of the incident .....	66
A.1.2.2. The reasons and mitigation .....	66
A.1.3. StartSSL (2008) .....	66
A.1.3.1. Description of the incident .....	66
A.1.3.2. The reasons and mitigation .....	66
A.1.4. Comodo and CertStart (2008) .....	67
A.1.4.1. Description of the incident .....	67
A.1.4.2. The reasons and mitigation .....	67
A.1.5. VeriSign (2010) .....	67
A.1.5.1. Description of the incident .....	67
A.1.5.2. The reasons and mitigation .....	67
A.1.6. Comodo (2011) .....	67
A.1.6.1. Description of the incident .....	67

A.1.6.2. The reasons and mitigation .....	68
A.1.7. StartSSL (2011) .....	68
A.1.7.1. Description of the incident .....	68
A.1.7.2. The reasons and mitigation .....	68
A.1.8. DigiNotar (2011) .....	69
A.1.8.1. Description of the incident .....	69
A.1.8.2. The reasons and mitigation .....	69
A.1.9. GlobalSign (2011) .....	70
A.1.9.1. Description of the incident .....	70
A.1.9.2. The reasons and mitigation .....	70
A.1.10. Pos Digicert Sdn.Bhd. (2011) .....	70
A.1.10.1. Description of the incident .....	70
A.1.10.2. The reasons and mitigation .....	70
A.1.11. Trustwave (2012) .....	71
A.1.11.1. Description of the incident .....	71
A.1.11.2. The reasons and mitigation .....	71
A.1.12. Cyberoam (2012) .....	71
A.1.12.1. Description of the incident .....	72
A.1.12.2. The reasons and mitigation .....	72
A.1.13. TURKTRUST (2012) .....	72
A.1.13.1. Description of the incident .....	72
A.1.13.2. The reasons and mitigation .....	72
A.1.14. ANSSI (2013) .....	73
A.1.14.1. Description of the incident .....	73
A.1.14.2. The reasons and mitigation .....	73
A.1.15. NIC of India (2014) .....	73
A.1.15.1. Description of the incident .....	73
A.1.15.2. The reasons and mitigation .....	74
A.1.16. Comodo (2015) .....	74
A.1.16.1. Description of the incident .....	74
A.1.16.2. The reasons and mitigation .....	74
A.1.17. CNNIC (2015) .....	74
A.1.17.1. Description of the incident .....	75
A.1.17.2. The reasons and mitigation .....	75
A.1.18. Symantec (2015) .....	75
A.1.18.1. Description of the incident .....	75
A.1.18.2. The reasons and mitigation .....	76
A.1.19. SK ID Solutions (2015) .....	76
A.1.19.1. Description of the incident .....	76
A.1.19.2. The reasons and mitigation .....	76
A.1.20. StartSSL and WoSign (2015-2016) .....	76

A.1.20.1. SHA-1 Certificates (January-March 2015) . . . . .	77
A.1.20.1.1. Description of the incident . . . . .	77
A.1.20.1.2. The reasons and mitigation . . . . .	77
A.1.20.2. Two identical certificates with different NotBefore date/time (March 2015) . . . . .	77
A.1.20.2.1. Description of the incident . . . . .	77
A.1.20.2.2. The reasons and mitigation . . . . .	78
A.1.20.3. Various Violations of Baseline Requirements (April 2015) . . . . .	78
A.1.20.3.1. Description of the incident . . . . .	78
A.1.20.3.2. The reasons and mitigation . . . . .	78
A.1.20.4. Any port for a domain validation(January-March 2015) . . . . .	78
A.1.20.4.1. Description of the incident . . . . .	78
A.1.20.4.2. The reasons and mitigation . . . . .	78
A.1.20.5. Secret acquisition of StartCom/StartSSL (November 2015) . . . . .	79
A.1.20.6. Backdated SHA-1 Certificates (January 2016) . . . . .	79
A.1.20.7. Aftermath . . . . .	79
A.1.21. Symantec (2016) . . . . .	79
A.1.21.1. Description of the incident . . . . .	79
A.1.21.2. The reasons and mitigation . . . . .	80
A.1.22. Comodo and PositiveSSL (July 2016) . . . . .	80
A.1.22.1. Description of the incident . . . . .	80
A.1.22.2. The reasons and mitigation . . . . .	80
A.1.23. Comodo (October 2016) . . . . .	80
A.1.23.1. Description of the incident . . . . .	80
A.1.23.2. The reasons and mitigation . . . . .	81
A.1.24. GoDaddy (2017) . . . . .	81
A.1.24.1. Description of the incident . . . . .	81
A.1.24.2. The reasons and mitigation . . . . .	81
A.1.25. Symantec (2017-2018) . . . . .	81
A.1.25.1. Description of the incident . . . . .	81
A.1.25.2. The reasons and mitigation . . . . .	82
A.1.26. Certinomis (2018) . . . . .	82
A.1.26.1. Description of the incident . . . . .	82
A.1.26.2. The reasons and mitigation . . . . .	82
A.1.27. Kazakhstan Root CA (2019) . . . . .	83
A.1.27.1. Description of the incident . . . . .	83
A.1.27.2. The reasons and mitigation . . . . .	83
A.2. Discussion of incidents . . . . .	83
A.2.1. Scenarios . . . . .	83
A.2.2. Prevention and incident response techniques . . . . .	84
A.3. Incidents affecting availability . . . . .	85
B. Appendix: Proposed trust models . . . . .	86

B.1. Speaking about beliefs and trust .....	86
B.2. Trust models without weak responsiveness .....	87
B.3. Classical PKI .....	88
B.3.1. Dedicated Domain PKI .....	89
B.3.2. Shared Domain PKI .....	89
B.3.3. Mutual exchange .....	89
B.3.4. Domain trusted lists .....	90
B.3.4.1. eIDAS trust list .....	91
B.4. Mesh PKIs (with multiple/distributed CAs) .....	91
B.4.1. Without Thresholds .....	91
B.4.1.1. Mesh PKI .....	91
B.4.1.2. Hierarchical PKI .....	92
B.4.1.3. Bridge CA .....	93
B.4.1.4. Bridge Validation Authority (VA) .....	93
B.4.2. With Thresholds .....	94
B.4.3. Guardtime KSI .....	95
B.4.4. Web of Trust on a Distributed Ledger .....	97
B.4.4.1. Trust model .....	97
B.4.4.2. Example .....	97
B.4.4.3. Summary .....	98
B.4.5. ESSIF .....	98
B.4.5.1. Roles .....	99
B.4.5.1.1. Issuer .....	99
B.4.5.1.2. Holder .....	99
B.4.5.1.3. Verifier .....	99
B.4.5.1.4. ESSIF Onboarding Service (EOS) .....	100
B.4.5.2. Trust model .....	100
B.4.5.3. Discussion .....	101
B.5. Qualitative comparison .....	103
B.5.1. Signing a document .....	103
B.5.1.1. Classical PKI .....	103
B.5.1.2. Web of Trust .....	103
B.5.1.3. Generic SSI .....	103
B.5.1.4. ESSIF .....	103
B.5.2. Signature verification .....	104
B.5.2.1. Classical PKI .....	104
B.5.2.2. Web of Trust .....	104
B.5.2.3. Generic SSI .....	104
B.5.2.4. ESSIF .....	104
B.5.3. Authentication .....	104
B.5.3.1. Both the server and the client are Estonian .....	105

B.5.3.2. The server is Estonian, the client is from EU .....	105
B.5.3.3. The server is Estonian, the client is outside of EU .....	105
B.5.3.4. The client is Estonian, the server is from EU .....	105
B.5.3.5. The client is Estonian, the server is outside of EU .....	106
<b>B.6. Quantitative comparison .....</b>	<b>106</b>
B.6.1. Costs .....	106
B.6.2. Attack model .....	107
B.6.3. Analysis results .....	108
B.6.3.1. Dedicated domain .....	109
B.6.3.2. Shared domain .....	109
B.6.3.3. Mutual Exchange .....	109
B.6.3.4. Trusted Lists .....	110
B.6.3.5. Hierarchical PKI .....	110
B.6.3.6. Bridge CA .....	110
B.6.3.7. Bridge VA .....	110
B.6.3.8. General mesh .....	111
B.6.3.9. Threshold .....	111
B.6.3.10. Double CA .....	111
B.6.3.10.1. Cold standby .....	111
B.6.3.10.2. Hot balancing .....	111
B.6.3.10.3. Active-active .....	111
B.6.3.10.4. Active-active-active .....	112
B.6.3.11. KSI .....	112
B.6.3.12. Web of Trust .....	112
B.6.3.13. ESSIF Model .....	112
<b>B.7. High-level comparison from the legal perspective .....</b>	<b>112</b>
B.7.1. Classical PKI .....	113
B.7.2. Mesh PKI with thresholds .....	113
B.7.3. KSI .....	113
B.7.4. SSI .....	113
<b>C. Appendix: Applicable laws .....</b>	<b>115</b>
C.1. Introduction .....	115
C.2. Applicable law .....	115
C.2.1. Individual identity system rules .....	115
C.2.2. Generic identity system law .....	116
C.2.3. General law .....	117
<b>D. Lisa: Väljapakutud arhitektuuri kvantitatiivne analüüs .....</b>	<b>118</b>
<b>E. Appendix: Logic for describing trust models .....</b>	<b>122</b>
E.1. Syntax .....	122
E.2. Expressing trust relationships .....	123
E.3. Axioms and inference rules .....	124

E.4. Semantics . . . . .	126
--------------------------	-----

# 1. Sissejuhatus

## 1.1. Töö eesmärk

Hiljuti tellis Riigi Infosüsteemi amet uuringu "eID infrastruktuuri tõrkekindluse analüüs", mille uuringuaruanne [1] ilmus 2020. aasta suvel. Selles uuringus püüti tuvastada eID taristu komponentide omavahelisi sõltuvusi ning leida nende sõltuvuste sõlm punkte, kus üksikutest komponentidest sõltub väga palju. Uuring leidis, et üheks selliseks sõlmkomponendiks on sertifitseerimistaristu:

Juhul, kui seda ei ole spetsiaalselt teistmoodi korraldatud, siis tavapäraselt sõltuvad kõik isikutuvastusvahendid ühest CA-st.

— eID infrastruktuuri tõrkekindluse analüüs uuringuaruanne, jaotis 3.6

Sõlmkomponendi kannatadasaamise riski set hinnates iseloomustati võimaliku ohuolukorra parameetreid järgmiselt:

Ohuolukorra mõju avaldumise võimalikkuse hinnang: *väga kõrge*, kuna praegu pole rakendatud ühtegi kompenseerivat meedet, mis CA tühistamise mõju vähendaks.

Ohuolukorra mõju kaalukuse hinnang: *väga kõrge*, kuna kahjustada saavad kõik e-teenused.

— eID infrastruktuuri tõrkekindluse analüüs uuringuaruanne, sealsamas

Seetõttu andis tehtud uiring järgmised soovitused tulevaste uuringute planeerimiseks:

[Soovitus S02] Soovitame analüüsida alternatiivseid usaldusmudeleid ning teadusprojekte kodanike digitaalse identiteedi haldamiseks ning krüptograafilise võtmepaari ja kodaniku identiteedi omavahelise seose tõendamiseks.

[Soovitus S03] Soovitame detailsemalt analüüsida üheainsa CA ebaturvaliseks muutumisega seotud kriisiolukordasid ning sõnastada prioriteedid ning vajadused. Tuleb tähele panna, et näiteks CA dubleerimine ei ole triviaalne ülesanne ning erinevate tehniliste lahenduste kulukust ning omadusi tuleb lähemalt uurida.

— eID infrastruktuuri tõrkekindluse analüüs uuringuaruanne, sealsamas

Riigi Infosüsteemi Amet võttis tehtud soovitused arvesse ja tellis käesoleva uuringu. Uuringu eesmärgiks on uurida avaliku võtme infrastruktuurile alternatiivseid usaldusmudeleid ja pakkuda välja Eestile sobiv usaldusmudel tulevikuks. Usaldusmudelit välja pakkudes kirjeldame, kuidas eID

taristu teenuseid peaks muutma. Oluliseks loeme, et väljapakutav usaldusmuodel töötaks just Eesti kontekstis. Ühilduvus rahvusvaheliste, sealhulgas Euroopa Liidu regulatsioonidega on ka soovitav, kuid väiksema prioriteediga. Samuti loeme oluliseks, et eID taristu peal töötavad teenused toimima jääksid.

## 1.2. Usaldusteenused, -mudelid ja -taristu

Euroopa Parlamendi ja Nõukogu välja antud eIDAS-määrus [2] nimetab ja reguleerib teatud usaldusteenuseid, millest *e-allkirjad* ehk *digisignatuurid* on käesoleva uuringu kontekstis kahtlemata kõige olulisemad. Teine käesoleva uuringu jaoks oluline, kuid eIDAS-es reguleerimata usaldusteenus on *autentimine*.

Autentimises osalevad aktiivselt kaks osapoole — *kasutaja* ja *sõltuv osapool*. Autentimise käigus üritab esimene neist teist veenda, et ta tõepoolest on see isik, kellena ta ennast esitleb. Siin esitlemine tähendab teatud atribuutide kogumiku nimetamist, mida kasutaja väidab endal olevat. Tüüpilisteks atribuutideks on nimi ja isikukood. Veenmise käigus esitab kasutaja sõltuvale osapoolele *sertifikaate*, mis seovad atribuute mingi krüptograafilise materjaliga, tüüpiliselt avalike võtmetega. Samuti tõendab kasutaja, et ta käsutab teatud krüptograafilist materjali, tüüpiliselt neile avalikele võtmetele vastavaid privaatvõtmeid. Selleks, et autentimine õnnestuks, peab sõltuv osapool esitatud sertifikaatidele mingi tähenduse omistama ja neis sertifikaatides esitatud väiteid mingil kombel uskuma. Uskumise ja usaldamise reeglid ja detailid moodustavad *usaldusmuodeli*.

E-allkirjastamisel huvitavad meid eelkõige kaks kasutusjuhtu, kus teine laiendab esimest. Esimesel kasutusjuhul on meil kaks aktiivset kasutajat  $U_1$  ja  $U_2$ , kaks ajahetke  $t_1$  ja  $t_2$  (kus  $t_1$  on varasem kui  $t_2$ ), digitaalne dokument  $D$  ja sellele antud digiallkiri  $\sigma$ . Ajahetkel  $t_2$  tegutsevat kasutajat  $U_2$  huvitab, kas digiallkiri  $\sigma$  dokumendile  $D$  võiks olla antud kasutaja  $U_1$  poolt ajahetkel (umbes)  $t_1$ . Otsustamiseks kasutab  $U_2$  jällegi sertifikaate, mis võivad olla kaasas  $\sigma$ -ga, või mida  $U_2$  ise mingitest registritest küsimusele vastamiseks peab  $U_2$  jällegi sertifikaatides olevat informatsiooni mingil moel usaldama.

E-allkirjastamise teine oluline kasutusjuht toob lisaks kasutaja  $U_3$  ja ajahetke  $t_3$ , mis on hilisem kui  $t_2$ . Kui kasutaja  $U_2$  on ajahetkel  $t_2$  veendunud digiallkirja  $\sigma$  kehtivuses, siis soovib ta lisaks veenduda, kas kasutaja  $U_3$  ajahetkel  $t_3$  sedasama allkirja verifitseerima hakates seda aktsepteeriks. Sellele küsimusele vastamiseks peab  $U_2$  teadma, mida usaldab  $U_3$ . Samuti võib tal olla vaja usaldada, et mingid registrid tulevikus kättesaadavad on ja päringutele õigesti vastavad.

Usaldusteenuste tehnilised realisatsioonid kasutavad sertifikaate, mida väljastavad ja haldavad turutingimustes tegutsevad *usaldusteenuste pakkujad* (TSP-d). TSP-de tegutsemist turul, nende tulemist ja lahkumist toetab riiklik *usaldustaristu*, mis informeerib kasutajaid, millised isikud on TSP-d, milliseid usaldusteenuseid nad pakuvad ja milliste tehniliste kanalite kaudu neid teenuseid on võimalik kasutada. Eesti kontekstis on mõistetel *usaldustaristu* ja *eID taristu* ilmselt suur ühisosa, aga võib kujutada ette ka teenuseid ja funktsionaalsusi, mis on neist kahest täpselt ühe osaks.

## 1.3. Käesoleva aruande sisu

Käesoleva aruande struktuur, selle jaotised ja alamjaotised lähtuvad nii käesoleva uuringu tellinud pakkumiskutsest kui ka hilisematest mõttevahetustest tellijaga. Uuring koosnes kahest suurest

osast: analüüs ja konstruktsioon. Analüüsi käigus omandati ja anti ülevaade senini teaduskirjanduses ja protokolliarenduses väljapakutud usaldusmudelitest, samuti serifitseerimiskeskuste töenäolisematest nõrkustest. Need teadmised olid üheks aluseks Eestile sobiva usaldusmudeli ja -taristu väljapakkumisel. Samuti lähtuti konstruktsiooniosas tellijaga peetud aruteludest, millest selgus, millist sorti süsteemi konstruktsiooni nad saada soovisid. Veel üheks sisendiks konstruktsiooniprotsessi olid täitja senised teadmised Eesti usaldusteenuste turu iseärasustest (s.t. väiksusest).

Käesolev aruanne on koostatud kakskeelsena, peegeldades, kes on meie arvates aruande eri osade tõenäoline lugejaskond. Me usume, et uuringu konstruktsiooniosa tulemused on huvitavad eelkõige Eesti kontekstis, eestikeelsele lugejaskonnale. Konstruktsioon toetub analüüsile, mille järelased võtab lühidalt kokku peatükk 2. Järelduste aluseks on põhjalikumad kirjeldused senitoimunud intsidentidest sertifitseerimiskeskustes ja nendega seotud organisatsionides. Samuti on järelduste aluseks seni teaduskirjanduses ja usaldustaristu disainis väljapakutud usaldusmudelite kirjeldused, mis on esitatud nende omavahelist võrdlemist lihtsustaval viisil. Nii intsidentide kui ka usaldusmudelite kirjelduses on oluline uudsusmoment, mis teeb need huvitavaks ka iseseisvalt, s.t. väljaspool käesolevas uuringuaruandes esitatud konstruktsiooni konteksti. Samuti on uuringu analüüsiosa tulemused ühtmoodi huvitavad lugejaile nii Eestis kui ka kaugemal. Seetõttu oleme nad kirjutanud inglise keeles ning vormistanud nad käesoleva aruande lisadena.

Käesoleva aruande põhiosa algab seega kokkuvõttega uuringu käigus läbiviidud analüüside tulemustest. Sellele järgneb usaldusmudel ja -taristu, mille oleme välja pakkunud arvestades Eesti kontekstiga. Seejuures kirjeldame kõigepealt, milliseid ülesandeid meie väljapakutav süsteem täitma hakkab, ja seejärel esitame süsteemi komponendid, nendevahelised infovoored ja süsteemi osapoolte vahelised usaldussuhted. Infovooge ja usaldussuhteid kirjeldame kõigepealt abstraktsel viisil ja seejärel näitame, kuivõrd ja milliste tehniliste standarditega neid esitada saab. Samuti arutame, kuidas ühildub meie väljapakutud usaldusmudel ja -taristu olemasolevate õiguslike raamistikega Eestis ja kaugemal. Põhjendame oma valikuid analüüsist tehtud järelustega. Aruande (eestikeelses) lisas esitame väljapakutud taristu kvantitatiivse terviklus- ja käideldavusanalüüsi. Õiguslik analüüs toetub raamistikule, mis on esitatud ingliskeelses lisas.

Aruandel on kaks peamist ingliskeelset lisa—sertifitseerimiskeskuste intsidentide loetelu, kirjeldused ja kategoriseerimine ühes allikaviidetega ning varem väljapakutud usaldusmudelite ja neid toetavate tehniliste lahenduste kirjeldused. Teises lisas üritame erinevaid usaldusmudeliteid esitada võimalikult ühtemoodi, kasutades nende kirjeldamiseks ühte ja sama keelt ja mõistestikku. Samuti anname seal usaldusmudelite kvalitatiivse ja kvantitatiivse võrdluse. Me püüame mudeleid võrrelda nii tehniliselt—millised kasutusjuhud on kui hästi toetatud, millised osapooled milliseid teisi kui palju usaldama peavad, kui palju halba põhjustab mõne usaldatava osapoole vale käitumine—kui ka juriidiliselt—kuivõrd sobituvad need mudelid olemasolevatesse õiguslikesse raamistikesse. Kasutataval kirjelduskeelega on formaalne semantika, mille esitame kolmandas ingliskeelses lisas.

## 1.4. Kasutatud metoodika

Tegemist on klassikalise kahest osast—analüüs ja konstruktsioon—koosneva uuringuga. Uuringu analüüsiosa koosnes andmete ja muu teadmuse kogumisest ja süstematiserimisest vastavalt kogutava materjali iseloomule. Sertifitseerimiskeskustega seotud intsidentide juures võisime eeldada, et varasemaid selleteemalisi kokkuvõtteid pigem ei leidu ja seetõttu on materjalide

leidmiseks vaja õigete võtmesõnadega veebiotsinguid. Olles sel viisil leidnud arvestatava koguse intsidente, koguseime kokku nende kirjeldused ja samuti püüdsime leida nende toimumise põhjusi. Põhjuste kirjeldustest leidsime osi, mis korduma kippusid; need osad on meie analüüs is välia toodud. Usaldusmudelite (mille kohta eeldasime, et leidub nii teadusartikleid kui ka reaalsete süsteemide kirjeldusi) süsteematiserimisel soovisime neid võimalikult ühte moodi kirjeldada. Pakkusime välja kaks keelt—pildilise ja intuitiivsema, mida kasutame lisas **B**, ja täpse, modaalloogikal põhineva, mille esitame lisas **E**. Pildilise keele elementide tähendus on antud loogiliste valemitega. Metoodiliselt on huvitav, et esimesena töötasime välja loogikal põhineva keele süntaksi. Selles keeltes eri usaldusmudelite kirjeldades tuvastasime korduvad fraasid, mille võtsime pildilise keele elementideks.

Usaldusmudelite omavaheline kvantitatiivne ja kvalitatiivne võrdlus oli eraldi analüüsietapp. Pakkusime välja viisid, milliste osapoolte millise halva käitumise vastu me kaitsta tahame ja kuidas me kaitstust kvantitatiivselt mõõta soovime. Analüüsime väljapakutud sertifitseerimismudeliteid nende mõõdikute suhtes. See andis meile usaldusmudelite kohta ülevaate, mida kasutada projekti konstruktsioniosas. Uuringu konstruktsioniosaks valmistudes uurisime ka sertifitseerimist reguleerivaid õiguslike raamistikke ja tehnilisi standardeid, et saada aimu, kas need toetavad rohkem üht või teist usaldusmudelit.

Uuringu konstruktsioniosa metoodika oli harjumuspärane: analüüs käigus selgunud võimalikest variantidest valiti välja lootustandvamat ja arendati neid edasi. Tulemusi võrreldi omavahel ja olemasolevate õiguslike ning tehniliste raamistikega, samuti toimusid arutelud tellijaga. Võrdluste ja arutelude põhjal valiti välja meie arvates parim konstruktsioon, mis sobitub olemasolevate raamistikega. Lisaks sellele ühele konstruktsioonile pakume ka välja, millises suunas võiks olemasolevaid raamistikke muuta, selleks et tulevikus veelgi parem usaldustaristu võimalik oleks.

## 1.5. Määratlused ja lühendid

### **ajatempel (timestamp)**

Andmekogum, mis tõestab mingi digitaalse dokumendi või selle osa (näiteks sertifikaadi või digi allkirja) eksisteerimist antud hetkel (vt ka <https://akit.cyber.ee/term/620>).

### **alam-sertifitseerimiskeskus (subCA)**

sertifitseerimiskeskus hierarhilises PKI-s, mis ei ole juur-sertifitseerimiskeskus ja mille õigus sertifikaate väljastada lähtub mõne hierarhias kõrgemal asuva sertifitseerimiskeskuse antud sertifikaadist.

### **autentimine (authentication)**

Tegevus, mille käigus üks olem kontrollib, et teine olem on see, kelleks ta end nimetab (vt ka <https://akit.cyber.ee/term/14>).

### **bridge CA**

vt sild-sertifitseerimiskeskus

### **certificate**

vt sertifikaat

### **(certificate) trust list**

vt usaldusnimmekiri

### **certification authority (CA)**

vt sertifitseerimiskeskus

### **certificate revocation list, CRL**

vt tühistusloend

### **certificate pinning**

vt sertifikaadi kinnistus

### **cross-certification**

vt ristsertifitseerimine

### **detsentraliseeritud identifikaator (decentralized identifier, DID)**

Olemi enda poolt endale valitud identifikaator, mille valikut ei ole teised olemid mõjutanud. Ühel olemil võib olla mitu erinevat DID-i.

### **digisignatuur (digital signature)**

Bitijada, mis krüptograafiliselt seob omavahel mingi digitaalse dokumendi mingi olemiga, keda loeme nõustuvat selle dokumendi sisuga (vt ka <https://akit.cyber.ee/term/579>)

### **distributed CA**

vt hajus sertifitseerimiskeskus

### **distributed ledger**

vt hajusraamat

### **DNSSEC**

*Domain Name System Security Extensions* [3] e. domeeninimesüsteemi turvalahendid (vt ka <https://akit.cyber.ee/term/12517>).

### **EBSI**

*European Blockchain Services Infrastructure*, Euroopa Plokiahela Partnerluse loodav plokiahelataristu.

### **eelsertifikaat (pre-certificate)**

*Certificate Transparency* (CT; vt. <https://akit.cyber.ee/term/14348>) protokollistikus kasutatav andmestruktuur, mis lisatakse CT logidesse enne samade andmetega (päris)sertifikaadi väljastamist, et luua CT-protokollistiku tööks vajalikke andmesõltuvusi.

### **eIDAS**

e-identimise ja e-tehingute jaoks vajalike usaldusteenuste määrus [2].

### **ESSIF**

*European Self-Sovereign Identity Framework*, Suveräänidentiteedi raamistik, mis toetub EBSI-le.

## **GDPR**

*General Data Protection Regulation*, isikuandmete kaitse üldmäärus [4].

## **hajusraamat**

Elektrooniline arvestusraamat (tehinguregister), millest on palju koopiaid, mida peavad üksteisest sõltumatuud haldajad, kes krüptograafilisi meetmeid kasutades püüavad koopiate sisu omavahel võrdsena hoida (vt ka <https://akit.cyber.ee/term/13096>).

## **hajus sertifitseerimiskeskus (*distributed CA*)**

Sertifitseerimiskeskus, mille usaldust nõudvad toimingud on jagatud eri arvutuskeskkondade vahel, mille töö tulemusi on võimalik omavahel mingil määral üksteise vastu võrrelda.

## ***intermediate certificate***

vt vahesertifikaat

## **juur-sertifitseerimiskeskus (*root CA*)**

Sertifitseerimiskeskuste hierarhia tipus olev sertifitseerimiskeskus, mida süsteem kasutab usaldusankruna.

## **KSI**

*Keyless Signature Infrastructure*, Guardtime AS-i pakutav, plokiahelal põhinev teenus digiallkirjade terviklusomaduste parandamiseks.

## **loaline hajusraamat (*permissioned ledger*)**

Hajusraamat, mille halduriks saamiseks on vajalik registreeruda (vt ka <https://akit.cyber.ee/term/13097>).

## **lävi-PKI (*threshold PKI*)**

Lävisignatuure kasutav avaliku võtme taristu, kus sertifikaadid signeeritakse mitme sertifitseerimiskeskuse koostöös.

## ***mesh PKI***

vt sertifitseerimiskeskuste usaldusvõrk

## **MitM**

vt vahendusrünne

## **nõrk lüli (*single point of failure, SPoF*)**

Süsteemi osa, mille tõrge halvab kogu süsteemi (vt ka <https://akit.cyber.ee/term/485>).

## **OCSP**

*Online Certificate Status Protocol*, protokoll, mis võimaldab pärida sertifikaadi staatust reaalajas (vt ka <https://akit.cyber.ee/term/1371>)

## ***permissioned ledger***

vt loaline hajusraamat

## **PKI**

Avaliku võtme taristu (vt ka <https://akit.cyber.ee/term/609>).

### **registreerimiskeskus (*registration authority, RA*)**

Usaldatav osapool, mis teostab isikutuvastust (vt ka <https://akit.cyber.ee/term/1116>)

### ***pre-certificate***

vt eelsertifikaat

### ***relying party***

vt sõltuv osapool

### **ristsertifitseerimine (*cross-certification*)**

Mitut sertifitseerimiskeskust sisaldavas süsteemis ühele sertifitseerimiskeskusele teise poolt sertifikaadi andmine, mis kinnitab, et tegemist on usaldusväärse sertifitseerimiskeskusega (vt ka <https://akit.cyber.ee/term/14273>).

### ***root-CA***

vt juur-sertifitseerimiskeskus

### **sertifikaadi kinnistus (*certificate pinning*)**

Lõppkasutaja sertifikaadi väärtsuse fikseerimine rakenduses või lokaalses süsteemis, mitte selle leidmine usaldustaristu kaudu (vt ka <https://akit.cyber.ee/term/8969>).

### **sertifikaat (*certificate*)**

Digitaalne dokument, mille kaudu mingi olem kinnitab mingi väite (mis reeglina seob mingi olemi mingi avaliku võtmega) kehtivust (vt ka <https://akit.cyber.ee/term/577>).

### **sertifitseerimiskeskus (*certification authority, CA*)**

Usaldatav osapool, mis annab välja ja tühistab sertifikaate kasutaja identiteedi töendamiseks (vt ka <https://akit.cyber.ee/term/560>).

### **sertifitseerimiskeskuste usaldusvõrk (*mesh PKI*)**

avaliku võtme taristu, mis sisaldab mitut sertifitseerimiskeskust, kus sertifitseerimiskeskused muuhulgas ka üksteise usaldusväärssust kontrollivad ja seda kinnitavaid sertifikaate väljastavad.

### **sild-sertifitseerimiskeskus (*bridge CA*)**

Sertifitseerimiskeskus, mis kinnitab eri administratiividomeenide sertifitseerimiskeskuste usaldusväärssust, mis omakorda kinnitavad sild-sertifitseerimiskeskuse usaldusväärssust.

### ***single point of failure (SPoF)***

vt nõrk lüli

### **subCA**

vt alam-sertifitseerimiskeskus

### **suveräänideliteet (*self-sovereign identity*)**

Digitaalidentiteet, mille kasutamise üle on identiteedi omanikul oluline kontroll (vt ka

<https://akit.cyber.ee/term/14645>).

### **sõltuv osapool**

Sertifikaadi saaja, kes tegutseb toetudes sellele sertifikaadile ja/või sellega kontrollitavale digitaalsignatuurile (vt ka <https://akit.cyber.ee/term/1118>).

### **threshold PKI**

vt lävi-PKI

### **timestamp**

vt ajatempel

### **trust list**

vt usaldusnimikiri

### **trust model**

vt usaldusmudel

### **trust service**

vt usaldusteenus

### **tühistusloend (*certificate revocation list, CRL*)**

Usaldusväärne nimikiri sertifikaatidest, mis on tühistatud (vt ka <https://akit.cyber.ee/term/816>).

### **usaldusankur (*trust anchor*)**

Usaldatav osapool, mille usaldatavas lähtub väljastpoolt süsteemi, mitte ei kerki (*emerge*) süsteemsete tegevuste tulemusel. Vaata ka <https://akit.cyber.ee/term/1102> (teist tüüpi objekt, aga sama roll).

### **usaldusmudel (*trust model*)**

Üldised põhimõtted, mis panevad paika selle, millised olemid süsteemis peavad milliseid teisi olemeid milliste väidete osas usaldama (vt ka <https://akit.cyber.ee/term/13857>).

### **usaldusnimikiri (*trust list*)**

Euroopa Liidu ja Euroopa Majandusühenduse liikmesriikide avaldatud nimikiri kvalifitseeritud usaldusteenuste pakkujatest vastavalt eIDAS-ele.

### **usaldusteenus (*trust service*)**

Üks eIDAS-es loetletud teenustest nagu näiteks sertifikaatide väljastamine, digiallkirjastamine või ajatempel. Võib olla kvalifitseeritud või mitte-kvalifitseeritud.

### **usaldusteenuse pakkuja (*trust service provider, TSP*)**

Osapool, kes osutab ühte või mitut usaldusteenust.

### **usaldusvõrk (*web of trust*)**

Usaldusmudel, kus avalike võtmeid isikutega siduvaid sertifikaate vahetavad need isikud reeglina otse omavahel (vt ka <https://akit.cyber.ee/term/3837>).

### **vahendusrünne (*man-in-the-middle attack*)**

Krüptoprotokolli vastane rünne, kus ründaja muudab teateid, mis liiguvad ausate osapoolte vahel (vt ka <https://akit.cyber.ee/term/923>).

### **vahesertifikaat (*intermediate certificate*)**

sertifikaatide ahelas asuv sertifikaat, mis ei ole juur- ega lõppataseme sertifikaat (vt ka <https://akit.cyber.ee/term/8718>).

### **valideerimisautoriteet (*validation authority*)**

Usaldusankur, kes kinnitab, et sertifitseerimiskeskuste väljaantud sertifikaate võib usaldada.

### **verifitseeritav lubatähht (*verifiable credential*)**

Digitaalselt allkirjastatud dokument, mis seob süsteemisisese identifikaatori (näiteks detsentraliseeritud identifikaatori) mingi atribuudiga (näiteks vanus) või loaga (näiteks juhiluba), mis seda identifikaatorit kontrollival isikul on.

### **verifitseeritav identiteet (*verifiable ID*)**

Verifitseeritav lubatähht, mis seob süsteemisisese identifikaatori (näiteks detsentraliseeritud identifikaatori) mingi süsteemivälise identifikaatoriga (näiteks pärisnimega).

## **1.6. Viited**

[1] “eID infrastruktuuri törkekindluse analüüs.” Uuringuaruanne, Riigi Infosüsteemi amet ja Cybernetica AS (dok. nr. Y-1362-1), Jun. 2020.

[2] European Parliament and Council of European Union, “Regulation (EU) no 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.” OJ L 257, 28.8.2014, p. 73-114, 2014.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “DNS Security Introduction and Requirements, RFC 4033.” Internet Engineering Task Force (IETF), 2005.

[4] European Parliament and Council of European Union, “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance).” OJ L 119, 4.5.2016, p. 1-88, 2016.

[5] R. Zuccherato, P. Cain, D. C. Adams, and D. Pinkas, “Internet X.509 Public Key Infrastructure Time-Signature Protocol (TSP),” no. 3161. RFC Editor, Aug. 2001, doi: 10.17487/RFC3161.

[6] S. Haber and W. S. Stornetta, “How to Time-Stamp a Digital Document,” *J. Cryptol.*, vol. 3, no. 2, pp. 99–111, 1991, doi: 10.1007/BF00196791.

[7] Riigi Infosüsteemi amet, “Web eID.” 2022, [Online]. Available: <https://www.id.ee/artikkel/web-eid/>.

[8] European Commission, “Digital Signature Service.” 2022, [Online]. Available:

<https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Digital+Signature+Service++DSS>.

[9] "Elutähta teenuse kirjeldus ja toimepidevuse nõuded elektroonilise isikutuvastamise ja digitaalse allkirjastamise tagamisel. Ministri määruse eelnõu. 26.10.2018." Eelnõu toimiku number: 18-1102.

[10] "Euroopa parlamendi ja nõukogu direktiiv 1999/93/EÜ, 13. detsember 1999, elektroonilisi allkirju käsitleva ühenduse raamistiku kohta." Euroopa Liidu Teataja L 013 , 19/01/2000 Lk 0012 - 0020.

[11] "Isikuttõendavate dokumentide seadus." Eesti Vabariik, Riigikogu, RT I 1999, 25, 365 ... RT I, 15.10.2021, 1.

[12] "Hädaolukorra seadus." Eesti Vabariik, Riigikogu, RT I, 03.03.2017, 1 ... RT I, 18.06.2021, 1.

[13] "Elutähta teenuse kirjeldus ja toimepidevuse nõuded elektroonilise isikutuvastamise ja digitaalse allkirjastamise tagamisel. Ettevõtlus- ja tehnoloogiaministri 11.01.2019 määrus nr 4." RT I, 15.01.2019, 11.

[14] K. Laanest and L. Kask, "ID-kaardi turvarisk. Õiguslikud probleemid." Riigi Infosüsteemi amet, 2017.

[15] A. Paršovs, "Solving the Estonian ID Card Crisis: the Legal Issues," 2020.

[16] Microsoft, "Microsoft Security Bulletin MS01-017 - Critical Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard." <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2001/ms01-017> , Mar. 22, 2001.

[17] R. Lemos, "Microsoft warns of hijacked certificates." <https://www.cnet.com/news/microsoft-warns-of-hijacked-certificates/> , Jan. 02, 2002.

[18] I. Ristic, *Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications*. Feisty Duck, 2013.

[19] J. McCormick, "Look out for fraudulent Microsoft digital certificates." <https://www.techrepublic.com/article/look-out-for-fraudulent-microsoft-digital-certificates/> , Apr. 09, 2001.

[20] M. Zusman, "Criminal Charges Not Pursued Hacking PKI." [https://defcon.org/images/defcon-17/dc-17-presentations/defcon-17-zusman-hacking\\_pki.pdf](https://defcon.org/images/defcon-17/dc-17-presentations/defcon-17-zusman-hacking_pki.pdf) , Jul. 31, 2009.

[21] M. Zusman, "Domain Validated SSL Certificates." <http://schmoil.blogspot.com/2008/08/domain-validated-ssl-certificates.html> .

[22] M. Zusman, "DNS vuln + SSL cert = FAIL." <https://web.archive.org/web/20100809095612/http://intrepidusgroup.com/insight/2008/07/dns-vuln-ssl-cert-fail/> .

[23] StartSSL, "Critical incident report - 20th december 2008." <https://web.archive.org/web/20110615091751/https://blog.startcom.org/wp-content/uploads/2009/01/critical-event-report-12-20-2008.pdf> , Dec. 20, 2008.

- [24] E. Nigg, “Full Disclosure.” <https://web.archive.org/web/20110615080143/https://blog.startcom.org/?p=161>, Jan. 03, 2009.
- [25] E. Nigg, “Untrusted Certificates.” <http://web.archive.org/web/20120312061436/https://blog.startcom.org/?p=145>, Dec. 23, 2008.
- [26] E. Nigg, “Unbelievable!” <https://groups.google.com/g/mozilla.dev.tech.crypto/c/nAzIKSBEh78?pli=1>, Dec. 23, 2008.
- [27] L. Seltzer, “SSL Certificate Vendor Sells Mozilla.com Cert to Some Guy.” <http://web.archive.org/web/20120312061436/https://blog.startcom.org/?p=145>, Dec. 24, 2008.
- [28] SSLShopper, “SSL Certificate for Mozilla.com Issued Without Validation.” <https://www.sslshopper.com/article-ssl-certificate-for-mozilla.com-issued-without-validation.html>, Dec. 23, 2008.
- [29] VeriSign, Inc., “Form 10-Q, quarterly report.” <https://investor.verisign.com/sec-filings/sec-filing/10-q/0001193125-11-285850>, Sep. 30, 2011.
- [30] Comodo, “Comodo fraud incident report.” <https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html>, Mar. 2011.
- [31] ComodoHacker, “A message from Comodo Hacker.” <https://pastebin.com/74KXCaEZ>, Mar. 26, 2011.
- [32] E. Nigg, “Cyber War.” <https://web.archive.org/web/20120402115209/https://blog.startcom.org/?p=229>, Sep. 09, 2011.
- [33] D. Goodin, “Web authentication authority suffers security breach.” [https://www.theregister.com/2011/06/21/startssl\\_security\\_breach/](https://www.theregister.com/2011/06/21/startssl_security_breach/), Jun. 21, 2011.
- [34] ComodoHacker, “Another status update message.” <https://pastebin.com/85WV10EL>, Sep. 06, 2011.
- [35] M. J. Schwartz, “How StartCom Foiled Comodohacker: 4 Lessons.” <https://www.darkreading.com/attacks-breaches/how-startcom-foiled-comodohacker-4-lessons>, Sep. 08, 2011.
- [36] J. Prins, “Diginotar certificate authority breach, Diginotar certificate authority breach.” <https://www.enisa.europa.eu/media/news-items/operation-black-tulip>, Sep. 2011.
- [37] H. Hoogstraaten, “Black Tulip, Report of the investigation into the DigiNotar Certificate Authority breach.” Aug. 13, 2012, doi: 10.13140/2.1.2456.7364.
- [38] J. Nightingale, “Fraudulent \*.google.com Certificate.” <https://blog.mozilla.org/security/2011/08/29/fraudulent-google-com-certificate/>, Aug. 29, 2011.
- [39] Microsoft Security Response Center, “Microsoft Releases Security Advisory 2607712.” <https://msrc-blog.microsoft.com/2011/08/29/microsoft-releases-security-advisory-2607712/>, Aug. 29, 2011.

- [40] Y. N. Pettersen, “DigiNotar First Step: Disabling the Root.” <https://web.archive.org/web/2011111203522/http://my.opera.com/rootstore/blog/2011/09/06/diginotar-first-step-disabling-the-root>, Aug. 29, 2011.
- [41] Fox-IT, “DigiNotar public report version 1.” <https://web.archive.org/web/20150919015849/https://www.rijksoverheid.nl/ministeries/ministerie-van-binnenlandse-zaken-en-koninkrijksrelaties/documenten/rapporten/2011/09/05/diginotar-public-report-version-1>, Sep. 05, 2011.
- [42] ComodoHacker, “Striking Back...” <https://pastebin.com/1AxH30em>, Sep. 05, 2011.
- [43] GlobalSign GMO Internet Group, “Incident Response.” <https://web.archive.org/web/20111011180316/http://www.globalsign.co.uk/company/press/090611-security-response.html>, Sep. 06, 2011.
- [44] BBC News, “GlobalSign stops secure certificates after hack claim.” <https://www.bbc.com/news/technology-14819257>, Sep. 07, 2011.
- [45] GlobalSign GMO Internet Group, “Security Incident Report.” <https://web.archive.org/web/20161022215737/https://www.globalsign.com/en/resources/globalsign-security-incident-report.pdf>, Dec. 13, 2011.
- [46] C. Wisniewski, “Another certificate authority issues dangerous certificates.” <https://nakedsecurity.sophos.com/2011/11/03/another-certificate-authority-issues-dangerous-certificates/>, Nov. 03, 2011.
- [47] G. Markham, “Entrust SubCA: 512-bit key issuance and other CPS violations; malware in the wild.” <https://nakedsecurity.sophos.com/2011/11/03/another-certificate-authority-issues-dangerous-certificates/>, Nov. 01, 2011.
- [48] E. Barker and A. Roginsky, “Transitioning the use of cryptographic algorithms and key lengths,” National Institute of Standards and Technology, Mar. 2019. doi: 10.6028/nist.sp.800-131ar2.
- [49] A. K. Lenstra, “Key Lengths Contribution to The Handbook of Information Security,” 2010.
- [50] Entrust, “Entrust Bulletin on Certificates Issued with Weak 512-bit RSA Keys by Dicert Malaysia.” <https://web.archive.org/web/2011123014747/http://www.entrust.net/advisories/malaysia.htm>, Nov. 2011.
- [51] J. Nightingale, “Revoking Trust in DigiCert Sdn. Bhd Intermediate Certificate Authority.” <https://blog.mozilla.org/security/2011/11/03/revoking-trust-in-digicert-sdn-bhd-intermediate-certificate-authority>, Nov. 03, 2011.
- [52] DigiCert, Inc, “DigiCert, Inc. Of No Relation to Recent ‘Digi’ Insecure Certificates.” <https://web.archive.org/web/20111206222135/https://www.digicert.com/news/2011-11-1-breaches-and-similar-names.htm>, Nov. 01, 2011.
- [53] Trustwave SpiderLabs, “Clarifying The Trustwave CA Policy Update.” <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/clarifying-the-trustwave-ca-policy-update/>, Feb. 04, 2012.
- [54] J. Jarmoc, “SSL/TLS interception proxies and transitive trust,” *Black Hat Europe*, 2012.

- [55] S. Wiesinger, “Remove Trustwave Certificate(s) from trusted root certificates.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=724929](https://bugzilla.mozilla.org/show_bug.cgi?id=724929), Feb. 07, 2012.
- [56] K. Wilson, “Mozilla Communication: Action requested by March 2, 2012.” <https://groups.google.com/g/mozilla.dev.security.policy/c/6CX23NVaUvY>, Feb. 18, 2012.
- [57] J. Nightingale, “Message to Certificate Authorities about Subordinate CAs.” <https://blog.mozilla.org/security/2012/02/17/message-to-certificate-authorities-about-subordinate-cas/>, Feb. 17, 2012.
- [58] R. A. Sandvik, “Security vulnerability found in Cyberoam DPI devices (CVE-2012-3372).” <https://blog.torproject.org/security-vulnerability-found-cyberoam-dpi-devices-cve-2012-3372/>, Jul. 03, 2012.
- [59] R. A. Sandvik and B. Laurie, “Vulnerability in Cyberoam DPI devices.” <https://media.torproject.org/misc/2012-07-03-cyberoam-CVE-2012-3372.txt>, Jul. 30, 2012.
- [60] Cyberoam, “Cyberoam’s Proactive Steps in HTTPS Deep Scan Inspection.” <https://web.archive.org/web/20130301160605/http://blog.cyberoam.com/2012/07/cyberoam's-proactive-steps-in-https-deep-scan-inspection>, Jul. 09, 2012.
- [61] N. Willis, “Cyberoam deep packet inspection and certificates.” <https://lwn.net/Articles/506337/>, Jul. 11, 2012.
- [62] A. Langley, “Enhancing digital certificate security.” <https://security.googleblog.com/2013/01/enhancing-digital-certificate-security.html>, Jan. 03, 2013.
- [63] TURKTRUST, “Technical Details.” <https://web.archive.org/web/20130926134541/http://turktrust.com.tr/en/kamuoyu-aciklamasi-en.2.html>, Jan. 07, 2013.
- [64] P. Ducklin, “The TURKTRUST SSL certificate fiasco – what really happened, and what happens next?” <https://nakedsecurity.sophos.com/2013/01/08/the-turktrust-ssl-certificate-fiasco-what-happened-and-what-happens-next/>, Jan. 08, 2013.
- [65] Bugzilla, “Deal with TURKTRUST mis-issued \*.google.com certificate.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=825022](https://bugzilla.mozilla.org/show_bug.cgi?id=825022), Dec. 27, 2012.
- [66] Mozilla, “Revoking Trust in Two TurkTrust Certificates.” <https://blog.mozilla.org/security/2013/01/03/revoking-trust-in-two-turktrust-certificates/>, Jan. 03, 2013.
- [67] Microsoft, “Microsoft Security Advisory 2798897, Fraudulent Digital Certificates Could Allow Spoofing.” <https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2013/2798897?redirectedfrom=MSDN>, Jan. 03, 2013.
- [68] ANSSI, “IGC/A.” <https://www.ssi.gouv.fr/administration/services-securises/igca/>, Jan. 21, 2014.
- [69] A. Langley, “Further improving digital certificate security.” <https://security.googleblog.com/2013/12/further-improving-digital-certificate.html>, Dec. 07, 2013.
- [70] ANSSI, “Revocation of an IGC/A branch.” <https://web.archive.org/web/20140214235559/http://www.ssi.gouv.fr/en/the-anssi/events/revocation-of-an-igc-a-branch-808.html>, Dec. 07, 2013.

- [71] Microsoft, “Microsoft Security Advisory 2916652, Improperly Issued Digital Certificates Could Allow Spoofing.” <https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2014/2916652?redirectedfrom=MSDN> , Dec. 09, 2013.
- [72] Opera Team, “Certificate update.” <https://blogs.opera.com/security/2013/12/certificate-update/> , Dec. 09, 2013.
- [73] K. Wilson, “Revoking Trust in one ANSSI Certificate.” <https://blog.mozilla.org/security/2013/12/09/revoking-trust-in-one-anssi-certificate/> , Dec. 09, 2013.
- [74] A. Langley, “Maintaining digital certificate security.” <https://security.googleblog.com/2014/07/maintaining-digital-certificate-security.html> , Jul. 08, 2014.
- [75] TechNet, “Windows Root Certificate Program Members (October 2011).” <https://web.archive.org/web/20140708213211/http://social.technet.microsoft.com/wiki/contents/articles/5225.windows-root-certificate-program-members-october-2011.aspx> , Oct. 25, 2011.
- [76] Microsoft, “Microsoft Security Advisory 2982792, Improperly Issued Digital Certificates Could Allow Spoofing.” <https://docs.microsoft.com/en-us/security-updates/securityadvisories/2014/2982792> , Jul. 10, 2014.
- [77] Ramachandran, “Add CCA India root certificate.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=557167](https://bugzilla.mozilla.org/show_bug.cgi?id=557167) , Apr. 04, 2010.
- [78] O. Vänskä, “Suomalainen paljasti Microsoftin haavoittuvuuden: yhtiö kiitti sulkemalla sähköpostitilin.” <https://www.tivi.fi/uutiset/suomalainen-paljasti-microsoftin-haavoittuvuuden-yhtio-kiitti-sulkemalla-sahkopostitilin/f68bccb9-92ad-37d5-a105-2d70cf6b25f6> , Mar. 17, 2015.
- [79] D. Goodin, “Man who obtained Windows Live cert said his warnings went unanswered.” <https://arstechnica.com/information-technology/2015/03/man-who-obtained-windows-live-cert-said-his-warnings-went-unanswered/> , Mar. 17, 2015.
- [80] Comodo, “Alternative Methods of Domain Control Validation (DCV).” <https://web.archive.org/web/20160310065826/https://support.comodo.com/index.php?/Default/Knowledgebase/Article/View/791/16/> , 2015.
- [81] Microsoft, “Microsoft Security Advisory 3046310, Improperly Issued Digital Certificates Could Allow Spoofing.” <https://docs.microsoft.com/en-us/security-updates/securityadvisories/2015/3046310> , Mar. 16, 2015.
- [82] A. Langley, “Maintaining digital certificate security.” <https://security.googleblog.com/2015/03/maintaining-digital-certificate-security.html> , Mar. 23, 2015.
- [83] CNNIC, “Clarification on some media’s claim that ‘CNNIC has issued certificates for MITM attack.’” [https://web.archive.org/web/20150430071513/http://www1.cnnic.cn/AU/MediaC/Announcement/201503/t20150325\\_52019.htm](https://web.archive.org/web/20150430071513/http://www1.cnnic.cn/AU/MediaC/Announcement/201503/t20150325_52019.htm) , Mar. 25, 2015.
- [84] Mozilla, “The MCS Incident and Its Consequences for CNNIC.” <https://blog.mozilla.org/security/files/2015/04/CNNIC-MCS.pdf> , Apr. 2015.
- [85] Mozilla, “Revoking Trust in one CNNIC Intermediate Certificate.” <https://blog.mozilla.org/>

[security/2015/03/23/revoking-trust-in-one-cnnic-intermediate-certificate/](http://security/2015/03/23/revoking-trust-in-one-cnnic-intermediate-certificate/), Mar. 23, 2015.

[86] Microsoft, “Microsoft Security Advisory 3050995, Improperly Issued Digital Certificates Could Allow Spoofing.” <https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/3050995?redirectedfrom=MSDN>, Mar. 24, 2015.

[87] S. Somogyi and A. Eijdenberg, “Improved Digital Certificate Security.” <https://security.googleblog.com/2015/09/improved-digital-certificate-security.html>, Sep. 18, 2015.

[88] Certificate Transparency, “Working together to detect maliciously or mistakenly issued certificates.” <https://certificate.transparency.dev/> .

[89] Q. Liu and C. Mike-Billstrom, “A Tough Day as Leaders.” <https://archive.md/Ro70U> , Sep. 18, 2015.

[90] R. Sleevi, “Sustaining Digital Certificate Security.” <https://security.googleblog.com/2015/09/improved-digital-certificate-security.html>, Oct. 28, 2015.

[91] Symantec, “Symantec Trust Services Incident Report 1.” <https://web.archive.org/web/20151015023348/https://www-secure.symantec.com/connect/sites/default/files/TestCertificateIncidentReportOwnedDomains.pdf>, Oct. 12, 2015.

[92] Symantec, "Symantec Trust Services Incident Report 2." <https://web.archive.org/web/20151103084431/https://www-secure.symantec.com/connect/sites/default/files/TestCertificateIncidentReportUnregisteredv2.pdf>, Oct. 12, 2015.

[93] Opera Team, “Misissued certificates.” <https://blogs.opera.com/security/2015/10/misissued-certificates/>, Oct. 29, 2015.

[94] K. Wilson, “Symantec Test Cert Misissuance Incident.” [https://groups.google.com/g/mozilla.dev.security.policy/c/Hkyg\\_09EDYE/m/izCdFR7wBQAJ](https://groups.google.com/g/mozilla.dev.security.policy/c/Hkyg_09EDYE/m/izCdFR7wBQAJ), Oct. 13, 2015.

[95] SK ID Solutions, "Neljal tuhandel dokumendil tuleb uuendada eesti.ee meiliaadressi." <https://www.skidsolutions.eu/uudised/neljal-tuhandel-dokumendil-tuleb-uuendada-eestieemeiliaadressi>, Sep. 01, 2015.

[96] A. Paršovs, “Estonian electronic identity card and its security challenges,” PhD thesis, University of Tartu, 2021.

[97] K. Wilson, “Distrusting New WoSign and StartCom Certificates.” <https://blog.mozilla.org/security/2016/10/24/distrusting-new-wosign-and-startcom-certificates/>, Oct. 24, 2016.

[98] A. Whalley, “Distrusting WoSign and StartCom Certificates.” <https://security.googleblog.com/2016/10/distrusting-wosign-and-startcom.html>, Oct. 31, 2016.

[99] Apple Inc., “Blocking Trust for WoSign CA Free SSL Certificate G2.” <https://support.apple.com/en-us/HT204132>, Dec. 18, 2018.

[100] Microsoft Defender Security Research Team, “Microsoft to remove WoSign and StartCom certificates in Windows 10.” <https://www.microsoft.com/security/blog/2017/08/08/microsoft-to-remove-wosign-and-startcom-certificates-in-windows-10/>, Aug. 08, 2017.

- [101] G. Markham, “Incidents involving the CA WoSign.” <https://www.mail-archive.com/dev-security-policy@lists.mozilla.org/msg03665.html> , Aug. 24, 2016.
- [102] Mozilla Wiki, “CA:WoSign Issues.” [https://wiki.mozilla.org/CA:WoSign\\_Issues](https://wiki.mozilla.org/CA:WoSign_Issues) .
- [103] WoSign CA Limited, “WoSign Incidents Final Report.” [https://www.wosign.com/report/WoSign\\_Incident\\_Final\\_Report\\_09162016.pdf](https://www.wosign.com/report/WoSign_Incident_Final_Report_09162016.pdf) , Sep. 16, 2016.
- [104] Ernst and Young, “Report of Independent Accountants.” <https://web.archive.org/web/20160528234724/https://cert.webtrust.org/SealFile?seal=2019&file=pdf> , Mar. 31, 2016.
- [105] C. Palmer and R. Sleevi, “Gradually sunsetting SHA-1.” <https://security.googleblog.com/2014/09/gradually-sunsetting-sha-1.html> , Sep. 05, 2014.
- [106] R. Barnes, “Continuing to Phase Out SHA-1 Certificates.” <https://blog.mozilla.org/security/2015/10/20/continuing-to-phase-out-sha-1-certificates/> , Oct. 20, 2015.
- [107] CA/Browser Forum, “Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates.” <https://cabforum.org/wp-content/uploads/CA-Forum-BR-1.3.3.pdf> , Feb. 04, 2016.
- [108] S. Bradner, “Key words for use in RFCs to Indicate Requirement Levels.” <https://datatracker.ietf.org/doc/html/rfc2119> , Mar. 1999.
- [109] WoSign, “WoSign Certificates Policy And Practice Statement.” <https://www.wosign.com/policy/wosign-policy-1-2-11.pdf> , Apr. 07, 2015.
- [110] G. Markham, “WoSign’s Ownership of StartCom.” [https://groups.google.com/g/mozilla.dev.security.policy/c/0pqpLJ\\_lCJQ](https://groups.google.com/g/mozilla.dev.security.policy/c/0pqpLJ_lCJQ) , Sep. 09, 2015.
- [111] Mozilla Wiki, “CA:Problematic Practices.” [https://web.archive.org/web/20150904072836/https://wiki.mozilla.org/CA:Problematic\\_Practices](https://web.archive.org/web/20150904072836/https://wiki.mozilla.org/CA:Problematic_Practices) .
- [112] WoSign, “WoSign Incidents Report.” <https://www.mail-archive.com/dev-security-policy@lists.mozilla.org/msg03665.html> , Sep. 04, 2016.
- [113] A. Ayer, “Domain Validation Vulnerability in Symantec Certificate Authority.” [https://www.agwa.name/blog/post/domain\\_validation\\_vulnerability\\_in\\_symantec\\_ca](https://www.agwa.name/blog/post/domain_validation_vulnerability_in_symantec_ca) , Feb. 05, 2016.
- [114] CA/Browser Forum, “Baseline Requirements Documents (SSL/TLS Server Certificates).” <https://cabforum.org/baseline-requirements-documents/> .
- [115] Broadcom, “Security Updates Detail.” [https://www.broadcom.com/support/security-center/securityupdates/detail?fid=security\\_advisory&pvid=security\\_advisory&suid=20160204\\_00&year=](https://www.broadcom.com/support/security-center/securityupdates/detail?fid=security_advisory&pvid=security_advisory&suid=20160204_00&year=) , Feb. 04, 2016.
- [116] M. Bryant, “Keeping Positive – Obtaining Arbitrary Wildcard SSL Certificates from Comodo via Dangling Markup Injection.” <https://thehackerblog.com/keeping-positive-obtaining-arbitrary-wildcard-ssl-certificates-from-comodo-via-dangling-markup-injection/index.html> , Jul. 25, 2016.
- [117] R. Alden, “Incident Report - OCR.” <https://www.mail-archive.com/dev-security>

[policy@lists.mozilla.org/msg04654.html](mailto:policy@lists.mozilla.org/msg04654.html) , Oct. 19, 2016.

[118] R. Eikenberg, “Zertifikats-Klau: Fatale Sehschwäche bei Comodo.” <https://www.heise.de/security/meldung/Zertifikats-Klau-Fatale-Sehschwaechen-bei-Comodo-3354229.html> , Oct. 19, 2016.

[119] steffen, “Comodo: CA Comodo used broken OCR and issued certificates to the wrong people.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1311713](https://bugzilla.mozilla.org/show_bug.cgi?id=1311713) , Oct. 20, 2016.

[120] W. Thayer, “Incident Report – Certificates issued without proper domain validation.” <https://groups.google.com/g/mozilla.dev.security.policy/c/Htujoyq-pO8> , Jan. 11, 2017.

[121] W. Thayer, “Information about SSL bug.” <https://web.archive.org/web/20170112004624/https://www.godaddy.com/garage/godaddy/information-about-ssl-bug/> , Jan. 10, 2017.

[122] A. Ayer, “Misissued/Suspicious Symantec Certificates.” <https://www.mail-archive.com/dev-security-policy@lists.mozilla.org/msg05455.html> , Jan. 19, 2017.

[123] rsl...@chromium.org, “Intent to Deprecate and Remove: Trust in existing Symantec-issued Certificates.” <https://groups.google.com/a/chromium.org/g/blink-dev/c/eUAKwjhhBs/m/El1mH8S6AwAJ> , Mar. 23, 2017.

[124] Mozilla Wiki, “CA:Symantec Issues.” [https://wiki.mozilla.org/CA:Symantec\\_Issues](https://wiki.mozilla.org/CA:Symantec_Issues) .

[125] K. Wilson, “Symantec: Mis-issued test certificates by CrossCert.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1334377](https://bugzilla.mozilla.org/show_bug.cgi?id=1334377) , Jan. 26, 2017.

[126] Deloitte, “Independent Accountants’s Report.” <https://bug1334377.bmoattachments.org/attachment.cgi?id=8831930> , Sep. 09, 2016.

[127] D. O’Brien, R. Sleevi, and A. Whalley, “<https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html>.” <https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html> , Sep. 11, 2017.

[128] K. Wilson, “Distrust of Symantec TLS Certificates .” <https://blog.mozilla.org/security/2018/03/12/distrust-symantec-tls-certificates/> , Mar. 12, 2018.

[129] A. Ayer, “Certinomis: certificates for an unregistered domain, with unknown OCSP status.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1544933](https://bugzilla.mozilla.org/show_bug.cgi?id=1544933) , Apr. 16, 2019.

[130] OpenData .fr TLD, “Data from the .fr TLD to serve innovation.” <https://web.archive.org/web/20171109024523/https://opendata.afnic.fr/en/products-and-services/services/opendata-en.html> , Sep. 2017.

[131] Mozilla Wiki, “CA/Certinomis Issues.” [https://wiki.mozilla.org/CA/Certinomis\\_Issues](https://wiki.mozilla.org/CA/Certinomis_Issues) .

[132] A. Arampatzis, “Mozilla Distrusts Certinomis Issued Certificates.” <https://www.venafi.com/blog.mozilla-distrusts-certinomis-issued-certificates> , Jul. 16, 2019.

[133] W. Thayer, “Certinomis/Docapost: Non-BR-Compliant OCSP Responders.” [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1425998](https://bugzilla.mozilla.org/show_bug.cgi?id=1425998) , Dec. 18, 2017.

[134] W. Thayer, “Remove Certinomis - Root CA.” [https://bugzilla.mozilla.org/show\\_bug.cgi?](https://bugzilla.mozilla.org/show_bug.cgi?)

[id=1552374](#), May 16, 2019.

- [135] R. S. Raman, L. Evdokimov, E. Wustrow, A. Halderman, and R. Ensafi, “Kazakhstan’s HTTPS Interception.” <https://censoredplanet.org/kazakhstan> , Jul. 23, 2019.
- [136] D. Warburton, “Kazakhstan Attempts to MITM Its Citizens.” <https://www.f5.com/labs/articles/threat-intelligence/kazakhstan-attempts-to-mitm-itscitizens> , Aug. 01, 2019.
- [137] W. Thayer, “Protecting our Users in Kazakhstan.” <https://blog.mozilla.org/security/2019/08/21/protecting-our-users-in-kazakhstan/> , Aug. 21, 2019.
- [138] A. Whalley, “Protecting Chrome users in Kazakhstan.” <https://security.googleblog.com/2019/08/protecting-chrome-users-in-kazakhstan.html> , Aug. 21, 2019.
- [139] ImperialViolet, “Public key pinning.” <https://www.imperialviolet.org/2011/05/04/pinning.html> , May 04, 2011.
- [140] CA/Browser Forum, “Welcome to the CA/Browser Forum.” <https://cabforum.org/> .
- [141] CEF eDelivery, “CEF eDelivery Building Block v1.2. Trust Models Guidance.” May 2018.
- [142] H. Rifà-Pous and J. Herrera-Joancomartí, “An Interdomain PKI Model Based on Trust Lists,” in *Public Key Infrastructure, 4th European PKI Workshop: Theory and Practice, EuroPKI 2007, Palma de Mallorca, Spain, June 28-30, 2007, Proceedings* , 2007, vol. 4582, pp. 49–64, doi: 10.1007/978-3-540-73408-6\_4.
- [143] J. Ølnes, “PKI interoperability by an independent, trusted validation authority,” in *5 th Annual PKI R&D Workshop "Making PKI Easy to Use" Proceedings*, Apr. 2006, pp. 68–78.
- [144] A. D. Santis, Y. Desmedt, Y. Frankel, and M. Yung, “How to share a function securely,” in *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada* , 1994, pp. 522–533, doi: 10.1145/195058.195405.
- [145] D. F. Aranha, A. P. K. Dalskov, D. Escudero, and C. Orlandi, “Improved Threshold Signatures, Proactive Secret Sharing, and Input Certification from LSS Isomorphisms,” in *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings* , 2021, vol. 12912, pp. 382–404, doi: 10.1007/978-3-030-88238-9\_19.
- [146] A. Buldas, A. Kroonmaa, and R. Laanoja, “Keyless Signatures’ Infrastructure: How to Build Global Distributed Hash-Trees,” in *Secure IT Systems - 18th Nordic Conference, NordSec 2013, Ilulissat, Greenland, October 18-21, 2013, Proceedings* , 2013, vol. 8208, pp. 313–320, doi: 10.1007/978-3-642-41488-6\_21.
- [147] A. Buldas, D. Firsov, R. Laanoja, H. Lakk, and A. Truu, “A New Approach to Constructing Digital Signature Schemes (Extended Paper),” *IACR Cryptol. ePrint Arch.*, p. 673, 2019, [Online]. Available: <https://eprint.iacr.org/2019/673>.
- [148] A. Buldas, R. Laanoja, and A. Truu, “Keyless signature infrastructure and PKI: hash-tree signatures in pre- and post-quantum world,” *Int. J. Serv. Technol. Manag.*, vol. 23, no. 1/2, pp. 117–130, 2017, doi: 10.1504/IJSTM.2017.10002708.

- [149] B. Kulynych, W. Lueks, M. Isaakidis, G. Danezis, and C. Troncoso, “ClaimChain: Improving the Security and Privacy of In-band Key Distribution for Messaging,” in *Proceedings of the 2018 Workshop on Privacy in the Electronic Society, WPES@CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, 2018, pp. 86–103, doi: 10.1145/3267323.3268947.
- [150] European Blockchain Partnership, “ESSIF Documentation.” 2021, [Online]. Available: <https://ec.europa.eu/digital-building-blocks/wikis/display/EBSIDOC/ESSIF+Functional+Scope>.
- [151] European Blockchain Partnership, “EBSI Documentation.” 2021, [Online]. Available: <https://ec.europa.eu/digital-building-blocks/wikis/display/EBSIDOC/General>.
- [152] European Blockchain Partnership, “ESSIF Documentation.” 2021, [Online]. Available: <https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=379913698>.
- [153] European Blockchain Partnership, “Business Scenario: Natural Person Onboards On ESSIF (ESSIF-BS-01).” 2022, [Online]. Available: <https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=379913702>.
- [154] European Blockchain Partnership, “Business Scenario: A Natural Person Requests Verifiable ID from a Legal Entity (ESSIF-BS-03).” 2021, [Online]. Available: <https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=379913704>.
- [155] T. J. Smedinghoff, “Laws Governing Identity Systems (v2),” *IDPro Body of Knowledge*, vol. 1, no. 5, 2021.
- [156] “Electronic Identity Management Act.” Code of Virginia, Title 59.1, Chapter 50.
- [157] “Draft Provisions on the Use and Cross-border Recognition of Identity Management and Trust Services.” United Nations Commission on International Trade Law, Working Group IV (Electronic Commerce), Sixty-second session, Vienna, 22–26 November 2021.
- [158] “E-identimise ja e-tehingute usaldusteenuste seadus.” Eesti Vabariik, Riigikogu, RT I, 25.10.2016, 1 ... RT I, 15.10.2021, 1.
- [159] “Riigi Infosüsteemi Ameti põhimääerus.” Eesti Vabariik, Majandus- ja kommunikatsiooniminister, RT I, 28.04.2011, 1 ... RT I, 28.09.2021, 2.
- [160] “eIDAS Compliant eID Solutions.” European Union Agency for Cybersecurity (ENISA) report, Mar. 2020.
- [161] “Digital Identity. Draft Guidelines.” CONSULTATIVE COMMITTEE OF THE CONVENTION FOR THE PROTECTION OF INDIVIDUALS WITH REGARD TO AUTOMATIC PROCESSING OF PERSONAL DATA, CONVENTION 108, Council of Europe, Directorate General of Human Rights and Rule of Law, 20 September 2021, T-PD-BUR(2021)2rev2.
- [162] “Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity.” European Commission, Brussels, 3.6.2021, COM(2021) 281 final, 2021/0136(COD).
- [163] “Isikuandmete kaitse seadus.” Eesti Vabariik, Riigikogu, RT I, 04.01.2019, 11.

[164] “Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data.” Council of Europe, European Treaty Series - No. 108, Strasbourg, 28.I.1981.

[165] “Küberturvalisuse seadus.” Eesti Vabariik, Riigikogu, RT I, 22.05.2018, 1.

[166] “Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for a high common level of security of network and information systems across the Union.” European Union, OJ L 194, 19.7.2016, p. 1–30.

[167] “Directive 2013/40/EU of the European Parliament and of the Council of 12 August 2013 on attacks against information systems and replacing Council Framework Decision 2005/222/JHA.” European Union, OJ L 218, 14.8.2013, p. 8–14.

[168] “Convention on Cybercrime.” Council of Europe, European Treaty Series - No. 185, Budapest, 23.XI.2001.

[169] “Avaliku teabe seadus.” Eesti Vabariik, Riigikogu, RT I 2000, 92, 597 ... RT I, 15.03.2019, 2.

[170] “Regulation (EU) 2018/1724 of the European Parliament and of the Council of 2 October 2018 establishing a single digital gateway to provide access to information, to procedures and to assistance and problem-solving services and amending Regulation (EU) No 1024/2012 (Text with EEA relevance.) PE/41/2018/REV/2.” European Union, OJ L 295, 21.11.2018, p. 1–38.

[171] A. Herzig *et al.*, “Prolegomena for a Logic of Trust and Reputation,” in *Third International Workshop on Normative Multiagent Systems - NorMAS 2008, Luxembourg, July 15-16, 2008. Proceedings*, 2008, pp. 143–157.

[172] A. Herzig, E. Lorini, J. F. Hübner, and L. Vercouter, “A logic of trust and reputation,” *Log. J. IGPL*, vol. 18, no. 1, pp. 214–244, 2010, doi: 10.1093/jigpal/jzp077.

[173] C.-J. Liau, “Belief, information acquisition, and trust in multi-agent systems—A modal logic formulation,” *Artificial Intelligence*, vol. 149, no. 1, pp. 31–60, 2003, doi: 10.1016/S0004-3702(03)00063-8.

## 2. Järeldused analüüsist

Käesoleva uuringu käigus tehtud intsidentide ja usaldusmudelite analüüsi eesmärk on leida mustreid, millest oleks kasu Eestile sobiva usaldusmudeli ja seda toetava tehnilise raamistikku väljapakkumisel ja analüüsил. Sertifitseerimiskeskustega seotud intsidente kirjeldame ja rühmitame lisas A. Neist kirjeldustest joonistuvad välja järgmised mustrid.

- Intsidentid puudutavad ründeid terviklusele. Tõepoolest, kõik lisas A kirjeldatud juhtumid kujutavad endast mingit tervikluskaudu. Siit järel dame, et käideldavusintsidentid kas ei ole olnud olulised, või ununevad need veel kiiremini kui terviklusintsidentid (mille kohta ammendava või vähemalt põhjaliku materjali leidmisega me samuti hädas olime).
- Suur osa intsidentitest on tingitud sellest, et sertifitseerimisteenuse pakkuja millegipäras ei suutnud tuvastada, et see agent, kellega ta suhtleb ja kes temalt avalikule võtmele sertifikaati küsib, ei ole õigustatud esinema nime all, mida ta sertifikaadile saada tahab. Seetõttu andis teenusepakkija välja sertifikaadi, millel olevat võtit sertifikaadil oleva nimega isik tegelikult ei kontrollinud.
- Teine osa intsidentitest on tingitud sellest, et sertifitseerimisteenuse pakkuja infosüsteemidesse hakiiti sisse. Võib ütelda, et peale sellist sündmust teenusepakkija enam ei kontrollinud avalikku võtit, millega avalikkus teda seostas.
- Kolmas osa intsidentitest oli tingitud vale tüüpi sertifikaatide väljastamise tõttu, vastuolus sertifikaatide väljastamise reeglitega, mis rakendusid sertifitseerimisteenuse pakkujale läbi nende sisemise poliitika või teiste isikutega sõlmitud lepingute. Vale tüüp võib tähendada näiteks sertifikaati, mis lubab edasisertifitseerimist.

Seega tuleks meil Eestile sobivat usaldusteenust välja pakkudes pöörata tähelepanu esmajärjekorras neile kolmele võimalikule nõrkusele. Arvestame, et sertifikaate võib olla tarvis tühistada; see kehtib nii lõppkasutajatele kui ka sertifitseerimiskeskustele antud sertifikaatide kohta. Samuti püüame eri tüüpi sertifikaadid, kui neid peaks vaja olema, teha üksteisest hästi eristatavaks ja anname täpsse seletuse neist igaühe semantikale. Saadava usaldustaristu käideldavus on loomulikult midagi, mille poole püüelda, kuid lühikesed ja harvad katkestused võivad olla vastuvõetavad.

Usaldusmudelite ülevaade lisas B annab meile komponendid, millest Eestile sobiv taristu kokku panna. Üritades eri mudelitest teha suure üldistusega kokkuvõtet, märkame järgmisi detaile, mis korduvad ühest mudelist teise ja/või mida omavahel kombineerides saame koostada usaldustaristu.

1. Selleks, et üks agent teaks teise agendi avalikku võtit mingil ajal, on tarvis, et keegi ütleks, mis see võti oli. Ütleja leidmiseks peab jällegi keegi ütlema, kes on sobiv ütleja. Lõpuks peab leiduma mingi usaldusankur, kelle ülemisi me usaldame aksiomaatiliselt, s.t. see usaldus ei järeldu selle agendi või seda agenti puudutavatest tegemistest süsteemi sees. See võib järelduda seda agenti puudutavatest tegemistest väljaspool süsteemi, näiteks selle agendi auditeerimisest. Me võiksime auditeerimisprotseduurid lisada oma süsteemile, aga sellega me ainult nihutaksime kohta, kus toimub aksiomaatiline usaldamine.
2. Usaldusankur saab olla hajus. Usaldus võib olla jagatud nii konjunktiivselt kui disjunktiivselt, s.t. avaliku võtme kohta käiva väite uskumiseks nõuame me, et mingi hulk usaldusankruid (süsteemis on spetsifitseeritud, millised hulgad sobivad) seda väidaks.

3. Iga agenti, on ta siis süsteemi lõppkasutaja või mingi sertifitseerimiskeskus, usaldatatakse ainult mingi väidete klassi suhtes. Neisse klassidesse mittekuuluvad väited jäetakse tähelepanuta. Mingi agendi jaoks võib see väidete klass, mille suhtes teda usaldatatakse, sisaldada väiteid teiste agentide usaldamise kohta.
4. Avalike võtmete kohta käivad väited peavad kusagilt loetavad olema; seda on vajalik süsteemi käideldavuse jaoks. See, et mingi väide on kusagilt loetav, on jällegi väide, mille vastu on tarvis tekitada usaldus.
5. See, et mingit agenti mingite väidete klassi suhtes usaldatatakse, ei tähenda seda, et ta ei võiks korraga, erinevatele kuulajatele, esitada üksteisele vastukäivaid väiteid sellest klassist. Vastukäivate väidete puudumine on jällegi väide, mille vastu tuleb tekitada usaldus.
6. Nii loetavuse kui ka vastuolude vältimise jaoks on olemas tehnilised meetmed, tüüpiliselt mingite avalike registrite (plokiahelate) näol. Seega on võimalik nende väidete usaldatavus tagada tehniliste meetmetega. See aga jällegi suurendab süsteemi kulusid.

Ühes usaldusmudelite kirjeldamisega me ka võrdlesime omavahel eri usaldusmudeleid ja neid realiseerivaid taristuid. Me hindasime, kuidas nad sobituvad mõnede töenäolistele, autentimise või signeerimisega seotud stsenaariumitega. Samuti püüdsime võrrelda nende ülesseadmise ja kasutamise kulusid. Me leidsime, et:

- Kui soovime säilitada klassikalise avaliku võtme taristuga seotud usaldusmudelit, siis on käideldavust võimalik tõsta, kui lisada rohkem sertifitseerimiskeskusi. Nad kõik saaksid välja anda sertifikaate, nii et süsteem jätkaks tööd isegi siis, kui mõni sertifitseerimiskeskus töö lõpetab. Selle lähenemise halb külg on, et terviklusomaduste rikkumiseks ja valesertifikaatide väljastamiseks piisab ühestainsast pahatahtlikust sertifitseerimiskeskusest.
- Kui kasutada sertifitseerimiskeskuste usaldusvõrku ilma läviturvalisuseteta, siis peavad sertifitseerimiskeskused üksteist mingil viisil ristsertifitseerima. Kui mõni sertifitseerimiskeskus muutub pahatahtlikuks, siis loodetavasti teised keskused märkavad seda muutust. Selles mudelis usaldaks kasutaja ikkagi ühtainsat sertifitseerimiskeskust, mis oleks nõrk lüli sarnaselt klassikalise PKI-ga. Me võiksime ka süsteemi üles seada nii, et kasutaja usaldab mitut sertifitseerimiskeskust ja need ristsertifitseerivad üksteist, nii et kui ühes neist mingi tõrge tekib, siis teisi on võimalik ikkagi edasi usaldada. Meil tuleks siis täpsemalt uurida ja spetsifitseerida, mida ristsertifitseerimine tähendab ja kuivõrd me sellele loota saame.
- Läviturvalisusega sertifitseerimiskeskuste usaldusvõrk suurendab nii süsteemi käideldavust (kõigi sertifitseerimiskeskuste töötamine pole vajalik sertifikaadi välja andmiseks) kui ka terviklust (üksainus pahatahtlik sertifitseerimiskeskus ei saa valesertifikaate välja anda). Eelmiste lähenemistega võrreldes vajab läviturvalisusega taristu muudatusi sertifikaatide formaadis, et lävisignatuurid toetatud oleksid. Juhime tähelepanu ka sellele, et me hetkel ei oma head ettekujutust, kuidas efektiivselt moodustada postkvantturvalisi lävisignatuure.
- Usaldusvõrku on keeruline realiseerida ja samuti ei paista ta kõiki meie vajadusi rahuldavat, sealhulgas kasutaja identiteedi ja füüsilise identiteedi omavahel sidumist. Kui me fikseerime, et leiduvad mingid konkreetsed osapooled, kes verifitseerivad uute kasutajate identiteete, siis tekkiv usaldusmudel ei ole väga erinev tsentraliseeritud mudelist. Arvestusraamatu realiseerimine vajaks mingit turvalise ühisarvutuse protokolli, mille postkvantturvaliseks tegemine on mittetrviaalne. Avalik arvestusraamat nagu näiteks EBSI (sellel baseerub ESSIF) on mugav selles suhtes, et igasugune vastutus privaatsuse ja kättesaadavuse osas on delegeritav EBSI omanikele. Sellisel juhul tuleb aga jällegi veenduda selles, et nad on ka

tegelikult usaldatavad — kui midagi juhtuma peaks, siis ei piisa meile ainult sellest, et meil on keegi, keda süüdistada.

Läviturvalisuse juures tuleb hoolitseda selle eest, et erinevad sertifitseerimiskeskused ikka tõepoolest sõltumatud oleksid. Kui nad kõik kasutavad sama tarkvara ja/või neil on sama turvapolitiika ja/või nad asuvad samas hoones, siis on võimalik, et nõrkus, mille kaudu üks neist ära petetakse või pahatahtlikuks muudetakse, on olemas ka teistel. Selle peale tuleb mõtelda ka sotsiaalsete rünnete kontekstis, kus ründaja üritab saada sertifikaati mingile nimele, millele tal "õigust" ei ole. Kui sertifitseerimiskeskused annavad välja riiklikult tunnustatud sertifikaate, siis on tõenäoline, et kusgil süsteemis on mingi riiklik vahendaja (näiteks PPA), kes sertifikaate tellib ja neid elanike ID-kaartidele lisab. Nii võib PPA osutuda nõrgaks lülik. Kui iga sertifitseerimiskeskus oleks sertifikaatide väljastamisel teistest sõltumatu, siis peaks elanik oma identiteeti neist igaühele tööstama. See oleks sarnane sellega, kui oma dokumentide uuendamiseks tuleks külastada mitut erinevat politseijaoskonda; selline bürokraatia tundub olevat vastuvõtmatu. Sarnased probleemid on olemas ka SSI juures: verifitseeritava lubatähe saaja esitleb ennast ühele kindlale lubatähtede väljastajale, kes osutub siis selle süsteemi nõrgaks lülik.

# 3. Soovituslik eID usaldusmuodel

## 3.1. Ülesandepüstitus

Tellijaga toimunud mõttevahetuste tulemusel on käesoleva aruande autorid jõudnud arusaamisele, et välja tuleb pakkuda komponendid, liidesed ja toimimispõhimõtted (sealhulgas usalduspõhimõtted) süsteemile, millega usalduseteenuse pakkujad (TSP-d) saavad liituda ja Eesti kasutajatele usalduseenuseid pakkuda. Samuti võivad TSP-d süsteemist lahkuda; süsteemi ülesandeks on tagada, et usaldusseosed, mille nad on kasutajate vahel tekitanud, seejuures säiliksid.

Süsteem, mille toimimispõhimõtted järgnevas välja pakume, peaks suutma töödelda järgnevaid sündmusi:

- TSP liitub süsteemiga.
- TSP asub pakkuma teatud usaldusteenust. Me ei pööra järgnevas tähelepanu kõigile eIDAS-es nimetatud usaldusteenustele, vaid ainult neile, mida peame käesoleva uuringu kontekstis olulisemaks: digiallkirjastamine ning (isiku)sertifikaatide väljastamine ja haldamine.
- TSP lõpetab teatud usaldusteenuse pakkumise.
- TSP lahkub süsteemist. Eeldame, et kõik TSP väljastatud sertifikaadid kaotavad kehtivuse alates TSP lahkumise hetkest.
- TSP lahkub ootamatult umbusalduse töttu.
- Kasutaja saab endale sertifikaadi süsteemis olevalt TSP-lt.
- Kasutaja digiallkirjastab dokumendi, kusjuures allkirjastamiseks kasutatav võti on seotud süsteemis oleva TSP väljastatud sertifikaadiga.
- Kasutaja kontrollib sertifikaati või digiallkirja.
- Sertifikaat tühistatakse (väljaandja poolt).

Elektroonilise ID süsteemide põhilised kasutusjuhud kujutavad endast nende sündmuste järjendeid: TSP-d liituvad ja annavad kasutajatele sertifikaate. Kasutajad autendivad end teistele kasutajatele (sõltuvatele osapooltele; RP-dele) ja/või digiallkirjastavad dokumente. Kasutajad soovivad otsustada, kas autentimisi ja digiallkirju aktsepteerida või mitte. Digiallkirjade aktsepteerimise juures võib oluline olla ka see, kas otsust tegeva kasutaja arvates mõni teine kasutaja seda allkirja samuti tulevikus aktsepteerib. Kõigi nende tegevuste käigus võib juhtuda, et mõni TSP lahkub süsteemist. Põhilised kasutusjuhud tooma välja lisas [B.5](#), kus me ka arutame, kuidas nad sobituvad olemasolevate usaldusmuodelitega.

Käesolevas peatükis väljapakutava süsteemi terviklusomadused peaksid tagama kasutusjuhtude terviklusnõueteharuldatuse. Digiallkirjastamise põhilise terviklusnõue on, et üks kasutaja ei saaks teise eest digiallkirju anda. Formaalsemalt: (kolmas) kasutaja ei tohiks aktsepteerida digiallkirjastatud dokumenti mõnelt teiselt kasutajalt tulnuna, kui see teine kasutaja seda dokumenti allkirjastada ei soovinud. Autentimise terviklusnõue on sarnane: üks kasutaja ei tohiks saada esineda teise kasutajana.

Kasutusjuhtude käideldavusnõuded nõuavad, et ühel kasutajal õnnestuks enda tuvastamine teisele

kasutajale ja et korrektelt loodud digiallkirju aktsepteeritaks praegu ja tulevikus. Uskumine, et teatud digiallkirja aktsepteeritakse ka tulevikus, toetub usaldusele, et meie väljapakutav süsteem ka tulevikus töötab; konkreetsed väited, mida üks või teine osapool uskuma peab, on oluline osa süsteemi usalduspõhimõtetest.

## 3.2. Arhitektuurivalikute arutelu

Selles jaotises esitame mõttekäigud, mis viivad konkreetse arhitektuuri pakkumiseni jaotises 3.3. Võtame veel kord kokku eID süsteemide nõrkused, ründed ja võimalikud vastumeetmed. Meie kaalutlused viivad meid lävi-PKI eelistamiseni. Lisas B kirjeldatud mudelid käsitlevad olukorda, kus süsteemis olevate sertifikaatide väljastamise teenuse pakkujate hulk on staatiline. See ei vasta jaotises 3.1 toodud ülesandepüstitusele; dünaamiliselt ilmuvate ja kaduvate teenusepakkujate ja läviturvalisuse omavaheline sidumine tekib täiendavaid takistusi, mille ületamisviisi valiku kaalutlus samuti selles jaotises tutvustame.

### 3.2.1. Võimalikud nõrgad lülid eID süsteemides

Süsteemi kirjelduse lihtsustamiseks eeldame, et sertifikaatide väljastamise teenust pakuva TSP struktuuris on olemas järgmised osad:

- Sertifitseerimiskeskus (CA) haldab (isiku)sertifikaatide väljastamise teenust. See osa TSP-st signeerib sertifikaate ja haldab signeerimiseks kasutatavaid privaatvõtmeid.
- Registreerimiskeskus (RA) viib kokku isiku identiteedi ja isiku elektroonilise identifitseerimise vahendi avalikud võtmed. Ta pakub seega (TSP-sisest) teenust, mille tarbija(d) on CA(d).

Käesolevas peatükis me kirjeldame arhitektuuri, millesse paigutuvad sertifikaatide väljastamise teenust pakkuvad TSP-d. Arhitektuur puudutab eelkõige CA-de toimimist ja seetõttu me allpool sageli spetsifitseerimegi, et "CA teeb midagi", mõeldes selle all muidugi seda, et seda teeb TSP sertifikaatide väljastamise teenuse osutamise käigus.

Soovitusliku usaldusmudeli eesmärk on lahti saada *nõrgast lülist*. Lisas A.2 võtame kokku erinevad asjaolud, mis on põhjustanud TSP-de ja nende osaks olevate CA-dega seotud intsidente. Nõrga lüli rollis võivad olla järgmised komponendid:

- **Sertifitseerimiskeskus.** Ründaja võib CA süsteemi sisse tungida ning panna selle väljastama võltssertifikaate. Alternatiivselt võib ründaja leida privaatvõtme, mida CA sertifikaatide signeerimiseks kasutab. Viimast saab teha ka ilma CA-d otseselt ründamata, näiteks kui krüptograafilise võtme genereerimine on nõrk.
- **Registreerimiskeskus.** Ründaja võib RA süsteemi sisse tungida. Lisaks on võimalik võltssertifikaat saada ka suhtlusründe abil, kui ründaja suudab veenda RA-d, et ta esindab ettevõtet, mille esindusõigust tal tegelikult ei ole.

Nõrgaks lülikks võib osutuda ka osapool, kes vastutab kasutajate salajaste võtmete genereerimise eest. Näiteks ID-kaardi võtmete genereerimise osas peab usaldama kiibitootjat ning sertifitseerimiskeskus lihtsalt seob need võtmed kasutajate identiteediga. Kiibitootja võib olla nõrk lüli; seetõttu vajame me alternatiivlahendusi nagu Mobiil-ID või Smart-ID, mis sõltuvad erinevatest kiibitootjatest. Tegemist pole CA-dega seotud intsidendiga ning käesolevas arhitektuuris me seda probleemi ei käsite.

Üks ja sama TSP võib pakkuda sertifikaatide väljastamise teenust eri viisidel, tüüpiliselt eri sihtrühmadele, ja seetõttu võib tema osaks olla mitu erinevat CA-d ja RA-d, läbi mille pakutakse ja hallatakse mingeid konkreetseid variante sellest teenusest. Siin peatükis teeme me lihtsustuse, et iga TSP osaks on täpselt üks CA, millega on seotud täpselt üks RA. See teeb konstruktsiooni ja selle analüüslihtsamaks ega kitsenda üldisust järgmistel põhjustel.

- Juhul, kui ühel TSP-l on mitu CA-d, peame neid ühiseks nõrgaks lüliksi. Meie usaldusmudelis peaksime neid vaatlema kui ühtainsat CA-d.
- Juhul, kui üks ja sama RA on seotud mitme erineva CA-ga, on tegemist ikkagi ühe nõrga lüliga ning selliste CA-de kogum oleks sama nõrk kui üksik CA.

Kuna CA-d ja RA-d on nüüd omavahel üksüheses vastavuses, siis loeme allpool, et CA täidab ka RA ülesandeid ning RA-dele me allpool eraldi rolli ette ei näe. Seega ei hakka me eraldi RA komponenti välja tooma ning räägime lihtsalt CA-dest, nii et CA ründamise alla lähevad nii CA kui ka RA rikkumisega seotud ründed, muuhulgas ka inimvead ja sotsiaalsed ründed.

Rikutud CA võib mõjutada nii süsteemi käideldavust (ei väljasta uusi sertifikaate, ei vasta OCSP päringutele) kui ka terviklust (väljastab vältssertifikaate, vastab valesti OCSP päringutele). Meie eesmärk on maandada mölemad riskid.

### 3.2.2. Nõrkade lülide võimalikud mõjud ja nende leevendamine

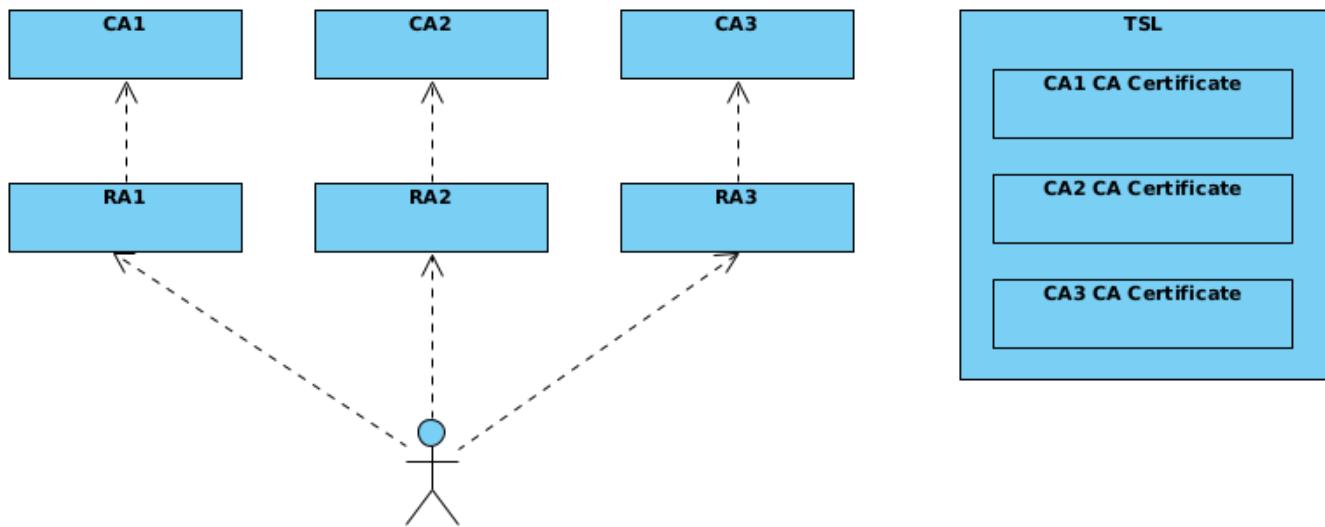
Arutame järgnevalt, mida kahjulikku üks nõrk lüli võib teha ja kuidas saaks kahjusid vältida või vähendada.

- **Ründed käideldavuse vastu.** CA võib keelduda uue sertifikaadi väljastamisest või vana tühistamisest. Samuti võib ta OCSP-päringutele mitte vastata. Need ründed on leevendatud süsteemides, kus sertifikaat peab olema kinnitatud *kas ühe või teise CA poolt*. Kui primaarne CA ei vasta, siis võime kasutada alternatiivset CA-d senikaua, kuni primaarne CA jälle päringutele vastama hakkab.
- **Ründed tervikluse vastu.** CA võib väljastada sertifikaate, mida ta väljastada ei tohiks. Samuti võib ta OCSP-päringutele valesid vastuseid anda (s.t. nimetada mittekehitivat sertifikaati kehtivaks). Need ründed on leevendatud süsteemides, kus sertifikaat peab olema kinnitatud *nii ühe kui ka teise CA poolt*. Siis üks CA ei saa valetada ilma teise CA nõusolekuta.

Pakutud leevendusmeetmed on mõningal määral vastuolulised, sest alternatiivsete CA-de arvu suurendamine käideldavusrünnete vähendamiseks omakorda lihtsustab ründeid tervikluse vastu. Üks võimalik universaallahendus on lävi-PKI (see usaldusmudel on kirjeldatud jaotises B.4.2), kus sertifikaadi väljastamiseks peaks sellega nõus olema piisavalt palju, kuid mitte liiga palju CA-sid. Jaotises 3.3 pakumegi välja minimaalse CA-de arvuga lävi-PKI lahenduse, kus süsteemis on kokku 3 aktiivset CA-d, millest teenuse osutamiseks (sertifikaadi väljastamiseks, tühistamiseks jne) peavad omavahel olema nõus vähemalt 2 CA-d. Teisisõnu, me teeme ettepaneku kasutusele võtta (3,2)-lävi-PKI. Sellisel juhul peab ründaja saama enda kontrolli alla vähemalt 2 erinevat CA-d nii käideldavuse kui ka tervikluse ründeks. Piirdudes oma pakkumises kahe CA-ga, oleksime pidanud valima kas käideldavuse või tervikluse vahel.

### 3.2.3. Lävi-PKI kasutuselevõtt ja sellega seotud kitsaskohad

Kõige lihtsam viis (3,2)-lävi-PKI kasutusele võtmiseks on kolme erineva TSP loomine ja ülalpidamine riigi poolt. Need TSP-d pakuksid sertifikaatide väljastamise teenust. See võib aga olla riigi jaoks liiga kallis. Uute TSP-de loomise asemel võiks kasutada olemasolevate teenusepakkujate abi, sõlmides lepingud näiteks *EU TSLS usaldusteenuse nimekirjas* elevate teenusepakkujatega.



Enda loodud TSP-dega võrreldes on olemasoleva TSP poolt hallatava ja pakutava sertifikaatide väljastamise teenusega seotud alltoodud kitsaskohad, mis võiksid siiski kõik leevedatavad olla.

#### 3.2.3.1. Stabiilsus

Teenusepakkujad võivad tulla ja lahkuda. Seega peaks usaldus nende vastu olema dünaamiline ning üks TSP peaks olema teisega lihtsasti asendatav. Kasutusel peab olema usaldusnimekiri, milles info TSP-de kohta uueneb pidevalt. Seda nimekirja peab kusgil hoidma ja see peab olema uuendatav; seda ei ole võimalik staatilisena salvestada kasutajate seadmetesse. Usaldusnimekirja haldaja on potentsiaalne nõrk lüli.

Kitsaskoha leevedamiseks eeldame, et usaldusnimekirja haldaja on usaldusväärne asutus, kes nimekirja sisu pahatahtlikult ei muuda. Lisaks saab RP alati kontrollida, kas nimekirjas loetletud TSP on ka *EU TSLS* liige. (3,2)-lävi-PKI tagab süsteemi toimimise ka juhul, kui üks kolmest seotud TSP-st lahkub, mis annab ajavaru vastava sertifikaadi uuendamiseks.

#### 3.2.3.2. Usaldusväärus

Üks oluline eeldus (3,2)-lävi-PKI kasutamiseks on see, et ründaja kontrolli all võib olla ülimalt üks kolmikusse kuuluv TSP. See võib olla praktikas liiga tugev eeldus.

Kitsaskoha leevedamiseks kohendame (3,2)-lävi-PKI lahenduse niimoodi, et see töötaks veidi nõrgematel eeldustel. Lubame, et rikutud võivad olla mitu TSP-d. Selle eest eeldame, et nad ei saa olla korraga *ühe ja sama ründaja kontrolli all*, ehk siis ei saa teha omavahel koostööd sertifikaatide vältsimise osas.

#### 3.2.3.3. Paindlikkus

*EU TSLS* liikme kontrollitud TSP on valinud tehnoloogiad, mida ta usaldusteenuse pakkumiseks

kasutab. Üldjuhul ei tohiks ta neist kõrvale kalduda, sest muidu ei pruugi ta enam olla vastavuses eIDAS-e regulatsiooniga. Näiteks, ei saa me eeldada, et sertifikaatide väljastamise teenust pakkova TSP kasutatav tehnoloogia lubab sertifikaatide signeerimiseks kasutatavat privaatvõtit turvaliselt jagada mitme osapoole vahel.

TSP-de poolt kasutatavate tehnoloogiate mitmekesisuse haldamiseks pakume allpool välja sellise arhitektuuri, kus neid tehnoloogiaid saab kasutada "musta kastina"; seda isegi siis, kui need olulisel määral erinevad klassikalisest PKI-st (näiteks, ESSIF). See tähendab, et meie arhitektuuripakkumine ei nõua võimalust luua üks sertifikaat mitme TSP koostöös. Pakume hoopis välja mitme erineva sertifikaadi kombineerimise eID süsteemi kasutusuhtudes. Peame aga arvestama, et olemasolevad autentimis- ja signeerimisrakendused kontrollivad hetkel vaid ühte sertifikaati. Jaotises 3.4 arutame seda probleemi detailse malt.

## 3.3. Pakutud arhitektuur

Selles jaotises kirjeldame lahenduse tehnilist poolt. Me ei kirjelda korralduslikku osa, näiteks seda, milliste kriteeriumite järgi eID taristut haldav riigiasutus otsustab, kas CA on piisavalt usaldusväärne, kas tema sertifikaadid peaks tühistama jne.

### 3.3.1. Osapooled

- **Olem** on keegi, kellele saab välja anda sertifikaati. Ta võib olla füüsiline või juriidiline isik.
- **TSP (trust service provider)** on usaldusteenuse pakkuja (tarnija).
- **CA (certificate authority)** on TSP osa — sertifitseerimiskeskus. Eeldame, et iga TSP osaks on täpselt üks CA.
- **AUX (auxiliary)** on abiteenus, mis haldab usaldusnimekirja ja lubab teha sellele teatud pärtinguid.
- **OS (onboarding service)** on usaldatav osapool, mis teeb otsuseid TSP-le teenuse osutamise loa andmise või ärvõtmise kohta.
- **RP (relying party)** on osapool, mis nõuab teenuse osutamiseks teatud tüüpi sertifikaati.

### 3.3.2. TSP-de eri usaldustasemed

Eeldus, et kaks CA-d (ja neid sisaldavat TSP-d) ei saa olla ühe ja sama ründaja kontrolli all, võib olla liiga optimistlik, sõltuvalt näiteks sellest, millised teenusepakkujad võivad alluda ühele ja samale jurisdiktsionile. Seega klassifitseeritakse meie arhitektuuriettepanekus TSP-sid "rohkem" ja "vähem" usaldusväärseteks. Usaldusväärsemast TSP-st eeldame, et ta ei tee koostööd teiste TSP-dega sertifikaatide vältsimise osas. Praktikas tähendaks see, et kui ründajal õnnestub saada sellise TSP osaks olevalt CA-lt konkreetset nime ja avalikku võtit siduv völtsseertifikaat, siis ei saa ta enam täpselt samale nimele ja samale avalikule võtmele völtsseertifikaati mõne teise TSP osaks olevalt CA-lt. Nimetame "rohkem" ja "vähem" usaldusväärseid TSP-sid vastavalt *I taseme* ja *II taseme* TSP-ks.

Kuidas otsustada, millist TSP-d usaldada I taseme ja millist TSP-d II taseme väärset? Usaldus selles mõttes, et usume TSP esitatud väiteid isikute ja avalike võtmete seoste kohta, lähtub usaldusest kui kontrollist TSP üle. See kontroll võib tähendada, et TSP ja tema osade protsessid ja (info)süsteemid

on mingil viisil auditeeritud. Samuti võib see kontroll tähendada, et meil on mingisugune jurisdiktsioon selle TSP üle. Käesolevas aruandes ei anna me konkreetseid juhiseid, kuidas otsustada, milline kontroll on piisav selleks, et usaldada TSP-d I või II taseme väärselt — selliste auditijuhiste koostamine oleks käesoleva uuringuprojekti skoobist väga kaugele jäänud.

Loomulik on ehk eeldada, et kõik TSP-d on mingisugused auditeerimised läbinud ning I ja II taseme TSP-de jaoks ei ole meil erinevaid auditinõudeid. Sellisel juhul võib *jurisdiktsioon* olla see aspekt, mis TSP-de usaldusväärsest mõjutab. Võime näiteks lugeda kõik Euroopa Liidu usaldusnimekirja kuuluvad teenusepakkujad I taseme TSP-deks. Sellisesse nimekirja mittekuuluvad TSP-d loeme II taseme TSP-deks. Kombineerime neist lähtuvat usaldust niimoodi, et II taseme esindaja saaks tugevdada I taseme esindajaid, kuid poleks samas ise nõrk lüli.

Kui otsustame valida usaldusmudeli aluseks (3,2)-lävi-PKI, siis eeldame, et iga võtmepaariga on esialgu seotud kolm sertifikaati, kuid autentimiseks või signatuuri verifitseerimiseks piisab kahest. Seejuures on oluline, et üks esitatud sertifikaatidest oleks väljastatud I taseme TSP osaks oleva CA poolt. Kui mõlemad esitatud sertifikaadid tuleksid II taseme TSP-de poolt, siis on suurem oht, et need on mõlemad vältstitud.

Ideaalselt peaks sertifikaatide kolmikus olema vähemalt kaks sertifikaati, mille on väljastanud I taseme TSP ja ülimalt üks sertifikaat, mille on väljastanud II taseme TSP. I taseme TSP-de arv peab olema II taseme TSP-de arvust rangelt suurem järgmistel põhjustel:

- Süsteem peab toimima ka siis, kui üks nende TSP-de osaks olevatest CA-dest hetkel ei tööta.
- II taseme TSP-d ei tohiks rikkuda terviklust isegi koostöös.

Sellisel juhul on esitatud kahest sertifikaadist vähemalt üks väljastatud I taseme TSP poolt. Selline süsteem toimiks ilma tõrgeteta ka ühe I taseme TSP riknemisel.

### **3.3.3. Usaldusteenuse pakkujate andmebaas**

Meie arhitektuur näeb ette, et eksisteerib riiklikul tasemel usaldatav abiteenus AUX, mis haldab usaldusteenuse pakkujate andmeid. Eeldame, et teenuse halduv ei hakka muutma selle sisu pahatahtlikult, nii et andmete terviklus on garanteeritud. Käideldavuse tagamiseks võime eeldada, et andmetest tehakse piisavalt tihti varukoopiaid. Mõlemaid omadusi on ilmselt võimalik tagada ka hajusraamatu tehnoloogiate abil.

AUX teenus hoiab iga usaldusteenuse pakkuja kohta unikaalset identifaatorit, teenuse tüüpi, usaldustaset ja teenuse olekut. Samuti hoiab ta muudatuste ajalugu.

Teenuse olek saab siirduda aktiivsest tühistatuks, aga mitte kunagi vastupidi. Korra ära võetud luba enam ei tagastata. Kui selliseid olukordi peaks ette tulema, siis käsiteeme seda nii, nagu TSP oleks sama teenusega uuesti turule tulnud.

Usaldusteenuse pakkujate andmetele kehtivad järgmised juurdepääsuõigused.

- Lugemine on kõigile lubatud.
- Ridade lisamine on lubatud ainult riiklikule registreerimisteenusele OS.
- Ridade eemaldamine pole üldse lubatud — vanemad kirjad säilitatakse hilisemaks auditeerimiseks.

Järgmises jaotises selgitame põhjalikumalt, kuidas eri osapooled hakkavad käituma sõltuvalt AUX teenuses olevast andmeseisust.

AUX teenus on sisuliselt realiseeritud täna kui Euroopa Liidu XML-formaadis signeeritud usaldusnimekiri.

### 3.3.4. Usaldusmuodel

Kasutajale teenuse osutamiseks peab sõltuv osapool uskuma järgmisi väiteid.

1. Kõik kirjad, mis OS lisab usaldusteenuse pakkujate teenusesse, saavad teenuse lugejatele nähtavaks. S.t. AUX ei varja kirjete lisandumist.
2. Kui vähemalt kaks AUX teenuses sertifitseerimisteenuse pakkujatena registreeritud TSP-d väidab "olemi A avalik võti on *pk*" ja vähemalt üks neist kahest TSP-st on OS-i poolt tunnustatud kui I tasemel olev, siis on *pk* olemi A avalik võti.
3. Ajatembeldusteenus on usaldusväärne. Ajatempel on oluline digiallkirja komponent, mis peab igal juhul olema ja mida me ei saa ignoreerida. Ajatembeldusteenust võiksime tegelikult vaadelda samuti ühe TSP teenusena (juhul, kui ajatembeldamine toimub vastavalt standardile RFC 3161 [5], mitte räsiahelapõhisena [6]) ning kasutada selle puhul samuti (3,2)-lävimudelit. Selleks peab olema vastavaid teenusepakkujaid usalduslistis piisavalt palju.

Süsteemi käideldavuseks on olulised järgmised eeldused:

- Igal ajahetkel on ülimalt üks mingit konkreetset usaldusteenust osutav TSP töökorrast väljas.
- Ülimalt üks mingit konkreetset usaldusteenust osutav TSP peatab samaaegselt selle teenuse osutamise. Siin samaaegsuse keelamine tähendab, et esimesena lahkunud TSP sertifikaadid jõuab uuendada enne teise TSP lahkumist.

### 3.3.5. Protsessid

#### 3.3.5.1. Uue TSP saabumine

Uue TSP saabumisel võtab ta ühendust registreerimisteenusega OS ning informeerib teda, et soovib süsteemiga liituda.

Uue TSP tulek on väga tundlik samm. Käesolev uuringuaruanne ei paku välja konkreetseid meetmeid, kuidas veenduda, et tegemist on töepooltest tunnustatud sertifitseerimisteenuse pakuja esindajaga või et pakutav teenus on tehniliselt mõistlik. Nende protseduuride formaliseerimine ja seotud riskide analüüs on meie töö skoobist väljas. Protseduuride paikapanemist toetavad ideid on võimalik võtta olemasolevatest regulatsioonidest nagu näiteks eIDAS-e usaldusnimekirjaga liitumise korraldusest.

#### 3.3.5.2. TSP asub pakkuma teatud usaldusteenust

Kui eelnevalt registreeritud TSP soovib hakata osutama mingit usaldusteenust, siis pöördub ta selle sooviga registreerimisteenuse OS poole. Teenuse osutamise loa aktiveerimiseks lisab OS AUX teenusesse kirje usaldusteenuse kohta koos taseme ja ajaga, millega alates teenus usaldutud on.

TSP jaoks hakkab tema töö välja nägema "tavapäraselt"; ta ei pea oma sisemisi protseduure

muutma selle tõttu, et OS andis talle loa teenust osutada. Kui näiteks TSP osutatavaks teenuseks on teatud sertifikaatide väljastamine, et olemitel on kontroll avalike võtmete üle (need võivad olla nii autentimis- kui ka signeerimisvõtmeh), siis jäabki TSP ülesandeks selliste sertifikaatide väljastamine vastavalt tema järgitavatele ja OS-i heaksiidetud eeskirjadele; TSP ei pea tegema pidevat koostööd teiste TSP-dega või riiklike teenustega.

### **3.3.5.3. TSP lõpetab teatud usaldusteenuse osutamise**

Kui TSP soovib lõpetada mingi usaldusteenuse osutamise, kuid mitte turult lahkuda, siis võtab ta ühendust OS-ga ning annab talle teada oma soovist teenuse osutamine lõpetada. OS muudab AUX teenuses vastava TSP usaldusteenuse kirje tühistatuks ja märgib ka tühistamise aja.

Kuigi TSP võib endiselt olla usaldusteenuste turul osutamas mõnda teist teenust, tuleb kõik tema väljastatud sertifikaadid ikkagi lugeda kehtivuse kaotanuks. Tõepoolest, sertifikaatide kehtivuse säilitamiseks teenuse osutamise lõpetamise hetkel peaks olema hiljem olemas võimalus neid sertifikaate tühistada. Samas on TSP selleks hetkeks, kui kasutajad mõnda sertifikaati tühistada sooviksid, teenuse osutamise juba lõpetanud ja ei pruugi seetõttu enam pakkuda viisi, mille kaudu tühistamissoovist teada anda. Üks võimalik alternatiiv oleks anda kontroll vanade sertifikaatide üle mõnele teisele TSP-le, kuid see on mittesoovitatav järgmistel põhjustel:

- Kui mõlemad TSP-d kuulusid mõne sertifikaadi jaoks seda sertifikaati väljastanud TSP-de kolmikusse (mille abil saavutati (3,2)-läviturvalisus), siis saab sertfikaadid vastu võtnud TSP-st nõrk lüli.
- Lahnunud TSP võib tulla turule tagasi uue nime all. Kui ta nüüd hakkab osalema sertifikaate väljastavates TSP-de kolmikutes koos tema vanad sertfikaadid vastu võtnud TSP-ga, siis saab temast nõrk lüli.
- Tehniliselt tähendab selline "sertifikaatide ülevõtmine" töötava teenuse üle andmist teisele organisatsioonile, mitte tühistamist.

Peame arvestama, et koos TSP-ga kaovad ka tema väljastatud sertifikaadid. Tänu lävi-PKI-le saavad lahnunud TSP-ga seotud kliendid ka edaspidi ennast autentida ning genereerida ja kontrollida digiallkirju. Samas peab arvestama, et see on võimalik vaid senikaua, kui veel üks TSP otsustab lahkuda. Seega peab lahnunud TSP-ga seotud sertifikaate võimalikult kiiresti uuendama. Juhul, kui TSP plaanib teenuse osutamise lõpetada, võiks ta sellest juba piisavalt varakult teada anda, et kasutajad jõuaks enne teenuse lõppemist vastavad sertifikaadid uuendada.

### **3.3.5.4. TSP lahkub turult (vabatahtlikult)**

Kui TSP soovib usaldusteenuste turult lahkuda, siis annab ta sellest OS-le teada. OS peatab kõik selle TSP osutatavad teenused AUX teenuses nii, nagu kirjeldab jaotis [3.3.5.3.](#)

### **3.3.5.5. TSP sunnitakse lahkuma**

Kui OS otsustab, et eelnevalt registreeritud TSP-d ei saa enam usaldada, siis peatab OS ise kõik TSP-ga seotud teenused, nagu kirjeldab jaotis [3.3.5.3.](#)

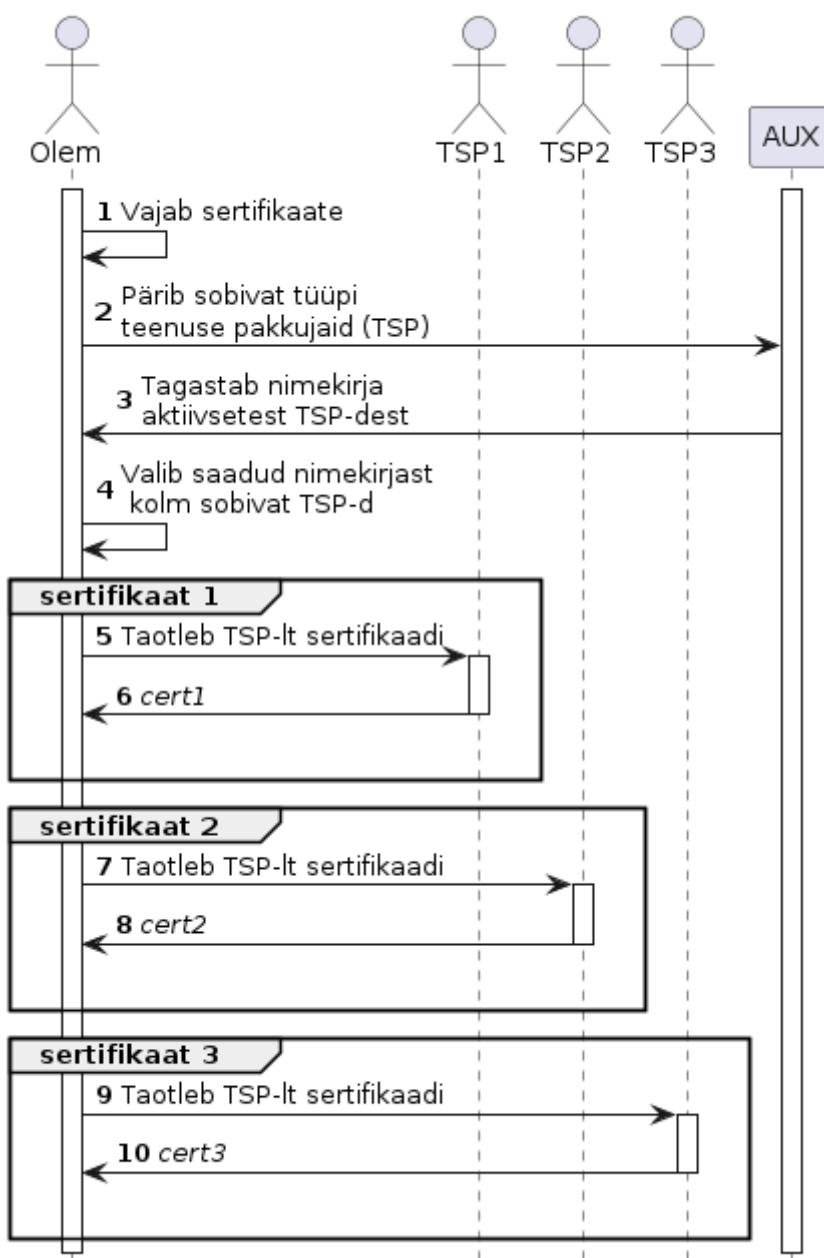
Erinevalt vabatahtlikust lahkumisest ei saa me enam eeldada, et TSP annab oma lahkumisest piisavalt varakult teada. Süsteemi käideldavust see ei häiri eeldusel, et kahe TSP korraga lahkumise tõenäosus on väga väike.

### 3.3.5.6. Olem küsib ja saab sertifikaadi teatud teenusest

Kui mõni olem soovib endale sertifikaate teatud teenusest, siis on tal tarvis võtta ühendust ja pärida sertifikaate vähemalt kolmelt TSP-lt, mis seda teenust osutavad. Selleks võtab ta kõigepealt ühendust riikliku andmebaasiga ning saab sealt nimekirja kõigist aktiivsetest sobivat tüüpi usaldusteenuse pakkujatest.

Olem peab sertifikaate saama vähemalt kahe I taseme TSP-lt, kolmas sertifikaat võib olla II taseme TSP väljastatud. Kuna olem peab niikuinii küsima värske TSP-de nimekirja teenuselt AUX, võime lihtsuse mõttes lugeda, et AUX valib ise välja sobivad TSP-d mingisuguse põhimõtte järgi (näiteks pakub välja need, kes on hetkel vähem koormatud).

Alternatiiv, kus lubatud on kaks II taseme sertifikaati kolmest, võib põhjustada olukorra, kus RP peab ise otsustama, kas ta usaldab olemit, millel on ainult kaks II taseme sertifikaati.



Sellise lähenemise oluline puudus see, et isik peab võtma ühendust kolme erineva TSP-ga seotud RA-ga ning tõendama oma identiteeti kolm korda. Praktikas oleks ilmselt vaja ühte usaldusväärset vahendajat (nagu Eestis PPA), mille kaudu selline tõendamine läbi viia.

### **3.3.5.7. Olem kontrollib sertifikaati**

Kui mõni olem näeb sertifikaati ja tahab kontrollida, kas see sertifikaat kehtib, siis võtab ta ühendust teenusega AUX ja küsib talt teenusepakkujale oleku kohta. Olem verifitseerib sertifikaadil oleva signatuuri krüptograafiliselt.

Sõltuvalt kontekstist võib oluline olla sertifikaadi staatus praegu või hoopis mingil ajahetkel minevikus.

- Kui on oluline praeguse hetke staatus ja TSP olek on aktiivne, siis teeb olem kehtivuse kontrollimiseks TSP-le OCSP päringu ja käsitleb sertifikaati vastavalt päringuvastusele. Lihtsuse mõttes loeme, et OCSP on osa TSP-st. OCSP käitlemine eraldi usaldusteenusena (mis võib olla omakorda nõrk lüli) on selle analüüsitoö skoobist väljas.
- Kui oluline on varasem sertifikaadi staatus, siis võib TSP praegu olek isegi olla tühistatud. Oluline on, et TSP olek oleks aktiivne sel varasemal ajahetkel. Sertifikaadi staatuse määramiseks varasemal ajahetkle peab olemas olema OCSP vastus ja kogu andmekomplekti kinnitatuv ajatempel. Kui sellist OCSP vastust ei ole või see ei kinnita sertifikaati kehtivaks, siis olem käsitleb sertifikaadi kehetuna.

### **3.3.5.8. Olem autendib ennast sõltuvale osapookele**

Kui mõni olem soovib ennast autentida mõnele sõltuvale osapookele (RP), siis esitab ta osapookele vähemalt kaks kolmest sertifikaadist, mis ta saanud on (nagu kirjeldatud jaotises 3.3.5.6). RP kontrollib kõiki sertifikaate (nagu kirjeldatud jaotises 3.3.5.7) ning veendub, et järgmised asjaolud on tõesed.

- Kõik sertifikaandid on väljastatud eri TSP-de poolt.
- Kontrolli läbivad vähemalt kaks sertifikaati, millega on väljastatud üks on I taseme TSP poolt.

Lisaks verifitseerib sõltuv osapool autentimissignatuuri krüptograafiliselt.

### **3.3.5.9. Olem kontrollib signatuuri**

Kui olem saab dokumendi  $D$  koos signatuuriga  $\sigma$ , millega kaasnevad kolm sertifikaati, siis kontrollib olem seda sertifikaatide kolmikut analoogiliselt kontrollile, mida teeb RP, kui mõni olem ennast talle autentida tahab (jaotis 3.3.5.8). Samuti verifitseerib ta dokumendi signatuuri krüptograafiliselt.

### **3.3.5.10. Ennustamine, kas kolmas osapool aktsepteerib signatuuri**

Olgu mõnel olemil dokumendi  $D$  koos signatuuriga  $\sigma$ . Ta soovib välja selgitada, kas kolmas osapool selle signatuuri aktsepteerib (praegu või tulevikus). Selleks, et veenduda, kas kolmas osapool võtab signatuuri vastu *praegu*, piisab, kui olem kontrollib selle signatuuri ise. Selleks on vaja vähemalt kahte praegusel hetkel kehtivat sertifikaati, nagu kirjeldatud jaotises 3.3.5.9.

Selleks, et kolmas olem aktsepteeriks signatuuri ka mingil *tulevasel ajahetkel*, tuleb eeldada, et kehtib vähemalt üks järgmistest asjaoludest:

1. signatuur on varustatud ajatempliga, mis lubab tulevikus tõendada, et vähemalt kaks

- sertifikaati olid praegusel hetkel kehtivad;
2. sertifikaadid väljastanud TSP-dest peavad vähemalt kaks olema ka tulevikus töökorras ning pakkuma (isiku)sertifikaatide väljastamise ja elutsükli halduse teenust.

Teine neist eeldustest on ebamõistlik, sest kuigi on mõistlik eeldada, et kolmikus osalevatest TSP-dest ei saa kaks olla *korraga* töökorrast väljas (ühe TSP sertifikaadid jõuab asendada enne, kui midagi juhtub teise TSP-ga), oleks liiga optimistlik uskuda, et vähemalt kaks kolmest TSP-st ei lahku turult *mitte kunagi* tulevikus. Seega, nagu praegugi, peab digiallkirjaga kaasas olema ajatempel ja igale sertifikaadile vastav OCSP-vastus, mis on saadud signatuuri loomise ajal.

### **3.3.5.11. Olem kasutab digiallkirjateenust**

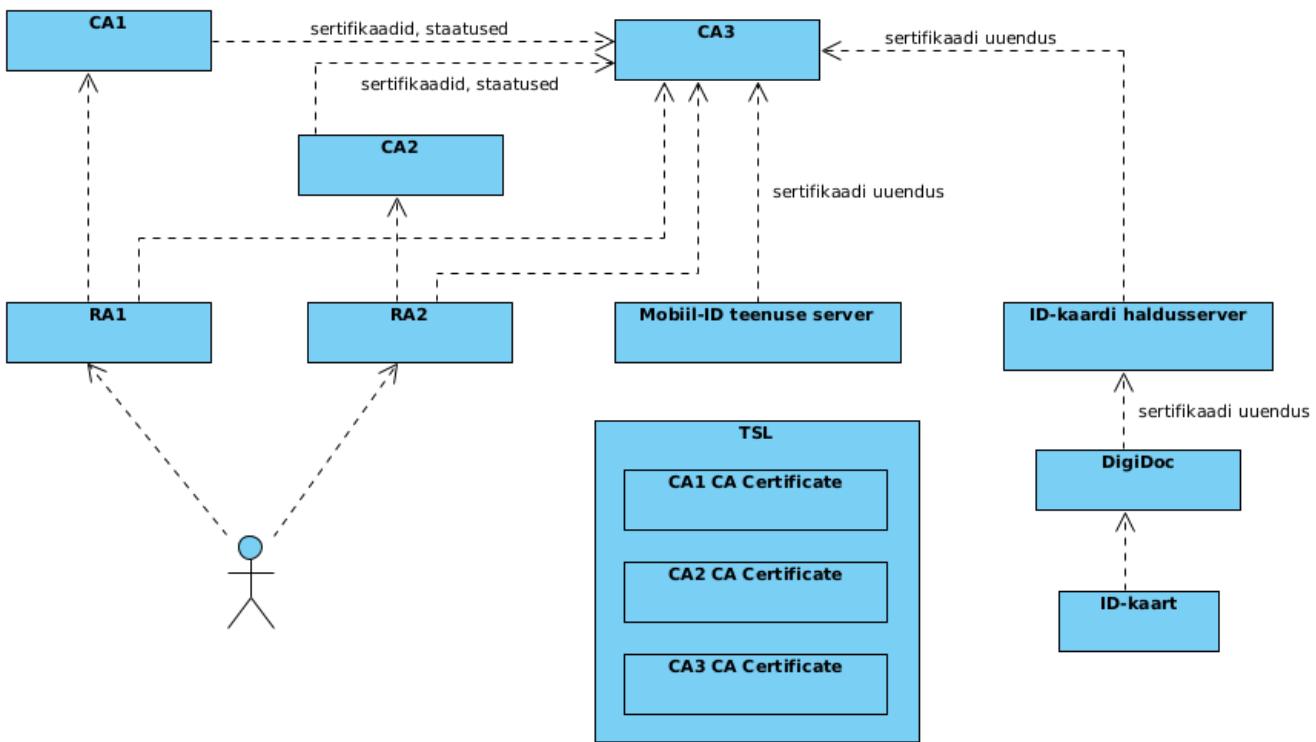
Olem kasutab oma privaatvõtit, et genereerida dokumendil  $D$  krüptograafiline signatuur  $\sigma$ . Ta edastab selle signatuuri digiallkirjateenusele ühes vastava avaliku võtmega seotud kolme sertifikaadiga. Teenus kontrollib signatuuri nagu kirjeldatud jaotises [3.3.5.9](#). Positiivse vastuse korral lisab teenus ajatempli ja saadab valmis digiallkirja olemile tagasi. Siin on oluline jaotises [3.3.4](#) mainitud eeldus, et ajatempliteenus on usaldusväärne. Ajatempli lisamine on oluline selleks, et mõni teine olem antud signatuuri ka tulevikus vastu võtaks, vt jaotis [3.3.5.10](#).

### **3.3.5.12. Väljaandja tühistab sertifikaadi**

Iga TSP võib tühistada iseenda väljastatud sertifikaate ja nendega seotud sertifikaadihelaid. Selline tühistamine toimub lokaalselt. On ilmne, et me ei saa nõuda, et ta teeks midagi teiste TSP-de poolt väljastatud sertifikaatidega. Isegi kui TSP kahtleb kasutaja aususes, on tal õigus tühistada vaid enda väljastatud sertifikaati. Kui kasutaja ise kahtlustab, et tema privaatvõti on lekkinud, siis peab ta sellest ise teavitama kõiki TSP-sid, mis on vastavale avalikule võtmele sertifikaadi andnud.

## **3.3.6. Aktiivsed ja passiivsed TSP-d**

Senikaua eeldasime, et kõik kolm sertifikaate väljastavat TSP-d on aktiivsed ning kasutajal on kolm erinevat sertifikaati, mida ta sõltuvatele osapooltele esitab. Kuna autentimiseks ja signatuuri kontrollimiseks piisab, et kolmest sertifikaadist oleksid kehtivad kaks, võime esialgu piirduda kahe sertifikaadiga ja kahe aktiivse TSP-ga. Kolmandat läheb vaja alles siis, kui ühega olemasolevatest aktiivsetest TSP-dest midagi juhtub.



Kui mõne aktiivse TSP-ga midagi juhtub, siis peab passiivne TSP aktiveeruma ja endise aktiivse TSP töö üle võtma. Ülevõtmiseks on vajalik, et passiivne TSP teab täpselt, milline on aktiivse TSP väljastatud sertifikaatide staatus. Seega peab passiivne TSP saama mõlemalt aktiivselt TSP-lt reaalajas kõik nende väljastatud sertifikaadid ja kõik sertifikaatide staatuse muutused.

Kahe aktiivse ja ühe passiivse TSP-ga mudeli eelised:

- Kahe sertifikaadiga opereerimiseks on vaja vähem mälu.
- Kahe sertifikaadi kontrollimise loogika on lihtsam (ei pea läbi vaatama eri kombinatsioone).
- Kahe aktiivse TSP ülalpidamine on kolmest aktiivsest TSP-st odavam.

Kahe aktiivse ja ühe passiivse TSP-ga mudeli puudused:

- Kolme aktiivse TSP-ga mudelis (kus igal TSP-l on oma sõltumatu RA ja CA) on sertifikaatide kuritarvitamise korral lihtsam välja selgitada, kes neist kolmest on süüdi. Tõepoolest: eeldusel, et rikutud on ülimalt üks TSP, on süüdi see, kelle väited ei ole kooskõlas kahe ülejäänud TSP väidetega. Kahe aktiivse TSP korral pole see võimalik.
- Kui passiivne TSP astub äralangenud aktiivse TSP asemele, peab ta kõik selle sertifikaadid uuendama, ehk siis uuesti välja andma, signeerides need oma privaatvõtmega. See võib süsteemis tekitada ajutise törke, mida kolme aktiivse TSP-ga ei juhu.
- Kui digiallkirja küljes on kolm sertifikaati, siis saab ka tulevikus ühe TSP lahkumise korral veenduda selle korrektsuses. Kui selle küljes on vaid kaks sertifikaati, siis ei saa passiivne TSP seda enam tagantjärgi uuendada, nii et digiallkirja kontrollimiseks peab suuremal määral tuginema ajatemplitele.

Juhime tähelepanu sellele, et aktiivsete ja passiivsete TSP-de loogika on üldistatav (3,2)-lävimudelist suvalissele ( $n,t$ )-lävimudelile. Esialgu on töös  $t$  aktiivset TSP-d ning sõltlane ootab kasutajalt  $t$  erinevat sertifikaati. Ülejäänud ( $n-t$ ) TSP-d on passiivsed. Kui mõne aktiivse TSP-ga juhtub midagi halba, asendab selle mõni neist passiivsest TSP-dest. Jaotises 3.5 kirljedame täpsemalt (2,1)-

lävimudelit, kus üks TSP on aktiivne ja üks on passiivne.

## 3.4. Väljapakutud arhitektuuri kitsaskohad

### 3.4.1. Mitme sertifikaadi kontrollimine

Terviklusomaduste tugevdamiseks on oluline, et isiku ja avaliku võtme vahelist seost peaks kinnitama enam kui üks CA. Nõude, et kasutaja peab saama ühe sertifikaadi asemel kolm ning RP peab nõudma ühe sertifikaadi asemel kaht (või enamat), sissetoomine ei valmista teoreetilisi raskusi. Samuti ei ole RP poolt kahe sertifikaadi kontrollimine tehniliselt keeruline. Samas pole tegemist sugugi standardse lahendusega. Seega jäavad ebaselgeks mitme sertifikaadi kontrollimise tehnilised detailid.

Kahe sertifikaadi kombineerimiseks näeme hetkel kahte võimalust. Kirjeldame mõlemat allpool. Näeme, et mõlemad neist nõuaksid mingisuguseid RP-poolseid muudatusi. See tähendab, et mitme sertifikaadi kombineerimist saaks küll rakendada Eesti-siseselt, kuid teenused poleksid automaatselt ühilduvad eIDAS-ega. Seega leiame, et kumbki alltoodud konstruktsioonidest pole siiski sobilik Eesti usaldustaristus kasutuselevõtuks.

#### 3.4.1.1. Mitme sertifikaadiga avalikud võtmeh

Hetkel on Eesti riigiteenustel ja pankadel kasutuses standardsete autentimisvahendite kogumid, nagu näiteks ID-kaart ja Smart-ID, millega kasutaja saab valida endale sobivama. Autentimistarkvaras toetab neid vahendeid korduvalt kasutatav Web eID tarkvarakomponent [7], mida saavad kasutada erinevad RP-d. Kahe sertifikaadi loogika peaks olema kasutusele võetud selles komponendis. RP võiks saada valida, kas lubada nõrgemat ühekihilist autentimist (ühe sertifikaadiga) või nõuda tugevamat kahekihilist (kahe sertifikaadiga). Web eID komponent peaks ka kliendi poolt toetama mitme sertifikaadi kasutamist; usutavasti ei tooks sellise toetuse lisamine kaasa olulisi muudatusi seda kasutatavates rakendustes.

Samas digiallkirjadele see lähenemine hästi ei sobi, sest nende loomisel ja kontrollimisel kasutatakse sagestasti olemasolevat EU DSS teeki [8], mille üle meil puudub Web eID-ga võrreldav kontroll. Tehniliselt võiks digiallkirja saaja nõuda, et ühe ja sama dokumendiga käiks kaasas mitu erinevat signatuuri, kuid siis oleks lisaks EU DSS teegis realiseeritud kontrollidele tarvis veenduda, et:

1. iga isik, kes on digiallkirjastatud dokumendi allkirjastanud, on sellele andnud (vähemalt) kolm signatuuri;
2. need signatuurid on seotud erinevate CA-de poolt väljastatud sertifikaatidega.

See vajab lisakontrolle. Selleks, et olla ühilduv eIDAS-ega, peaksid sama kontrolli teostama ka teistes Euroopa Liidu riikides asuvad isikud, mis vajaksid nende poolt muudatusi. Selliste lisakontrollimismeetmete lisamine ei pruugi ühilduda eIDASe digiallkirjade regulatsiooniga.

#### 3.4.1.2. Lävisigneeritud sertifikaadiga avalikud võtmeh

Me võiksime ka püüda mitte sundida RP-d muutma oma käitumist autentimisprotsessis. Kui RP-le esitatakse sertifikaat, siis ta kontrollib sellega kaasaskäivat sertifikaatide ahelat ning võtab selle

vastu siis, kui jõuab vähemalt ühe usaldatava lülini (näitkeks, süsteemi sisse kodeeritud juursertifikaadini). Probleem on selles, et isegi ainult üks usaldatud avalik võti võib RP-d veenda ükskõik milles. Et sellele avalikule võtmele vastav salajane võti ei oleks nõrgaks lülik, on oluline, et seda salajast võtit ei tunneks mitte keegi ning et sertifikaadi signeerimine oleks võimalik vaid kahe erineva CA koostöös. Seega on tarvis kasutada lävisigneerimist, kus sertifikaate signeeriv võti ei kuulu täielikult ühele CA-le, vaid on jagatud kahe erineva CA vahel. Meie (3,2)-lävipõhise lahenduse korral tähendaks see, et sertifikaadi võivad tekitada ükskõik millised kaks kolmest CA-st. Kasutaja kätte jäiks kolme sertifikaadi asemel üks, mis vastaks standardile. Seejuures on eriti oluline sertifikaadi väljastamise korraldus:

1. Isik võib kontakteeruda kolme erineva RA-ga, kes peaksid siis algatama sertifikaadi allkirjastamise erinevate CA-de koostöös. Selline protseduur võib olla liiga keeruline, sest vajab sünkroonsust erinevate CA-de vahel.
2. Kui vahendajaks isiku ja CA-de vahel on üks ainus RA, siis on see RA ise nõrk lüli. Eestis võime eeldada, et vahendajaks on PPA, kellel võib olla lihtsam eri CA-de tööd omavahel sünkroniseerida.

Peale väljastamist peab olema võimalik teha OCSP päringuid, et saada teada, ega sertifikaat pole tühistatud. Kuna seda päringut teeb RP, mitte kasutaja, siis on oluline, et OCSP vastus oleks standardne, ehk siis näeks välja kui see oleks tulnud ühe CA poolt. Siin oleks jällegi tarvis lävisigneerimist. Erinevalt sertifikaadi väljastamisest ei saa olla vahendajaks PPA, sest OCSP pärinuid hakatakse tegema pidevalt. Igasugune vahendaja oleks aga nõrk lüli.

Selline lahendus võiks töötada Eesti-siseselt, kuigi raskustega, mida me kõiki veel ei pruugi ette näha. Kuidas aga näeksid sellist, jagatud võtmega signeeritud sertifikaati teised kasutajad, kes pöörduvad EL-i usaldusnimekirja poole? Praegu peaksid nad tunnistama neid sertifikaate, mis on signeeritud usalduslistis loetletud teenusepakkujate poolt. Kui sertifikaadi signatuur on valminud kahe erineva TSP koostööna, siis seda enam otsest teha ei saaks. Pigem peaks siis tervet CA-de kolmikut vaatlema kui ühte virtuaalset CA-d. See aga tähendab, et CA-de kolmikut peaks tunnustama eraldi TSP-na. See on väga erinev sellest, et iga kolmikus esinev CA kuulub tunnustatud TSP-le.

### **3.4.2. Kas ühe või teise sertifikaadi kontrollimine**

Käideldavuse suurendamiseks on oluline, et isiku ja avaliku võtme seose kinnitamises võiksid osaleda erinevad CA-d. Jällegi ei ole sellise võimaluse sissetoomine keeruline ei teoreetiliselt ega ka tehniliselt, kuid tegemist on ebastandardse lahendusega. Jaotises [3.3](#) pakkusime välja, et isiku ühe ja sama võtmega on seotud mitu erinevat sertifikaati. Meie teada ei ole see aga praeguste seadustega lubatud. Allpool pakume välja veel mõne viisi käideldavuse töstmiseks. Kõik need lahendused annavad samasuguse käideldavusomaduse: süsteem töötab seni, kuni töötab mõni CA-d, mis on väljastanud mõne sertifikaadi mõnele avalikule võtmele, millele vastavat privaatvõtit kasutaja kontrollib. Lahendused erinevad tehnilise ja juriidilise teostuse lihtsuse poolest.

#### **3.4.2.1. Mitu erinevat võtit, igaühel üks sertifikaat**

Kasutajal võiks olla mitu erinevat avalikku võtit, millest igaüks on seotud ühe ainsa sertifikaadiga. See ei tohiks juriidilisi probleeme valmistada, sest ka praegu saab ühel isikul olla mitu erinevat võtit, näiteks ID-kaardis on üks ja mobiilis teine. See aga tähendab, et kui üks neile võtmetele

sertifikaate väljastanud CA-dest parasjagu ei tööta, siis ei saa RP enam lihtsalt esitada OCSP päringut alternatiivsele CA-le. Autentimise või digiAllkirjastamise nurjumisel peaks isik proovima sama asja uuesti, seekord aga teise võtmega. CA väljavahetamisel peaks uuendama ka vastavat võtit. Võrreldes jaotises 3.3 väljapakutud lahendusega vajab mitme võtme hoidmine rohkem mälu kasutaja seadmetes.

#### 3.4.2.2. Üks võti ja üks sertifikaat

Lihtsaimal juhul on isikul üksainus võti ja sellel üksainus serifikaat. Käideldavuse tagamiseks on aga oluline, et serifikaadi kontrollimise ajal saaks OCSP päringule vajadusel vastata ka mõni teine CA, s.t. selline, mis pole ise seda sertifikaati väljastanud. See teine CA ei saa kuidagi garanteerida, et esimese CA sertifikaadid pole võltsitud, nii et terviklusomadused on selle lahenduse korral nõrgemad.

Käideldavuse jaoks on oluline, et CA-d oleksid omavahel sünkroniseeritud: iga kord, kui üks CA väljastab sertifikaadi, peab ta sellest teavitama teist CA-d. See lahendus on teostatav, kui CA-sid pole palju. Juriidiliselt tuleks lubada, et OCSP päringule vastab mitte sertifikaadi väljastanud CA ise, vaid mõni teine CA. Selleks peavad CA-d üksteist ristsertifitseerima. Praktikas võib sertifikaatide väljastamisega tegeleda aktiivselt vaid üks CA ning teine võib olla ootel, valmis sekkuma, kui esimese CA-ga midagi juhtub.

Legaalsest vaatepunktist on oluline, et sertifikaadi võltsimise korral oleks jälgitav, kas esimene CA väljastas vale sertifikaadi või teine CA vastas valesti OCSP päringule. See ei tohiks olla probleem, sest nii sertifikaadid kui ka OCSP päringud on vastava CA poolt signeeritud.

### 3.5. Meie soovitatav lahendus

Jaotises 3.4 järeldasime, et vaatamata tehnilisele lihtsusele vajaks avaliku võtme omamise kinnitus "nii ühe kui ka teise CA poolt" muudatusi olemasolevates regulatsioonides ja standardsetes sertifikaadiformaatides. Samas peaks kinnitus "kas ühe või teise CA poolt" olema teostatav. Seega pakume alternatiivi, mis küll annab väiksemaid terviklusgarantiisiid, kuid selle eest on paremini sobivam praeguste regulatsioonide ja tänaste standardsete protokollidega. Selleks võtame kasutusele (2,1)-lävimudeli, kus sertifikaate väljastavad kaks erinevat CA-d ning mõlema sertifikaate loetakse õigeks.

Käitleme ajatemplite lahendust ja sertifikaatide lahendust eraldi, kuna ajatemplite ründe kindluse tagamine peale ajatempli väljastaja võtme ründaja kätte sattumist vajab samuti standardite täiendamist.

#### 3.5.1. Ajatembeldamine

Ajatempliteenuse sertifikaadis olevale avalikule võtmele vastava privaatvõtme ründaja kontrolli alla sattumisel saab ründaja tekitada mineviku kuupäevaga ajatempleid. Selle välimiseks peab ajatempli verifitseerija saama ajatempli väljastaja võtmest sõltumatult kontrollida, kas konkreetne ajatempel eksisteeris ajal, mis on selles ajatemplis määratud. Praktikas on lihtsaim lahendus kas avalik plokiahel või muu sõltumatu register, kust verifitseerija saab kontrollida, kas ajatempel tõesti oli olemas väidetaval ajahetkel.

Paraku olemasolevad standardid taolist verifitseerimist ei toeta. Lisakontroll on vajalik ainult

juhul, kui ajatempliteenust on edukalt rünnatud. Verifitseerija peab ka kuidagi teada saama, et selline sündmus on toimunud, ja teistmoodi käituma.

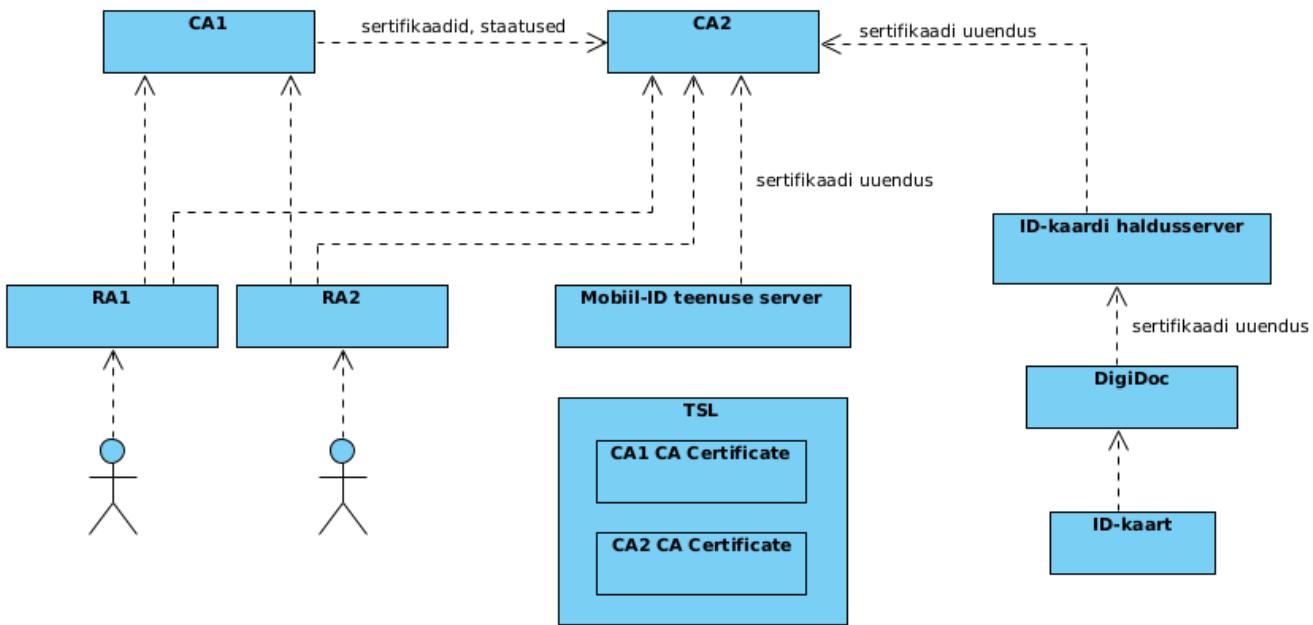
Standardset lahendust seega pole. Võimalik oleks ainult Eesti DigiDoc-programmi täiendamine lisakontrolliga ning kaitse laieneks seega ainult selle programmi kasutajatele. Parem lahendus oleks standardne protokoll, mida implementeeriks EU DSS teek.

### 3.5.2. Sertifitseerimine

Standardprotokolle muutmata on täiendava CA sissetoomiseks kaks valikut. Esimese variant on lisada autentimis- ja signeerimisvahendisse kohe mitu sertifikaati (ja võtit) ning toimingu ajal valida neist üks. See vastaks jaotises [3.3.6](#) kirjeldatud kahe aktiivse TSP variandile. Teise variandina oleks toetatud korraga ainult üks CA, kuid selle hävimisel väljastatakse kõigile uued sertifikaadid teise CA poolt. See vastaks jaotises [3.3.6](#) kirjeldatud ühe aktiivse ja ühe passiivse TSP variandile. Soovitame ja kirjeldame teist lahendust, kuna vajalikud muudatused oleksid väiksemad ja ei muudaks praegusi kasutusviise. Muudatused on vajalikud ainult juhuks, kui peamine CA hävib.

Meie väljapakutavas süsteemis on kaks eri TSP-de osaks olevat CA-d: primaarne CA1 ja sekundaarne CA2. Esialgu on aktiivne vaid CA1. Samuti on süsteemis olemas mitmed RA-d, kes korraldavad registeerimisprosesesse. Senikaua eeldasime, et iga CA-ga on seotud täpselt üks RA, nii et CA1-ga on seotud RA1 ja CA2-ga on seotud RA2, kuid praktikas võib see olla paindlikum (nii RA1 kui ka RA2 võivad suhelda aktiivse CA1-ga). Süsteemis on olemas erinevad autentimis- ja signeerimisvahendid ning teenused. Kõik töötab nagu tänagi primaarsel CA-d kasutades, kuid lisanduvad järgmised tegevused:

- CA1 saadab reaalajas kõik enda väljastatud sertifikaadid ja kõik sertifikaatide staatuste muudatused CA2-le. CA2 hoiab registrit kõigist sertifikaatidest ja nende staatustest. Lisaks on kasutusel mingit laadi sünkroniseerimismehhanism, mis garanteerib, et CA2-l oleks kindlasti olemas sertifikaatide ja nende staatuste andmed, mis langevad täpselt kokku CA1-e andmebaaside sisuga.
- Kui CA1 saadab värskelt väljastatud sertifikaadi CA2-le, siis CA2 salvestab selle, et olla tulevikus ise valmis väljastama sertifikaati, millel on täpselt sama subjekt, kehtivuse lõpuaeg, avalik võti ja tehniline profiil.
- Nii CA1 kui CA2 väljastatud sertifikaate võib usaldada. Praktikas tähendab see, TSP-d, mille osaks on vastavalt CA1 ja CA2, on registeeritud *EU TSL* usaldatud teenusepakkujate regisistris kui sertifikaatide väljastamise ja elutsükli halduse teenuse pakkujad.



Juhul kui CA1 kasutamine pole enam võimalik (näiteks CA1 privaatvõti lehib või hävib ning seega ei saa ka enam CA1 kehtivuskinnitusteenust kasutada), siis riik annab *korralduse ümber lülituda* CA2 peale ja *tühistada* CA1. Tühistamine tähendab, et kõik CA1 väljastatud sertifikaadid kaotavad kehtivuse. Ümber lülitumine tähendab, et CA2 peab väljastama need sertifikaadid, mille väljastamiseks ta tänu ülalkirjeldatud andmevahetusele valmis on. Sertifikaatide asendamisega seotud tehnilised detailid on järgmised:

- Teenused nagu Mobiil-ID ja Smart-ID, kus kasutaja sertifikaadid on salvestatud teenuse serveris, saavad asendada kasutajate sertifikaadid masspäringuga.
- Vahendid nagu ID-kaart peavad sertifikaati vahetama lõppkasutaja abiga, kasutades DigiDoc-programmi.
- Uute sertifikaatide väljastamiseks peavad RA-d CA1 pealt CA2 peale ümber lülituma. Samuti peavad nad seda tegema sertifikaatide peatamiseks ja tühistamiseks.
- Asendada on võimalik ainult kehtivaid sertifikaate, kuna CA2 omab värsket sertifikaatide staatust.
- Vanade sertifikaatide kehtivuskinnitusteenus pole oluline, kuna CA väljastaja kehtivuskinnitusteenus ütleb, et CA1 on ise kehtetu.
- Kehtivuskinnitusteenus on peale sertifikaadi uuendamist automaatselt CA2 käes ja jätkub.

Meie väljapakutaval lahendusel on järgmised eelised:

- Muutused on minimaalsed, sest kuni CA1 hävimiseni töötab kõik nagu tänagi.
- CA1 hävimisel on vajalik vaid sertifikaatide uuendamise protseduur, võtmed jäävad samaks.
- Mingit ristsertifitseerimist CA1 ja CA2 vahel pole vaja teha ning usaldus kummagi CA vastu on üksteisest sõltumatu.

Lahendusel on järgmised puudused:

- Täiendava CA ülalpidamine võib olla kallis ka siis, kui ta on passiivses olekus.
- Paljude sertifikaatide korraga väljastamine on tehniliselt keeruline ja koormav.

Viimast puudust võiks vältida nii, et CA2 väljastab sertifikaadid kohe, kui ta saab CA1-lt tema väljastatud sertifikaadid. Selle suureks eeliseks on peale ümberlülitumist suure sertifikaatide väljastamisega seotud koormuse puudumine. Teisest küljest jällegi võimaldab sertifikaatide kohesest väljastamisest ja staatusemuutmistest hoidumine ära hoida mõned ründed. Näiteks kui CA1 hävib, kuna teenusetõkestusrünnet sooritada püüdev ründaja on saanud ligipääsu tühistamise liidesele ja massiliselt sertifikaate tühistanud, siis oleks ümberlülitumisel võimalik lülitumishetkeks valida mõni varasem, ründele eelnened ajahetk ning sellega ka korrigeerida ründajapoolsetest tühistamistest tekkinud ebakõlad.

## 3.6. Kvantitatiivne analüüs

Käesolevas jaotises loeme, et süsteemis on kuni kolm erinevat CA-d (millest igaüks on erineva TSP osa); üldisem analüüs koos täpsemate arutluskäikudega on toodud Lisas D. Hindame pakutud arhitektuuri käideldavust ja terviklust sarnaselt jaotisega B.6. Nimelt oletame, et ründaja võtab üle mingi arvu CA-sid (üks kuni kolm) ning hindame, mitut kasutajat see mõjutab. Ründaja edukuse hindamiseks käideldavuse (AV) ja tervikluse (INT) rikkumisel kasutame mõjutatud kasutajate osakaalu. Piltlikult kujutame mõjutatud kasutajate osakaalu järgmiselt:

- : mitte ükski kasutaja pole mõjutatud.
- : kolmandik kasutajaid on mõjutatud.
- : pool kasutajaid on mõjutatud.
- : kaks kolmandikku kasutajaid on mõjutatud.
- : kõik kasutajad on mõjutatud.

Võrdleme ründaja edukust erinevate usaldusmudelite korral:

- Üksik CA on lihtne usaldusmudel ühe ainsa TSP ja tema osaks oleva CA-ga. Valisime selle mudeli võrdluseks, sest seda kasutatakse praegu.
- (3,2)-lävi on jaotises 3.3 kirjeldatud lahendus. Käesolevas analüüs is me ei erista I ja II usaldustaseme TSP-sid, sest need taseme reguleerivad ainult seda, kas CA-d võivad saada kontrollitud ühe ja sama ründaja poolt. Käesolevas analüüs is oleme me ühe ja sama ründaja kontrolli all olevate CA-de arvu võtnud üheks analüüsiparametrikks.
- (2,1)-lävi on jaotises 3.5 kirjeldatud lahendus.

Juhul, kui kasutajad pole CA-de vahel eelnevalt jagatud ning sertifikaate võib potentsiaalselt saada ükskõik milliselt CA-lt, siis annab (piisavalt suure arvu) sertifikaatide võltsimine ründajale võimaluse ükskõik kelleni esineda. Seega on tervikluse mõttes alati mõjutatud kas kõik kasutajad korraga või mitte keegi neist. Alternatiivselt võib kasutajad jagada mittekattuvateks domeenideks, millest igaüks tohib kasutada vaid konkreetsete TSP-de teenuseid läbi nende osaks olevate CA-de. Nimelt eeldame, et  $(n,t)$ -lävi mudeli korral on igal kasutajal konkreetne valik CA-sid (neid CA-sid on valikus  $n$  tükki), kellelt ta võib saada kehtivaid sertifikaate. Selleks peavad kõik CA-d olema aktiivsed (aktiivsed ja passiivsed CA-d on kirjeldatud jaotises 3.3.6), sest muidu oleks piisanud parasjagu aktiivsetest CA-dest vaid üheksainsaks domeeniks.

Üht domeeni teeninidava CA ülevõtmise ei anna ründajale võimalust võltsida mõnda teise domeeni kuuluvate kasutajate sertifikaate. See suurendab süsteemi terviklust, kuid vähendab lisapiirangute tõttu käideldavust. [Tabel 1](#) näitab, millised on eri mudelite käideldavus- ja terviklushinnangud (tabeli ridadel tähistatud vastavalt "AV" ja "INT").

Tabel 1. Kvantitatiivne ründaja edukus erinevate eelduste ja mudelite korral.

Kasutajad on domeenideks jagatud	Ei				Jah			
Maksimaalne rikutud CA-de arv	0	1	2	3	0	1	2	3

Kasutajad on domeenideks jagatud		Ei				Jah			
Üksik CA	AV	□□□	■■■	■■■	■■■	□□□	■■■	■■■	■■■
	INT	□□□	■■■	■■■	■■■	□□□	■■■	■■■	■■■
(3,2)-lävi	AV	□□□	□□□	■■■	■■■	□□□	□■■	■■■	■■■
	INT	□□□	□□□	■■■	■■■	□□□	□□□	□□■	■■■
(2,1)-lävi	AV	□□□	□□□	■■■	■■■	□□□	□■■	■■■	■■■
	INT	□□□	■■■	■■■	■■■	□□□	□■■	■■■	■■■

Tabel 2 esitab mudelite omadused, mis võivad olla seotud nende realiseerimisse ja ülalpidamisse kuludega. Täpsemat rahalist hinnangut me selles osas anda ei oska.

Tabel 2. Erinevate mudelite omadused, mis võivad olla seotud kuludega.

	kehtivaid sertifikaate kasutaja kohta	CA-de koguarv	vajab muudatusi sõltuvate osapoolte loogikas
Üksik CA	1	1	ei
(3,2)-lävi	2	3	jah
(2,1)-lävi	1	2	ei

## 4. Õiguslik hinnang

### 4.1. Sissejuhatus

Antud peatükis on seatud eesmärgiks hinnata, kas eelmises peatükis välja pakutud soovituslikku eID usaldusmudelit saab teostada kehtiva õiguse alusel ning milliseid muudatusi tuleks kehtivas õiguses teha selle kasutuselevõtmiseks.

Õigusliku hinnangu andmisel on lähtutud individuaalse identiteedisüsteemi ja üldise identiteedisüsteemi tasandil kehtivatest õigusaktidest Eestis, mida on täpsemalt kirjeldatud Lisas C. Muid õigusvaldkondi ei ole käsitletud, sest need nõuaksid konkreetsemat ülesandepüstitust ja täpsemaid asjaolusid kui antud analüüs teostamise ajal on võimalik mõistlikult ette prognoosida. Ka ei ole käsitletud nn pehme õiguse akte ega õigusaktide eelnõusid.

Õigusliku hinnangu kujundamisel on jäädud pigem üldisele kontseptuaalsele tasandile, et kaardistada võimalikke põhimõttelisi muudatusvajadusi kehtivas õiguses. Pole välisstatud, et eelmises peatükis välja pakutud soovitusliku eID usaldusmudeli tegelikul kasutuselevõtmisel ilmneb üksikasjalikumaid vastuolusid kehtiva õigusega, mida siinse õigusliku hinnangu raames ei tuvastatud. Praktikas elluviimiseks väljavalitud eID usaldusmudeli kasutuselevõtmisel tuleb teostada täiendavad õiguslikud analüüsides, mis võtavad arvesse vahepeal selgunud uusi asjaolusid ja vajadusi — mida selgemaks saab tulevase eID usaldusmudeli sisu ja rakendamise kontekst, seda põhjalikumaks tuleks minna ka vastavate õiguslike analüüsidega.

## 4.2. Asjaolude kirjeldus

Eesti eID toimimine sõltub sertifitseerimistaristust. Selle taristu keskmes on sertifitseerimiskeskus, kes seob kodanike identiteedid krüptograafiliste võtmepaaridega ja töendab vastavaid seoseid, pakkudes järgmisi teenuseid:

1. digitaalset tuvastamist ja digitaalset allkirjastamist võimaldavate sertifikaatide väljastamine ja kehtetuks tunnistamine;
2. digitaalset tuvastamist ja digitaalset allkirjastamist võimaldavate sertifikaatide kehtivuse kontrollimine.

Antud uuringu koostamise hetkel tegutseb Eestis üks sertifitseerimiskeskus. Vastavalt Politsei- ja Piirivalveametiga sõlmitud halduslepingule täidab sertifitseerimiskeskuse kohustusi IDEMIA, kes on isikut töendavate dokumentide tootmise teenust pakkuv ettevõte. Sama halduslepingu alusel on sertifitseerimiskeskuse ülesanded omakorda üle antud alltöövõtjale (SK ID Solutions AS), kes väljastab isikut töendavatele dokumentidele elektronilised sertifikaadid ning osutab mh ka kehtivuskinnitusteenust [9].

Kui sertifitseerimiskeskuse teenused ei toimi, siis mõjutab see eID väljaandmist ja elutsükli haldust ning eID kasutamist köikidel tasanditel peaaegu kogu eID taristu ulatuses. See on eID taristu väga oluline nõrkus. Sõltuvuse vähendamiseks ühest sertifitseerimiskeskusest tellis Riigi Infosüsteemi Amet uuringu, mille teostamiseks koostati käesolev aruanne. Aruande eelmises peatükis pakuti välja soovituslik eID usaldusmudel, kus ühe sertifitseerimiskeskuse asemel oleks taristus kaks teineteisest sõltumatut keskust:

1. **aktiivne sertifitseerimiskeskus**—esimene sertifitseerimiskeskus on aktiivne, s.t väljastab kõik sertifikaadid, tunnistab neid kehtetuks ja vastab sertifikaatide kehtivuspäringutele.
2. **passiivne sertifitseerimiskeskus**—teine sertifitseerimiskeskus on passiivne, s.t peab enda juures hoidma ja ajakohastama koopiat kõigist esimese sertifitseerimiskeskuse väljastatud sertifikaatide olekutest, kuid ei väljasta ega tunnista kehtetuks sertifikaate ega vasta sertifikaatide kehtivuspäringutele.

Väljapakutud soovitusliku eID usaldusmudeli kohaselt, kui esimene sertifitseerimiskeskus ei saa toimida ega teenuseid pakkuda, siis lülitatakse ümber teisele keskusele. Ümberlülitumise hetkel muutub teine sertifitseerimiskeskus aktiivseks, esimene aga lõpetab teenuste osutamise. See tähendab, et omandades aktiivse staatuse, hakkab teine sertifitseerimiskeskus väljastama sertifikaate, neid kehtetuks tunnistama ja vastama sertifikaatide kehtivuspäringutele. Seejuures teise keskuse väljastatavad sertifikaadid jagunevad kaheks:

1. sertifikaadid, mis põhinevad üks-ühele esimese sertifitseerimiskeskuse väljastatud sertifikaatide andmetel viimase hetke seisuga enne ümberlülitumise toimumist;
2. muude andmetega sertifikaadid.

Pakutava lahenduse puhul eeldatakse, et on olemas alternatiivsed teenusepakkujad, kes suudavad osutada samu teenuseid nagu üksik sertifitseerimiskeskus vastavalt praegu Eestis kasutusel olevale eID usaldusmudelile (sertifikaatide väljastamine ja kehtetuks tunnistamine, kehtivuskinnitusteenus). On töenäoline, et praktikas pakuks selliseid teenuseid üks ja sama alternatiivne teenusepakkija. Lähtume eeldusest, et pakutud lahenduse puhul on Eesti riik

sõlminud lepingu sellise alternatiivse teenusepakkujaga, kes on suuteline esimese sertifitseerimiskeskuse äralangemisel võtma ise aktiivse sertifitseerimiskeskuse rolli.

## 4.3. Õiguslik analüüs

### 4.3.1. Õiguslikud küsimused

Uuringu tellija soovib teada, mil määral oleks pakutav lahendus teostatav kehtiva õiguse raamides ja mida tuleks lahenduse kasutuselevõtul kehtivas õiguses muuta või täiendada.

### 4.3.2. Ülevaade kehtiva õiguse raamistikust

Kehtiv õigus sätestab nõuded, mida sertifitseerimiskeskused peavad täitma, et siduda kodanike identiteete krüptograafiliste võtmepaaridega ja tõendada vastavaid seoseid usaldusväärselt.

#### 4.3.2.1. Euroopa Liidu õigus

Kõige üldisemad reeglid tulenevad Euroopa Liidu õigusest, täpsemalt eIDAS artiklist 24, mis kehtestab nõuded kvalifitseeritud usaldusteenuse osutajatele. eIDASe eelkäija, direktiivi 1999/94/EC [10] artikli 12 punkti 11 kohaselt tunnustati "sertifitseerimisteenuste osutajat" kui isikut, kes väljastab sertifikaate või osutab muid elektrooniliste allkirjadega seotud teenuseid. eIDASega võeti vastu üleminekusäte — eIDAS artikkel 51 lõige 3 —, mille kohaselt vastavushindamise läbinud kvalifitseeritud sertifikaate väljastavaid "sertifitseerimisteenuste osutajaid" hakati käsitama eIDASe järgi "kvalifitseeritud usaldusteenuse osutajatenä". Seega eIDASe kehtima hakkamisega kaotati termin "sertifitseerimisteenuste osutaja" kui eraldi õigusmõiste. Seejuures, eIDAS artikli 24 lõigetes 2-4 eristatakse muude kvalifitseeritud usaldusteenuse osutajate seas neid teenuseosutajaid, kes väljastavad kvalifitseeritud sertifikaate. Kui kvalifitseeritud usaldusteenuse osutaja väljastab kvalifitseeritud sertifikaate, siis on tal eIDASe alusel järgmised kohustused:

1. ta peab looma sertifikaatide andmebaasi ja seda ajakohastama (eIDAS artikkel 24 lõige 2 p k));
2. kui ta otsustab sertifikaadi tühistada, siis registreerib ta tühistamise oma sertifikaatide andmebaasis ning avaldab sertifikaadi tühistatud staatuse aegsasti, igal juhul 24 tunni jooksul pärast vastava taotluse saamist. Tühistamine jõustub kohe pärast selle avaldamist (eIDAS artikkel 24 lõige 3);
3. ta peab andma tuginevatele isikutele teavet oma väljastatud kvalifitseeritud sertifikaatide kehtivus- või tühistamisstaatuse kohta. Kõnealune teave tuleb teha kättesaadavaks igal ajal ning pärast sertifikaadi kehtivusaja lõppu vähemalt iga sertifikaadi kohta eraldi automaatsel viisil, mis on usaldusväärne, tasuta ja tõhus (eIDAS artikkel 24 lõige 4).

Erinevate eIDASe alusel antud alamaktidega on omakorda täpsustatud tehnilisi tingimusi, mida kvalifitseeritud usaldusteenuse osutaja peab täitma. Need tuginevad enamasti rahvusvaheliselt kokkulepitud standarditele.

Kvalifitseeritud usaldusteenuse osutaja võib alustada kvalifitseeritud usaldusteenuse osutamist, kui kvalifitseeritud staatus on kantud eIDAS artikli 22 lõikes 1 osutatud usaldusnimekirjadesse (eIDAS artikkel 21 lõige 3). Kvalifitseeritud staatuse usaldusnimekirja kandmine ei tähenda automaatselt, et teenusepakkuja osutab kvalifitseeritud usaldusteenuseid ka praktikas — selleks tuleb sõlmida täiendavalt teenuse osutamise lepingud klientidega, kes neid teenuseid soovivad ostaa. Riikliku

identiteedisüsteemi puhul on niisiis vajalik ka teenuse osutamise lepingu sõlmimine vastava riigiga — seal lepitakse kokku teenuste osutamise periood, täpsemad tingimused ja tasustamine.

Juhime tähelepanu, et digitaalset tuvastamist võimaldavate sertifikaatide väljastajaid ei loeta eIDASe kohaselt usaldusteenuste pakkujateks, s.t. kvalifitseeritud usaldusteenuse osutajaid ja kvalifitseeritud sertifikaatide väljastamist puudutavad eIDASe normid ei kohaldu digitaalsele tuvastamisele. Küll aga tuleb digitaalse tuvastamise puhul täita eIDASe II peatüki reegleid seoses teavitatud e-identimise süsteemide, e-identimise vahendite ja nende usaldusvääruse tasemetega, samuti e-identimise vahendite vastastikuse tunnustamise ning internetipõhise autentimise võimaluse tagamise kohustuse kohta.

#### 4.3.2.2. Eesti õigus

Eesti siseriiklikus õiguses on vähe norme, mis adresseerivad otseselt kvalifitseeritud sertifikaate väljastavaid kvalifitseeritud usaldusteenuse pakkujaid. Need jagunevad peamiselt kolme seaduse vahel:

1. ITDS [11] kehtestab dokumendikohustuse, reguleerib ka digitaalset tuvastamist võimaldava sertifikaadi ja digitaalset allkirjastamist võimaldava sertifikaadi kandmist isikuttõendavasse dokumenti, nende kehtivuse peatamist ja taastamist ning kehtetuks tunnistamist ning isikusamasuse kontrollimise viise (sh masin-masin olukordades). Seejuures ITDS § 9<sup>4</sup> lõige 3<sup>1</sup> näeb ette, et ITDSi alusel välja antud isikut töendavatele dokumentidele kantud sertifikaadiga digitaalset tuvastamist ja digitaalset allkirjastamist võimaldava sertifitseerimisteenuse osutaja on HoS §36 lõike 1 punktis 8 nimetatud elutähta teenuse osutaja.
2. HoS [12] § 37 lõike 2 alusel vastu võetud Ettevõtlus- ja tehnoloogiaministri määrus nr 4 [13] määratleb ITDS [11] § 9<sup>4</sup> lõikes 3<sup>1</sup> viidatud sertifitseerimisteenuse osutaja pakutava kehtivuskinnitusteenuse kui elutähta teenuse ning sätestab nõuded sellise teenuse toimepidevusele.
3. EUTS reguleerib e-identimist ja e-tehinguteks vajalikke usaldusteenuseid ning riikliku järelevalve korraldust ulatuses, milles need ei ole reguleeritud eIDASes, sh kehtestab täiendavad nõuded kvalifitseeritud usaldusteenuse osutajale ja usaldusteenuse osutamisele (EUTS § 5), täpsustab sertifikaadi kehtivusaja algust ja lõppu, sertifikaadi kehtivuse peatamise ja taastamise ning sertifikaadi kehtetuks tunnistamise korda (EUTS §§ 16-21). Samuti, eIDAS artikkel 51 lõikest 3 tulenevalt loetakse EUTS § 25 lõike 3 järgi EUTSi jõustumise hetkel sertifitseerimise registrisse kantud "sertifitseerimisteenuse osutaja" kvalifitseeritud usaldusteenuse osutajaks eIDAS artikli 51 lõikes 3 sätestatud eritingimust arvestades. Siin tuleb aga tähele panna, et kuna autentimine ega digitaalne tuvastamine ei ole eIDASe mõistes usaldusteenus, siis ei saa eIDAS ega selle reegleid täiendav EUTS kohalduda digitaalset tuvastamist võimaldavale sertifikaadile, vaid üksnes digitaalset allkirjastamist võimaldavale sertifikaadile. Ka L. Kask ja K. Laanest on asunud seisukohale, et EUTS ja eIDAS ei reguleeri isikutuvastamist võimaldavat sertikaati ning selle peatamist, taastamist või kehtetuks tunnistamist, sest see allub ITDSile [14].

#### 4.3.3. Üldised õiguslikud nõuded soovitusliku eID usaldusmudeli toimimiseks

Aruande eelmises peatükis välja pakutud soovitusliku eID usaldusmudeli kasutuselevõtmisel ei muutu põhimõte, et Eesti riigile annab sertifikaatide väljastamise ja kehtetuks tunnistamise ning

kehtivuskinnituse teenust üks sertifitseerimiskeskus korraga. Seni, kuni esimene sertifitseerimiskeskus suudab neid teenuseid pakkuda, toimib eID usaldusmudel ka pakutava lahenduse puhul samamoodi nagu praegu. Järelikult saab pakutava lahenduse puhul kehtivat õigust samamoodi edasi rakendada, vähemalt kuni teisele sertifitseerimiskeskusele ülemineku hetkeni.

Juhime tähelepanu, et sertifitseerimisteenuseid ja sertifitseerimiskeskuse tegevust puudutav kehtiv õigus ei ole probleemideta. Pole välisstatud, et kehtivas õiguses tuleb teha parandusi ja täiendusi sõltumata sellest, kas antud uuringu raames välja pakutud soovituslik eID usaldusmudel võetakse praktikas kasutusele või mitte. Seetõttu ei ole varem tõstatatud kehtiva õiguse probleeme antud uuringus eraldi käsitletud, kuid need vajaksid soovitusliku eID usaldusmudeli kasutuselevõtu korral põhjalikku läbitöötamist ja lahendamist. Näiteks üks varem tõstatatud õigusprobleemidest, mis mõjutab ka soovituslikku eID usaldusmudelit, puudutab sertifikaatide masstühistamise, -peatamise ja -uuendamise reeglistikku — siin on tuvastatud puudujääke, mis võivad põhjustada törkeid sertifikaatide masstühistamisel ja massväljastamisel, kui on vaja kiiresti ümber lülituda ühelt aktiivselt sertifitseerimiskeskuselt teisele. [15, 14]

Pakutava lahenduse peamine erinevus võrreldes praegu kasutusel oleva eID usaldusmudeliga seisneb selles, et sertifitseerimiskeskuse rolli saavad täita kaks erinevat teenusepakkujat ajalises järgnevuses, s.t mitte samaaegselt, kusjuures ühelt teisele ülemineku hetk ei ole ette kokku lepitud ning võib juhtuda igal ajal ootamatult. Kui soovitakse pakutavat lahendust kasutusele võtta, siis tuleks seda erinevust täpsemalt reguleerida, eelkõige kehtestada täiendavaid reegleid lisaks kehtivale õigusele. Antud õigusliku hinnangu raames oleme kaardistanud esmase loetelu nõuetest, mis tuleks pakutava lahenduse kasutuselevõtmisel õiguslikult lahendada:

1. tuleb määratleda, milliste asjaolude saabumisel muutub aktiivset staatust omav esimene sertifitseerimiskeskus passiivseks. Lisaks tuleks täpsustada, kui kaua need asjaolud peavad kestma, et esimesel sertifitseerimiskeskusel tekiks kohustus aktiivsest staatusest loobuda või riigil tekiks õigus aktiivne staatus üle anda teisele sertifitseerimiskeskusele.
2. tuleb selgelt teineteisest eristada aktiivse ja passiivse sertifitseerimiskeskuse rollid (õigused ja kohustused) ning määratleda need ajakohastes õigusaktides või lepingutes. Näiteks:
  - a. täpsustada aktiivse ja passiivse sertifitseerimiskeskuse omavahelise sõltumatuse tingimused, mh kummagi rolli täitjad võivad olla seotud valitseva mõju kaudu. Samuti, kas kas riik võib kummagi rolli täitja leidmiseks korraldada ühe riigihanke, nii et esimene ja teine sertifitseerimiskeskus osalevad hankes ühispankumusega või peab mõlemal juhul korraldama eraldi konkursid.
  - b. panna passiivsele sertifitseerimiskeskusele kohustus hoida aktiivse sertifitseerimiskeskuse sertifikaatide andmebaasi koopiat ja tagada selle ajakohasus (kui vastavat ülesannet ei saa anda riigile).
  - c. sätestada või kokku leppida, kas aktiivse staatuse kaotamisel leping Eesti riigi ja esimese sertifitseerimiskeskuse vahel lõppeb või kehtib edasi osaliselt.
  - d. määratleda, kas esimese sertifitseerimiskeskuse teenuste taastumisel tuleb aktiivne staatus talle tagasi anda (teine sertifitseerimiskeskuse tegutseb ajutise puhverlahendusena) või mitte (teine sertifitseerimiskeskus võtab aktiivse kohustuse üle ja täidab seda kuni Eesti riigiga sõlmitud lepingu lõppemiseni, misjärel esimene sertifitseerimiskeskus võib taas kandideerida aktiivsele staatusele uue lepingu alusel).

- e. kehtestada mõlema keskuse puhul tasu saamise õigus.
- 3. aktiivse sertifitseerimiskeskuse roll peab olema kiiresti üleantav ühelt teenusepakkujalt teisele, kellega riik on sõlminud vastava lepingu selle rolli täitmiseks. See tähendab, et riigil tuleb kehtestada ümberlülitumiseks vajalikud reeglid ja nõuded nii aktiivsele kui ka passiivsele sertifitseerimiskeskusele, et tagada vajalikud ettevalmistused ja ümberlülitumise plaanid.
- 4. tuleb selgelt määratleda, mis hetkest alates läheb aktiivse sertifitseerimiskeskuse roll üle alternatiivsele teenusepakkujale, sh kas see on automaatne üleminek või nõuab eraldi otsustuskorda.
- 5. tuleb määratleda, kuidas avalikustatakse info aktiivse sertifitseerimiskeskuse rolli ülemineku kohta, nii et eID kasutajatel ja tuginevatel isikutel oleks võimalik sellega igal ajal tutvuda.
- 6. tuleb kehtestada reegel, et alternatiivne teenusepakkija võib sertifitseerimiskeskuse teenuseid pakkuda Eesti riigile alates hetkest, mil ta on omandanud aktiivse sertifitseerimiskeskuse rolli, s.t muul ajal ei ole alternatiivsel teenusepakkual väljastatud sertifitseerimiskeskuse teenuseid osutada, v.a aktiivse sertifitseerimiskeskuse sertifikaatide andmebaasi koopia hoidmine ja selle ajakohasuse tagamine.
- 7. esimese sertifitseerimiskeskuse poolt väljastatud sertifikaatide kehtivusinfo haldamiseks vajalike andmete jagamine teise sertifitseerimiskeskusega peab olema lubatud nii enne aktiivse sertifitseerimiskeskuse rolli üleandmist kui ka pärast seda. Mõlemal juhul tuleb kehtivusinfo haldamiseks vajalike andmete kaitseks valida sobivad meetmed, arvestades et teisel keskusel on eelkõige vaja neid andmeid täielikult näha ja töödelda aktiivse sertifitseerimiskeskuse rolli ülevõtmiseks ja täitmiseks, aga mitte muul juhul.

#### **4.3.4. Hinnang üldiste õiguslike nõuete teostatavusele kehtiva õiguse alusel**

Eesti riigi väljastatud isikut tööndavates dokumentides sisalduva digitaalset allkirjastamist võimaldava sertifikaadiiga peab olema võimalik anda kvalifitseeritud elektroonilisi allkirju, mis on võrdsed omakäelise allkirjaga (eIDAS artikkel 25 lõige 2). See tähendab, et vähemalt digitaalset allkirjastamist võimaldav sertifikaat peab vastama e-allkirjade kvalifitseeritud sertifikaadi nõuetele, mis on ette nähtud eIDAS artiklis 28 ja lisas I. Digitaalset tuvastamist võimaldav sertifikaat ei allu eIDASele, mistõttu siin on rohkem vabadust siseriiklikul tasandil uusi nõudeid kehtestada, kuid samal ajal peab olema tagatud digitaalset tuvastamist võimaldavat sertifikaati sisaldaava e-identimise vahendi kõrge usaldusvääruse tase vastavalt eIDASe artiklitele 6 ja 8.

Kõiki kvalifitseeritud sertifikaate, sh e-allkirjade kvalifitseeritud sertifikaate saavad eIDASe kohaselt väljastada üksnes kvalifitseeritud usaldusteenuse pakkujad. Järelikult saavad antud uuringu eelmises peatükis välja pakutud soovitusliku eID usaldusmudeli aktiivse ja passiivse sertifitseerimiskeskuse teenuseid pakkuda ainult kvalifitseeritud sertifikaate väljastavad kvalifitseeritud usaldusteenuse osutajad vähemalt senikaua, kuni Eesti riik ostab sertifitseerimisteenuste osutamist korraga ühelt ja samalt teenusepakkujalt nii digitaalse allkirjastamise kui ka digitaalset tuvastamise võimaldamiseks. Kvalifitseeritud usaldusteenuse osutajate staatust ja selle usaldusnimekirjas ajakohasena hoidmist, samuti kvalifitseeritud sertifikaatide väljastamist, kehetetuks tunnistamist ja kehtivuspäringutele vastamist on kehtivas õiguses piisavalt reguleeritud ning ainuüksi soovitusliku eID usaldusmudeli kasutuselevõtmiseks ei ole siin töenäoliselt vajalik muudatusi teha. Küll aga vajaks täpsustamist sertifikaatide masstühistamise, -peatamise ja -uuendamise reeglistik — puudujäägid selles reeglistikus mõjutavad sertifikaatide masstühistamist ja massväljastamist ühelt aktiivselt sertifitseerimiskeskuselt

ümberlülitumisel teisele.

Kehtiva õiguse kohaselt ei ole piiratud, mitut sertifitseerimiskeskust võib riikliku identiteedisüsteemi pidamiseks kasutada. Niisiis pole välistatud, et ühes ja samas identiteedisüsteemis eksisteerib mitu teenusepakkujat, kellel on teatud ajaperioodil õigus pakkuda riigile sertifitseerimiskeskuse teenuseid. Samas ei käsitle kehtiv õigus küsimusi sellest, kas ja kuidas toimub sertifitseerimiskeskuse teenusepakkuja vahetus, kui hetkel teenuseid osutav pakkija ei suuda ootamatult oma sertifitseerimiskeskuse kohustusi täita. Neid küsimusi reguleeritakse hetkel tõenäoliselt Eesti riigi ja sertifitseerimiskeskuse rolli täitvate äriühingute vahelistes lepingutes, mille sisu ei ole aruande koostajatele teada.

Kui Eesti riik otsustab kasutusele võtta aruande eelmises peatükis välja pakutud soovitusliku eID usaldusmudeli, siis on tõenäoliselt vajalik sertifitseerimiskeskuse teenusepakkuja vahetusega seotud küsimusi õigusaktides täpsemalt reguleerida ning ei saa piirduda üksnes lepinguliste õigussuhetega. Seda eelkõige põhjusel, et vahetuse protsess peab olema läbipaistev ning usaldusväärne nii eID kasutajatele kui ka tuginevatele isikutele ja ühiskonnale tervikuna, tagamaks usalduse säilimine Eesti riikliku identiteedisüsteemi suhtes. Põhjalikult tuleks läbi möelda nii teenusepakkuja vahetuse eeldused, õiguslikud alused kui ka tagajärjed ning leida sobivad õiguslikud mehhanismid nende realiseerimiseks.

Niisiis tuleks enamus sertifitseerimiskeskuse teenusepakkuja vahetusega seotud reeglistikust luua nö valgelt lehelt, arvestades lisaks eID valdkonna normidele ka erinevaid muude õigusvaldkondade norme (riigiõigus, andmekaitseõigus, konkurentsõigus, küberkaitseõigus, riigihangete õigus, haldusõigus, lepinguõigus, korrakaitseõigus jne). Leiame, et kehtivas õiguses ei ole põhimõttelisi piiranguid sellise reeglistiku väljatöötamiseks, jõustamiseks ja rakendamiseks, kuid see vajab täiendavaid õiguslikke analüüse, mis viiakse läbi erinevate õigusvaldkondade ekspertide vahelises koostöös.

# 5. Kokkuvõte

Käesolevas aruandes analüüsime sertifitseerimistaristu kui usaldusteenuse toimimispõhimõtteid, kirjeldasime usalduse edasikandumist selles taristus ja andsime ülevaate juhtunud intsidentidest, mis on selle taristuga seotud. Analüüs põhjal pakkusime välja sertifitseerimistaristu ülesseadmise viisi, kus nõrkasid lülisid oleks varasemast vähem, kuid mis siiski sobituks oluliste seaduste ja tehniliste standarditega. Aruande analüüsiosa esitavates lisades on toodud varem koondamata teadmus, need võivad olla oluliseks sisendiks sertifitseerimistaristu toimimise aluste reguleerimisel tulevikus.

- Lisas A esitatud sertifitseerimiskeskustega seotud intsidentide loetelule ja analüüsile me põhjalikkuse poolest analooge ei tea. Ligi kahte aastakümmet kattev loetelu põhineb suurel hulgal turvateavitustel, intsidendiaruannetel, blogi- ja foorumipostitustel. Sageli on viited neile teadetele ja postitustele aegunud, seega tuli neid otsida veebiarhiivist *Wayback Machine*.
- Lisas B esitatud usaldusmudelite kirjelduse muudab unikaalseks väljapakutud graafiline "keel", milles nende mudelite usalduseeldused ja tulemid kirjeldatud on. Kasutades nende kõigi jaoks ühte ja sama keelt, on mudeleid lihtsam omavahel võrrelda ja aru saada nende sarnasustest ja erinevustest. Meie kasutatava graafilise keele alused lähtuvad matemaatilisest loogikast, seega saame olla kindlad, et keele elemente kombineerides ei jõua me vasturääkivusteni.

Me pakume välja sertifitseerimistaristu konstruktsiooni, mis meie arvates on nii heade omadustega, mida on üldse võimalik saavutada, minemata seejuures vastuolu olemasolevate tehniliste standarditega ja ajamata süsteemi ülesseadmise ja käitamise kulusid vastuvõetamatult suureks. Meie lahendus põhineb süsteemi osade dubleerimisel, kusjuures me püüame dubleeritavaid osasid võimalikult väiksena hoida. Sertifitseerimistaristu võib olla nõrgaks lüliksi nii tervikluse kui ka käideldavuse seisukohast. Konstruktsiooni käigus veendusime, et kasutuses olevad tehnilised standardid muudavad parema terviklusega süsteemi väljapakkumise, kus tervikluse kasv on saavutatud tänu süsteemi teatud osade dubleermisele, praktiliselt võimatuks. Meie lahendus parandab süsteemi käideldavust, kuid ka siin on tehniliste standarditega vastavuse saavutamine mittetriviaalne. Seadusandlus samas nii suuri kitsendusi ei tundu loovat.

Pakutud konstruktsioon on potentsiaalne lahendus probleemile, kus kahe aktiivse sertifitseerimiskeskuse pidamine on liiga kallis. Samas praktikas ei pruugi passiivses olekus sertifitseerimiskeskuse ülalpidamine tingimata olla (oluliselt) odavam. Sellisel juhul võib mõistlikumaks osutuda lepinguliste suhete loomine kahe erineva aktiivse sertifitseerimiskeskusega, mis vaikimisi tegutsevad eri turusegmentides, näiteks pakuvad eri teenuseid või on aktiivsed eri riikides / turgudel. Kumbki keskus on siis teise keskuse suhtes passiivses rollis (on valmis tema töö vajadusel üle võtma), pakkudes samal ajal *aktiivselt* oma teenuseid.

Meie uuringust tuleb ka välja, et isegi kui meil on mitu sõltumatut sertifitseerimiskeskust, kuid on endiselt ühine registreerimiskeskus (seda rolli täidab Eestis praegu PPA), siis on see registreerimiskeskus omakorda nõrk lüli. Seos võib olla ka kaudne: näiteks kui ID-kaardi olemasolust piisab Smart-ID saamiseks, siis oleks ID-kaardi väljastaja nõrk lüli ka siis, kui Smart-ID kasutab ID-kaardist erinevat sertifitseerimiskeskust. Teorias on lihtne eeldada, et igal sertifitseerimiskeskusel on oma sõltumatu registreerimiskeskus, kuid tarvis oleks eraldi uuringut, mis pakuks välja, kuidas seda praktikas mugavamalt saavutada.

Valminud aruanne võib olla edasiste arendustööde plaanimise aluseks. Pikemas perspektiivis

toetab ta sertifitseerimist puudutavate tehniliste ja organisatsiooniliste regulatsioonide väljatöötamist.

# 6. Summary in English

In this report, we have given an analysis of the operating principles of certification infrastructure as a trust service, described the transfer of trust in this infrastructure, and gave an overview of incidents that have taken place in this infrastructure. Based on the analysis, we proposed a set-up for certification infrastructure that has a smaller number of single points of failure, but still complies with the significant legislative acts and technical standards. The analytic part of this report systematizes existing knowledge in a novel manner, which could serve as an important input to the future regulations on operating principles of certification infrastructure.

- Appendix A presents a list and an analysis of incidents related to certification authorities. We are aware of no other treatments of the same topics with at least similar thoroughness. The list, covering almost two decades, is based on a large number of security announcements, incident reports, blog and forum posts. The original links to these announcements and posts have often become stale, hence they had to be found with the help of the Wayback Machine.
- Appendix B describes the existing trust models in a unified manner, including their trust assumptions and outcomes. The graphical "language" defined and used for this description is novel. It simplifies the comparison of the models, recognizing their similarities and differences. Our graphical language is founded in mathematical logic, giving us assurance that it contains no internal contradictions.

In the main part of this report, we propose a construction of certification infrastructure offering the best possible set of integrity and availability properties one can achieve without running afoul of existing technical standards or unreasonably raising the costs of setting up and running the system. Our solution is based on replicating (doubling) some of the components of the system, keeping the doubled part as small as possible. A certification infrastructure may contain single points of failure of both integrity and availability. While constructing our proposed system, we found that the existing technical standards more-or-less exclude the increase of the integrity of the system through the doubling of certain components. Our solution improves the availability of the system, even though the adherence to existing standards is still non-trivial. Compared to technical standards, the adherence to existing legislative acts was much easier to achieve.

While we double certain components of the certification infrastructure, we try to keep costs down by stating that one instance of certain doubled components may be "passive", i.e. dormant until something happens with the active component. It is possible that the maintenance of a passive certification authority is not much cheaper than the maintenance of an active one. In this case, it may make more sense to have contractual relationships with two different certification authorities, normally active in different market segments, e.g. in different countries, or offering different services. Each of the two authorities is in the *passive* role with respect to the other one (i.e. is prepared to take over the operations of the other one), while *actively* offering its own services.

Our study also shows that if several independent certification authorities make use of a single, common registration authority (in Estonia, the Police and Border Guard Board performs this role), then this registration authority is a single point of failure. The relationship may be indirect: for example, if the possession of an ID-card is sufficient for obtaining Smart-ID, then the issuer of ID-cards is a single point of failure even if Smart-ID makes use of a different certification authority. It is easy to make the theoretical assumption of each certification authority having an independent registration authority, but additional studies may be needed how to conveniently obtain this in

practice.

The current report may serve as basis for planning subsequent development of trust services. Long-term, it supports the development of technical and organizational regulations for these services.

# A. Appendix: Incidents through failing CAs

## A.1. Single point of failure cases of PKI

Certificate authorities (CA) in the Public Key Infrastructure (PKI) might be considered as a single point of failure (SPoF) due to the amount of trust given to them. In this section, we list historical cases of crises and incidents related to CAs. All information in this chapter is based on a public resources, often blog or newsgroup posts, referenced throughout the text.

### A.1.1. VeriSign and Microsoft (2001)

#### A.1.1.1. Description of the incident

VeriSign is a U.S. based company founded in 1999. They are operating as a Domain Name System (DNS) registrar, but have also been providing certificate authority services in the past, until their authentication business unit [was acquired by Symantec](#).

On January 30-31, 2001, someone claiming to represent Microsoft deceived VeriSign employees and made them issue two Class 3 code-signing certificates [16]. VeriSign discovered the incident via email feedback loop from Microsoft several weeks later [17] and shared it publicly in the end of March after Microsoft's mitigation measures were put in place.

The issuance of these certificates affected almost every Microsoft OS user at the time (Windows 95/98/Me/2000/NT 4.0). Although the certificates stated that they were owned by Microsoft, the code and applications signed by these certificates would not have been trusted by default and would raise a warning when attempted to execute. A warning should have caused distrust by a regular user, but because the certificates were issued under the name "Microsoft Corporation", the user might have been misled to run the code signed by fraudulent certificates.

#### A.1.1.2. The reasons and mitigation

At the moment of writing this report, there is no public information about how the malicious party managed to pose themselves as a Microsoft employee to VeriSign. To perform such an attack, the attacker would need establish a fake identity, convince several employees of VeriSign, and pay the certificate fees of about 400\$ for each of them [18].

Once the fraudulent issuance was discovered, VeriSign revoked the certificates and added them to VeriSign's Certificate Revocation List (CRL). However, it was not possible for any browser's CRL-checking mechanism to locate and use the VeriSign CRL, because VeriSign's code-signing certificates did not specify a CRL Distribution Point (CDP) [16]. Such situation forced Microsoft to issue a software update which blacklisted the fraudulent certificates.

It is difficult to estimate how many users became victims of the attacks executed using these certificates.

More information about the incident can be found in the following resources [16, 19, 17, 20, 18].

## A.1.2. Thawte (2008)

### A.1.2.1. Description of the incident

Thawte is a certification authority which was founded in 1995 in South Africa. On July 2008, white hacker and security researcher Mike Zusman bypassed domain validation and managed to obtain a fraudulent certificate for the domain [login.live.com](#) [18]. Zusman did not use this certificate in any malicious way and reported the vulnerability without revealing the name of the CA [21, 22]. Later, he publicly demonstrated the exploit details in his DEFCON 17 presentation [20].

### A.1.2.2. The reasons and mitigation

Thawte used email for domain name validation. Zusman found out that Thawte recognized a large list of aliases for domain confirmation purposes. Anyone could register a [@live.com](#) account at that time; hence he created [sslcertificates@live.com](#), which Thawte accepted during domain validation. Subsequently, Zusman requested and obtained the certificate for [login.live.com](#).

More information about the incident can be found in the following resources [21, 22, 20, 18].

## A.1.3. StartSSL (2008)

### A.1.3.1. Description of the incident

StartSSL was a certificate authority that belonged to the Israeli company StartCom. They offered introductory SSL certificates for no cost.

On 20th December 2008, security researcher Mike Zusman registered an account at StartSSL and manipulated web traffic between the StartSSL website and the web browser, leading to the exploitation of the domain validation interface [23]. Zusman caused fraudulent certificates for the domains [verisign.com](#), [phishme.com](#), [paypal.com](#), and [intrepidusgroup.com](#) to be issued. After the issuance of the certificate for [verisign.com](#), it got flagged for review by StartSSL's secondary control system for high-profile domains. The system had determined the fraudulence of the initial request and StartSSL revoked all certificates issued to Zusman. Subsequently, Zusman cooperated with employees of StartSSL to investigate and fix the bug in the system. The whole incident was discovered and the flaws were fixed within fourteen hours.

### A.1.3.2. The reasons and mitigation

The main cause for this incident was a vulnerability in the web front-end interface of StartSSL. The interface was unconditionally trusting user input, allowing domain validation emails to be sent to arbitrary email addresses at unrelated domains. Using local client proxy, Zusman managed to send all domain validation emails to his personal email. Only the fraudulent issuance of [verisign.com](#) triggered StartSSL's policy of human checks for high-value domains, thereby noticing the attack, and the employees were able to take preventative action within minutes. Consequent modifications and analysis of the attack vector lead to the complete removal of used vulnerability [24].

During this incident, no damage was done to any relying party. This attack did not result in any criminal charges being filed.

More information about the incident can be found in the following resources [24, 23, 20, 18].

## A.1.4. Comodo and CertStart (2008)

### A.1.4.1. Description of the incident

Comodo CA was a certificate authority that was offering SSL, email security, and code signing certificates. In 2017, it was re-branded and now the company is known as Sectigo.

CertStart was a Danish company and Comodo's reseller. In December 2008, Eddy Nigg, CEO of StartCom, reported on his personal blog [25] that CertStart was not performing any domain control validation. Exploiting this vulnerability, Nigg obtained a SSL server certificate for the domains [www.mozilla.com](http://www.mozilla.com) and [startcom.org](http://startcom.org). The certificate appeared valid in the Firefox browser at that moment.

### A.1.4.2. The reasons and mitigation

The main cause for this incident was Comodo CA trusting their resellers, including CertStart to perform domain control validation, instead of performing it themselves. Comodo CA investigated this vulnerability, ceased CertStart's ability to issue certificates, and revoked the certificate for [mozilla.com](http://mozilla.com) [26].

There are no known cases that malicious parties exploited this vulnerability.

More information about the incident can be found in the following resources [25, 27, 28].

## A.1.5. VeriSign (2010)

### A.1.5.1. Description of the incident

According to their October 2011 quarterly report [29], VeriSign suffered series of security breaches throughout 2010 that were not sufficiently reported to upper management at the moment of the breaches. During these attacks against VeriSign's internal network, a malicious party gained access to a small piece of information on their computers and servers. VeriSign claimed that their employees had investigated those breaches, and the company did not believe any malicious actor managed to access their DNS server. However, VeriSign made no claims that their CA infrastructure was affected or unaffected.

### A.1.5.2. The reasons and mitigation

At the moment of writing this report, there is no public information about what were the exact assets under attack, or how did the malicious party manage to perform the underlying attack. VeriSign stated that their information security group had implemented remedial measures designed to mitigate the attacks' impact and detect future similar attacks [29].

More information about the incident can be found in the following resource [29].

## A.1.6. Comodo (2011)

### A.1.6.1. Description of the incident

On March 15th, 2011, one of Comodo's registration authorities (RA) suffered an attack that gave a

malicious party access to the RA user's account [30]. The intruder, using the compromised account, caused 9 certificates for the domains [mail.google.com](#), [www.google.com](#), [login.yahoo.com](#) (3 certificates), [login.skype.com](#), [login.live.com](#), [addons.mozilla.org](#), and the name "Global Trustee" to be issued. There was no evidence that any of the issued certificates were received by a malicious party, except for one of the certificates issued for [login.yahoo.com](#), which was spotted on a web server in Iran. Several hours after the fraudulent issuance, all these certificates were revoked and Comodo immediately informed browser providers and web domain owners of the security breach. On March 23rd, the incident was revealed to the public [30].

On March 31st, Comodo noticed another attempt of attack on another of their RA-s. However, new security measurements detected and stopped the intruder from malicious actions.

#### A.1.6.2. The reasons and mitigation

The main cause for this incident was Comodo CA still trusting their resellers to perform domain control validation, even after the CertStart incident in 2008. After these attacks, Comodo suspended the practice of trusting resellers to perform domain validation. An individual going by the pseudonym *ComodoHacker* had taken credit for the underlying attack in a Pastebin post [31]. The hacker stated that they acquired complete access to the RA network and reverse-engineered the C# dynamic library [TrustDll.dll](#) that took care of signing certification requests. This DLL file had hardcoded usernames and passwords that provided access to the certificate signing API. Using this API, the attacker generated his own certificate signing requests (CSR) to generate the listed 9 domain certificates.

As Comodo stated [30], their CA infrastructure and the keys in their hardware security modules (HSM) were not compromised after neither of these attacks.

More information about the incident can be found in the following resources [30, 31].

### A.1.7. StartSSL (2011)

#### A.1.7.1. Description of the incident

On June 15th, 2011, StartSSL infrastructure suffered from an attack, and their servers were compromised to fraudulently obtain certificates. According to Eddy Nigg, CEO of StartCom, the attacker was aiming to issue certificates for Google, Twitter, and Yahoo domains [32]. The hacker was unable to cause any certificates to be issued, due to the fact that StartSSL's private signing key was stored on the computer that was not connected to the Internet [33]. After this incident, StartSSL suspended the issuance of new certificates and related services for several weeks.

#### A.1.7.2. The reasons and mitigation

At the moment of writing this report, there is no public information on how exactly the malicious party managed to perform the underlying attack. The person *ComodoHacker* claimed credit for this incident. In a Pastebin post [34], they stated:

StartCom was lucky enough, I already connected to their HSM, got access to their HSM, sent my request, but lucky Eddy (CEO) was sitting behind HSM

and was doing manual verification.

Eddy Nigg confirmed this in the interview to *InformationWeek (DarkReading)*, although he did not reveal any additional details about the incident [35]. As no certificates were issued, no harm nor damage was done to any party, except for the temporary suspension of StartSSL.

More information about the incident can be found in the following resources [33, 32, 34, 35].

## A.1.8. DigiNotar (2011)

### A.1.8.1. Description of the incident

DigiNotar was a Dutch CA company that provided certificate authority services. They were also managing the Dutch e-government PKI system *PKIoverheid*.

In mid-2011, DigiNotar had a security breach that resulted in the issuance of fraudulent certificates [36]. According to Fox-IT report [37], using vulnerabilities in software installed in DigiNotar premises, an attacker obtained access to the company's demilitarized zone (DMZ) on the 17th of June 2011. On July 1st, the intruder obtained access from the DMZ to DigiNotar's certificate authority servers by the means of tunneling through other segments of DigiNotar's network. The attacker issued the first set of 128 fraudulent certificates on July 10th, and subsequently at least 531 more for 53 unique common names. Due to the scale of the attack and the tampering of DigiNotar's logs, the actual number of fraudulent certificates remains unknown.

In July 19th, DigiNotar performed their routine check and discovered that several fraudulent certificates had been issued. By the end of July, DigiNotar had patched all noticed vulnerabilities and revoked a number of fraudulent certificates that they were aware of after the routine check. Unfortunately, DigiNotar neglected to share any information about the incident to the public or to any vendors, assuming that the incident had been contained.

In August 27th, the incident became public due to an Iranian Gmail user with the nickname *abibo*. They reported that they could not access Google's services due to an uncommon certificate error in Google Chrome. It turned out that the error was caused by a Man-in-the-Middle (MitM) attack that used fraudulent DigiNotar certificates. Since Google had used certificate pinning in its Chrome web browser, the latter detected the attack and prevented it by denying access to Google's services. Later it became clear that the problem reported by *abibo* was a part of a country-scale MitM attack which affected around 298,140 users in Iran, who had been intercepting Google's certificates installed on their devices and issued by DigiNotar.

By August 29th, DigiNotar had revoked fraudulent certificates that they were aware of. Major browser vendors disabled DigiNotar's root certificate [38, 39, 40]. The Dutch government commissioned the security company Fox-IT to investigate the whole incident.

### A.1.8.2. The reasons and mitigation

According to the initial report by Fox-IT [41], an unpatched piece of software installed on DigiNotar's servers, poor network segmentation, weak passwords, and the absence of malware and intrusion detection software were the main causes of the security incident. *ComodoHacker*, who had taken responsibility for two other attacks in 2011, took credit for this incident in the Pastebin

post [42]. They provided several pieces of evidence that they were responsible for this attack. A particularly strong piece of evidence was a binary file signed with one of the fraudulent certificates. Other described details about the attacks also matched the official report. It is not yet known if *ComodoHacker* is responsible for MitM attacks in Iran.

In the end, all DigiNotar's root certificates were revoked and subsequently DigiNotar declared bankruptcy in September 2011.

More information about the incident can be found in the in the following resources [41, 37, 36].

## A.1.9. GlobalSign (2011)

### A.1.9.1. Description of the incident

GlobalSign is a Belgian certificate authority. On September 6th, 2011, *ComodoHacker* claimed in a Pastebin post that they had managed to breach GlobalSign's CA systems [34]. GlobalSign took this claim seriously and temporarily halted new certificate issuance between 6th and 15th of September [43, 44]. Additionally, they hired the security firm Fox-IT to investigate the potential incident. As reported [45], the attacker breached a peripheral web server, located outside the certificate issuance infrastructure. That lead to the espousal of publicly available HTML pages, publicly available PDFs, and the SSL certificate and the key issued to the website [www.globalsign.com](http://www.globalsign.com).

### A.1.9.2. The reasons and mitigation

There is no evidence that any fraudulent certificates were issued, customer data was exposed, or GlobalSign certificate issuance infrastructure was compromised during the breach. Nevertheless, GlobalSign decided to implement additional controls around their infrastructure, customer data protection and access to their systems [45].

As no certificates were issued, no harm nor damage was done to any party, except for the temporary suspension of GlobalSign.

More information about the incident can be found in the in the following resources [45, 43].

## A.1.10. Pos Digicert Sdn.Bhd. (2011)

### A.1.10.1. Description of the incident

Pos Digicert Sdn.Bhd. (Digicert Malaysia) is a certificate authority based in Malaysia, with no relation to DigiCert Inc. based in Utah, U.S. Pos DigiCert Sdn.Bhd. was an intermediate CA of Entrust and Verizon. In November 2011, it issued around 22 weak certificates to the Malaysian government [46]. This incident was noticed because the private key for one of the 512-bit key certificates had been obtained by an attacker, and used to sign malware. This malware was used in a spear-phishing attack against the Asia-Pacific office of another CA company [47].

### A.1.10.2. The reasons and mitigation

Pos Digicert Sdn.Bhd. issuance procedures were in violation of accepted CA standards. The 22 weak certificates had:

- **Cryptographically weak certificate keys.** Pos Digicert Sdn.Bhd. issued certificates which had RSA keys with the length of only 512 bits. The keys of such length do not provide the required security level and can be easily factored with modern computing devices [48, 49].
- **Missing revocation information.** Pos Digicert Sdn.Bhd. did not include any revocation information in these certificates. Hence there was no reliable way to revoke the certificates.
- **Missing usage restrictions.** Some of the issued certificates were missing the usage restrictions in the Extended Key Usage (EKU) field. Pos Digicert Sdn.Bhd. was only allowed to issue certificates to websites, due to their contract with Entrust. However, the missing usage restrictions could cause certificates to be used for any purpose (including malware code signing).

When this incident had become public, Entrust and Verizon revoked their intermediate certificate and informed the browser vendors [50]. Mozilla revoked the trust in all certificates issued by DigiCert Sdn.Bhd. [51]. Moreover, DigiCert Inc. had to publish a press release, stating that there was no affiliation between DigiCert Inc. and Pos Digicert Sdn.Bhd. [52].

More information about the incident can be found in the following resources [46, 47, 50, 51, 18].

## A.1.11. Trustwave (2012)

### A.1.11.1. Description of the incident

Trustwave is an American company that provides network and internet security services. They also have a division SecureTrust which operates as a certificate authority. In February 4th, 2012, Trustwave stated that they had issued a subCA certificate to one of their internal corporate network customers [53]. This organisation did not provide CA services. It could use the subCA certificate to issue arbitrary certificates without any regulations, or monitor its workers' HTTPS traffic by performing a MitM attack.

### A.1.11.2. The reasons and mitigation

The issuance of subCA certificate can be used by the holder of the certificate to set up an interception proxy. This can ease the deployment process, because existing endpoints would already trust these certificates that chain to a trusted root [54].

Trustwave decided to revoke the contradictory certificate and promised that they would issue no similar certificates in the future [53]. However, Trustwave's behaviour caused a discussion whether Trustwave certificates should be trusted by Mozilla [55, 56]. After much discussion, it was decided that only the offending subCA certificate would be distrusted. Mozilla stated that they would not tolerate the issuance of subordinate CA certificates for the purposes of SSL man-in-the-middle interception, and asked all CAs to halt such practices [57]. The Trustwave root certificates were not removed.

More information about the incident can be found in the following resources [53, 55].

## A.1.12. Cyberoam (2012)

#### A.1.12.1. Description of the incident

Cyberoam is an Indian device manufacturer selling deep packet inspection (DPI) products. On July 3rd, 2012, Tor Project reported that one of Tor users from Jordan had come upon a fraudulent certificate for website [torproject.com](#) [58]. Originally, maintainers of Tor Project considered that a certificate authority (CA) might have been compromised. After initial investigation, however, they found a security vulnerability—all Cyberoam DPI devices shared the same CA certificate and had the same private key. Hence the user who had reported the problem was not witnessing an attack, but rather the interception of the SSL connection to Tor by a Cyberoam DPI device. The discovered vulnerability allowed traffic interception from any user of a Cyberoam device to any other Cyberoam device, or to a device with the common private key.

#### A.1.12.2. The reasons and mitigation

At the moment of writing this report, it is not known why Cyberoam had such key and certificate management practices. After this vulnerability was discovered, Tor Project suggested the potential victims to uninstall the Cyberoam CA certificate from their browsers, and decline any connection if a certificate warning is raised [59]. Since Cyberoam CA was not listed in browsers' trust anchors, no revocation had to be initiated by browser vendors; only users who trusted the Cyberoam CA by themselves were affected. On July 9th, 2012, Cyberoam released a security update that generated a unique CA certificate for each of their devices [60].

More information about the incident can be found in the following resources [58, 61, 59, 60].

### A.1.13. TURKTRUST (2012)

#### A.1.13.1. Description of the incident

TURKTRUST is a Turkish certificate authority and a subsidiary company of Turkish Armed Forces' Foundation. On December 24th, 2012, Google noticed that the users of their Chrome browser were presented with a fraudulent certificate for the [\\*.google.com](#) domain [62]. This certificate was detected via Chrome's certificate pinning mechanism. With access to the certificate chain, Google discovered that the fraudulent certificate was issued by an intermediate certificate authority linking back to TURKTRUST. Google immediately reported their findings to TURKTRUST and other browser vendors.

#### A.1.13.2. The reasons and mitigation

TURKTRUST published a report several days after Google had found the fraudulent certificate [63]. Apparently, in August 2011 during a software migration operation and system tests, TURKTRUST mistakenly issued two subCA certificates for [\\*.EGO.GOV.TR](#) and [e-islem.kktcmekzebankasi.org](#). These certificates were not intended to be subCA certificates. This incident remained unnoticed until December 2012.

In the beginning of 2012, one of the owners of subCA certificates, EGO installed in their network a firewall with MitM proxy configuration to analyze HTTPS packages [64, 63, 65]. Moreover, EGO had imported their subCA certificate into the firewall, which lead to the firewall generating fraudulent website certificates and intercepting traffic. One of those fraudulent certificates for [\\*.google.com](#) was later detected by Chrome's pinning mechanism. There is no evidence of any of those

certificates being used outside the internal network of EGO.

By the beginning of January, Google had blocked both subCA certificates in Google Chrome and decided to no longer indicate Extended Validation (EV) status for certificates issued by TURKTRUST [62]. On January 8th, 2013, Mozilla revoked trust for the two subCA certificates and suspended the inclusion of TURKTRUST's root certificate until their next audit [66]. Microsoft decided to update their Certificate Trust List (CTL), blocking the fraudulent certificate for the \*.google.com domain [67].

More information about the incident can be found in the following resources [62, 64, 67, 63, 66, 65].

## A.1.14. ANSSI (2013)

### A.1.14.1. Description of the incident

ANSSI (Agence nationale de la sécurité des systèmes d'information) is the French National Agency for the Security of Information Systems. ANSSI is operating the IGC/A (une Infrastructure de Gestion de clés Cryptographiques) root certificate and issuing certificates for French Government websites [68]. On December 3rd, 2013, Google detected several fraudulent certificates issued for a number of Google domains [69]. Initial investigation demonstrated that ANSSI's subordinate certificate authority had issued an intermediate certificate. The certificate was installed on a network monitoring device, which enabled the device to inspect encrypted traffic. Network monitoring device had generated numerous certificates, including certificates for Google's domains. These were detected via certificate pinning mechanism. Google blocked the intermediate CA and reported their findings to other browser vendors.

### A.1.14.2. The reasons and mitigation

According to ANSSI official statement [70], the main reason of the fraudulent issuance was a human error that had been made during the procedures of security enhancement of the infrastructure of the French Ministry of Finance. There is no evidence of any of the rogue certificates outside the French Ministry of Finance's internal network. After the incident, ANSSI made a promise to revise their practices and procedures [70].

On December 9th, 2013, Microsoft, Mozilla and Opera removed the trust of the fraudulent intermediate certificate [71, 72, 73]. On December 12th, 2013, Google announced that in future, ANSSI certificate authority would be recognized only for domains under French territories' top-level domains (.fr, .gp, .gf, .mq, .re, .yt, .pm, .bl, .mf, .wf, .pf, .nc, .tf) in Google Chrome [69].

More information about the incident can be found in the following resources [69, 68, 70, 71, 72, 73].

## A.1.15. NIC of India (2014)

### A.1.15.1. Description of the incident

The National Informatics Centre (NIC) of India is a science and technology group under the control of Ministry of Electronics and Information Technology of the Indian government. NIC of India is also a subordinate certificate authority of the Indian Controller of Certifying Authorities (India CCA). On July 2nd, 2014, Google detected numerous fraudulent certificates issued for a number of

Google domains, which had been issued by NIC of India [74]. Google immediately reported their findings to NIC of India, India CCA and Microsoft (The India CCA certificates were included in the Microsoft Root Store [75]).

#### A.1.15.2. The reasons and mitigation

On July 3rd, India CCA revoked all intermediate certificates of NIC of India. By July 9th, 2014, India CCA had conducted an investigation and shared their findings with Google [74]. Someone had compromised NIC of India and managed to issue at least four fraudulent certificates, one of them for a Yahoo domain and three of them for Google domains. However, according to Google [74], they witnessed other rogue certificates issued by NIC of India. Google blocked the fraudulent certificates in Google Chrome and limited the recognition of India CCA root certificate to only certain Indian domains ([gov.in](#), [nic.in](#), [ac.in](#), [rbi.org.in](#), [bankofindia.co.in](#), [tcs.co.in](#), [ncode.in](#)) [74]. Microsoft updated their Certificate Trust List to revoke the three CA certificates for the NIC of India [76]. This incident did not affect Mozilla Firefox because NIC of India and India CCA certificates were not included in Mozilla's root store [77].

More information about the incident can be found in the following resources [74, 76].

### A.1.16. Comodo (2015)

#### A.1.16.1. Description of the incident

In January of 2015, a Finnish user noticed that Microsoft's email service [live.fi](#) allowed the creation of many different aliases for a single email account [78, 79]. Due to curiosity, the user tried to create an account with the alias [hostmaster@live.fi](#), and succeeded. Next, the user asked Comodo to issue a fraudulent certificate for the domain [live.fi](#), using the created account. Comodo automatically generated the certificate.

#### A.1.16.2. The reasons and mitigation

The main reasons for this incident were flaws in Comodo's Domain Control Validation (DCV), and the vulnerability in [live.fi](#) that allowed the creation of accounts with highly privileged aliases. According to Comodo support page [80], to receive a domain-validated certificate, the person needed to send an application from some of the domain's "admin type email addresses" ([admin](#), [administrator](#), [postmaster](#), [hostmaster](#), [webmaster](#)). Comodo's system responded with an email containing a unique validation code and link. Clicking the link and inputting the code was enough to validate the ownership of a domain.

The user immediately reported their findings to Finnish Communications Regulatory Authority and Microsoft [78, 79]. However, Microsoft published the security advisory only around six weeks after vulnerability was found, on March 16th, 2015. Microsoft blocked the user's [live.fi](#) account and revoked the fraudulent certificate [81].

More information about the incident can be found in the following resources [81, 78, 79].

### A.1.17. CNNIC (2015)

### A.1.17.1. Description of the incident

The China Internet Network Information Center (CNNIC) is a Chinese government department responsible for [.cn](#) domain registry. On March 20th, 2015, Google noticed several fraudulent certificates issued for a number of Google domains [82]. These rogue certificates were issued by the Egyptian company MCS Holdings. For that, MSC Holdings used an unconstrained intermediate certificate issued by CNNIC. Google reported their findings to CNNIC and other browser vendors.

### A.1.17.2. The reasons and mitigation

According to CNNIC official statement [83], MCS Holdings improperly issued the "sub-ordinate" certificates, however, they were only used for internal tests in MCS Holdings' protected environment. On March 22nd, 2015, CNNIC revoked their authorization to MCS Holdings. According to CNNIC's response to Google [82], MCS Holdings had a legal agreement with CNNIC that MCS Holdings would only issue certificates for their own domains. However, MCS Holdings had installed intermediate certificate in a firewall proxy device, which started intercepting encrypted traffic. Someone from their network accessed exterior servers, which lead the device to issue certificates for other domains, including Google domains. These certificates were detected via Chrome certificate pinning mechanism.

On March 22nd, Google had blocked the MCS Holdings' certificate in Google Chrome and later, on April 1st, Google made a statement that CNNIC Root and EV certificates are no longer recognized in Google products [82].

In April 2015, Mozilla released the report on the incident [84]. According to this report, MCS holdings did not have a Certificate Practice Statement (CPS) in place, hence having no approved Key Generation Script, nor a Point-in-Time-Readiness Assessment. Moreover, MCS Holdings stated that CNNIC had not given them any guidance or instructions on how to manage an unconstrained intermediate certificate. Such lack of communication led to the storage of certificate's private key on a firewall device. Mozilla stated that CNNIC issuance of this intermediate certificate violated Mozilla's CA Certificate Inclusion Policy and Baseline Requirements.

Mozilla revoked MCS Holdings' fraudulent intermediate certificate [85]. Due to the fact that CNNIC had violated their own CPS, Mozilla decided to not trust any certificate issued by a CNNIC root with a [notBefore](#) date on or after 1st of April 2015 [84]. On March 24th, 2015, Microsoft blocked the fraudulent MCS Holdings certificate in Internet Explorer but did not remove CNNIC from their Certificate Trust List [86].

More information about the incident can be found in the following resources [82, 83, 84, 85, 86].

## A.1.18. Symantec (2015)

### A.1.18.1. Description of the incident

Symantec Corporation was an American software company that was also providing certificate authority services. In 2019, it was re-branded and is now known as NortonLifeLock. On September 14th, 2015, Google detected that Symantec had issued an unauthorized Extended Validation pre-certificate for Google domains [87]. Google detected the suspicious issuance via Certificate Transparency logs [88]. Google notified Symantec and other vendors about the incident.

Symantec responded to Google's statement, disclosing that 23 certificates had been issued for testing purposes without notifying domain owners [89]. After this discovery, Symantec revoked unauthorized test certificates and stated that there had been no impact for domain owners. However, Google was not satisfied with Symantec's findings, due to the number of questionable certificates in the Certificate Transparency logs [90]. After additional audit, Symantec declared that they issued 164 certificates for 76 domains [91] and 2458 certificates for domains that were never registered [92].

#### A.1.18.2. The reasons and mitigation

Google demanded that starting from June 1st, 2016, all certificates issued by Symantec shall support Certificate Transparency. Moreover, Google required a third-party security audit [90]. Opera decided to make no changes to Opera root store and wait for a more detailed report from Symantec [93].

More information about the incident can be found in the following resources [87, 88, 90, 94, 89, 93]. Due to the acquisition of Symantec by Broadcom Inc. and the restructuring of Symantec's website, the public links to incident reports are no longer accessible.

### A.1.19. SK ID Solutions (2015)

#### A.1.19.1. Description of the incident

SK ID Solutions AS (formerly AS Sertifitseerimiskeskus) is an Estonian company operating a certificate authority that issues certificates for Estonian national identity documents. On September 1st, 2015, SK ID Solutions announced that in summer 2015 they had issued certificates for 4120 Estonian ID cards with incorrectly duplicated [@eesti.ee](#) email addresses [95]. The error meant, that people sharing both the first and the last name had the same email address specified in their certificate, and emails sent to that address were forwarded to both namesakes [96].

#### A.1.19.2. The reasons and mitigation

The main reason of this incident was error in SK ID Solutions' software, which generated repeated email addresses for namesakes, which were then included in certificates [95]. The right to forward emails sent to their [@eesti.ee](#) addresses was suspended for all victims of the incident. From September 8th, 2015, Estonian Police and Border Guard Board started inviting people affected by the incident to renew their certificates. This incident had no effect on the electronic use of documents or the creation of digital signatures.

More information about the incident can be found in the following resources [95, 96].

### A.1.20. StartSSL and WoSign (2015-2016)

WoSign Limited is certificate authority company located in China. Between 2015 and 2016, a series of incidents involving WoSign and StartSSL took place, leading to the distrust of their certificates by all major browser vendors [97, 98, 99, 100]. Most of the listed incidents were publicly disclosed after Gervase Markham reported three security incidents associated to WoSign [101]. It lead to Mozilla asking WoSign to provide clarification about each of the incidents. On September 4th, 2016, WoSign published an incident report and, subsequently, Mozilla created a wiki page with security issues,

asking WoSign to provide a response to them [102]. WoSign published another report that responded to Mozilla's wiki page [103]. Let us describe some of the major issues here.

### A.1.20.1. SHA-1 Certificates (January-March 2015)

#### A.1.20.1.1. Description of the incident

Between January 16th and March 5th, 2015, WoSign issued 1132 certificates using the [SHA-1 hash function](#), whose validity extended beyond January 1st, 2017 [104]. At that time, most browser vendors were avoiding the use SHA-1 due to the potential collision attacks [105, 106]. Also, *Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates* [107] stated

Effective 16 January 2015, CAs SHOULD NOT issue Subscriber Certificates utilizing the SHA-1 algorithm with an Expiry Date greater than 1 January 2017.

— Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates, Section 7.1.3

In this statement, the phrase "SHOULD NOT" has to be interpreted according to RFC2119 [108], which states

This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

— RFC2119, definition of the phrase SHOULD NOT

Hence the issuance of SHA-1 certificates by WoSign was technically not violating the Baseline Requirements (BR).

#### A.1.20.1.2. The reasons and mitigation

According to WoSign response [103], they were aware of SHA-1 BR recommendations and were updating their PKI system. However, due to unexpected delays, WoSign could not perform required updates by March 5th, 2015. WoSign also contacted all their customers for whom they had issued SHA-1 certificates, and offered to replace them with SHA-2 certificates. Moreover, WoSign promised to revoke all these certificates if they would not have been replaced by December 21st, 2016.

### A.1.20.2. Two identical certificates with different NotBefore date/time (March 2015)

#### A.1.20.2.1. Description of the incident

In March 2015, WoSign issued two identical certificates which differed only in `notBefore` date by 37 seconds [102, 103]. Later investigation by WoSign [103] showed that there were at least 16 similar

cases.

#### A.1.20.2.2. The reasons and mitigation

According to WoSign response [103], this incident was caused by their CMS (Certificate Management System). Apparently, it was sending a signing request to first signing server, but got no response. After that, CMS was sending the request to another signing server. Both servers eventually signed and returned the certificate, with the first being sent to the subscriber, but both stored by the CMS with the certification order, with the second certificate overwriting the first. WoSign eventually fixed this bug, but did not revoke the certificates, stating that duplicates were never used in public.

#### A.1.20.3. Various Violations of Baseline Requirements (April 2015)

##### A.1.20.3.1. Description of the incident

On April 3rd, 2015, Google shared their concerns with WoSign about some certificates violating Baseline Requirements [102]. The problems found by WoSign and Google were:

- The Certificate Practice Statement (CPS) required use of obscure "Issuer Alternative Name" field, which was not being used;
- The CPS documentation of validity periods did not match WoSign's internal practices;
- Incorrect or missing policy Object Identifiers (OIDs) in subscriber certificates;
- Inclusion of information in the Subject of the certificate (an advert), which was not validated subscriber information, and which contained a domain name, violating section 7.1.4.2 of the BR [107].

##### A.1.20.3.2. The reasons and mitigation

WoSign resolved these by improving some of their practices, and by making changes to their CPS [109].

#### A.1.20.4. Any port for a domain validation(January-March 2015)

##### A.1.20.4.1. Description of the incident

Between January 10th and April 23rd, 2015, WoSign was allowing an applicant to use any port (including the standard HTTP and HTTPS ports 80 and 443) during the domain validation process [101]. In total, there were 72 certificates issued, where domain validation had used ports different from 80 and 443.

##### A.1.20.4.2. The reasons and mitigation

According to WoSign report [103], some of their customers were not able use ports 80 or 443 for performing the website control validation and requested a change to use any port for this validation. WoSign fixed the bug and disabled the website control validation for ports different from 80 and 443. The issued certificates were not revoked.

### A.1.20.5. Secret acquisition of StartCom/StartSSL (November 2015)

During the investigation of a CA related incident, Mozilla found out that on November 1st, 2015, WoSign acquired 100% of StartCom, including the StartSSL certification authority. The acquisition took place through intermediary companies in the UK and Hong Kong [110]. While purchasing another CA is not illegal, the change of CA ownership must be disclosed, which was not done and was denied for several months. Subsequently, StartSSL started using WoSign's infrastructure and violate some of the Baseline Requirements similarly to WoSign.

### A.1.20.6. Backdated SHA-1 Certificates (January 2016)

WoSign issued certificates after January 1st, 2016, but backdated the `notBefore` date to December 2015 [102]. Such action allowed to bypass the browsers' rejection of SHA-1 certificates issued after January 1st, 2016. The number of affected certificates was around 67.

The issuance of backdated certificates was not forbidden by Mozilla's policy at that time, but was a part of CAs' *Problematic Practices* [111] that Mozilla had identified. Additionally, *Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates* [107] forbade the issuance of any new subscriber certificates or Subordinate CA certificates in 2016 or later.

### A.1.20.7. Aftermath

Despite WoSign publishing two reports covering a majority of the issues and providing the response [112, 103], majority of browser vendors lost their trust in WoSign and also in StartSSL, which they had acquired. On October 31st, 2016, Google stated that starting with Chrome 56, certificates issued by WoSign and StartSSL after October 21st, 2016 will not be trusted by Google Chrome [98]. Mozilla's actions were stronger [97]. They

- distrusted all WoSign and StartCom/StartSSL certificates with a `notBefore` date after October 21st, 2016;
- revoked the backdated SHA-1 certificates;
- no longer accepted audits carried out by Ernst and Young Hong Kong (the company that provided audits for WoSign).

Microsoft removed WoSign and StartCom certificates from Windows 10 [100]. Opera did not take any actions against WoSign or StartCom.

More information about the incidents can be found in the following resources [97, 98, 104, 102, 103, 101, 112, 110].

## A.1.21. Symantec (2016)

### A.1.21.1. Description of the incident

In October 2015, Andrew Ayer, software developer and founder of SSLMate, discovered that Symantec had been not accurately parsing administrative email addresses with `+` and `=` characters from WHOIS records [113]. The Baseline Requirements [114] define the administrative addresses that can be used for domain validation. The Baseline Requirements also allow email addresses from

a domain's WHOIS record to be used for a domain validation. Using the found vulnerability, a potential attacker could obtain certificates from Symantec for domains' WHOIS emails. For example, if a domain's WHOIS contact was [example+whois@domain.com](mailto:example+whois@domain.com), then Symantec would allow [whois@domain.com](mailto:whois@domain.com) address to approve certificates for the domain. Additionally, Andrew Ayer demonstrated how the vulnerability could be exploited showing an attack for a test domain [cloudpork.com](http://cloudpork.com) [113].

#### A.1.21.2. The reasons and mitigation

The main reason of this incident was Symantec interpreting WHOIS records in an insecure way, failing to parse special characters in email addresses. Andrew Ayer reported this vulnerability to Symantec and major browser vendors on October 21st, 2015. Symantec reported that the issue had been fixed on October 28th. Moreover, Symantec conducted additional audit to verify that this vulnerability had not been exploited [113]. The vulnerability was publicly disclosed on February 3rd, 2016 [115].

More information about the incident can be found in the following resources [113, 115].

### A.1.22. Comodo and PositiveSSL (July 2016)

#### A.1.22.1. Description of the incident

PositiveSSL is a subsidiary company of Comodo (Sectigo). On June 4th, 2016, security researcher Matthew Bryant found a vulnerability in PositiveSSL's website: the domain validation email field was not length-limited, which allowed anyone to inject arbitrary HTML to the end of the email address [116]. Using this vulnerability, a potential attacker could obtain fraudulent certificates by constructing dangling markup injection. For example, if the domain administrator would open the validation email in a client that supports HTML, and would click the rejection link, the attacker would have received the verification code for certificate issuance. Matthew Bryant demonstrated the exploitation of the vulnerability on his own domain [116].

#### A.1.22.2. The reasons and mitigation

The main reason of this incident was Comodo not sanitizing user's input on their website. On July 25th, 2016, Comodo confirmed they had fixed this vulnerability.

More information about the incident can be found in the following resource [116].

### A.1.23. Comodo (October 2016)

#### A.1.23.1. Description of the incident

On September 23rd, 2016, security researchers Florian Heinz and Martin Kluge reported a domain validation vulnerability [117, 118]. The Baseline Requirements [114] allow email addresses from a domain's WHOIS record to be used for a domain validation. The registries for the top-level domains (TLD) [.eu](#) and [.be](#) were only offering a port 43 WHOIS service which did not include the contact email addresses. On other hand, these registries also offered a web-based WHOIS service which presented the contact email addresses in the form of a graphical image. Comodo website was using Optical Character Recognition (OCR) to extract authorized administrative addresses from WHOIS

records. Heinz and Kluge showed that Comodo's OCR was not precise, and managed to obtain a server authentication certificate from Comodo for a domain which did not belong to them.

### A.1.23.2. The reasons and mitigation

The main reason of this incident was Comodo using an unreliable OCR system to interpret WHOIS information from .eu and .be TLD registries. Comodo performed an audit of all certificates issued that had been issued using OCR for domain validation [117]. Eventually, Comodo halted the use of the OCR technology [119].

More information about the incident can be found in the following resources [117, 119, 118].

## A.1.24. GoDaddy (2017)

### A.1.24.1. Description of the incident

GoDaddy Inc is an American Internet domain registrar and certificate authority. On January 6th, 2017, GoDaddy announced that they found a bug in their domain validation processing system [120, 121]. On January 9th, they performed the initial audit and found 8850 certificates that were issued without proper domain validation since July 29th, 2016.

### A.1.24.2. The reasons and mitigation

The main reason of this incident was a bug in GoDaddy's domain validation code that originally was meant to improve it [120, 121]. GoDaddy fixed the bug by January 6th, 2017. Additionally, they had to revoke 8850 certificates and provide a new certificate for each customer. There is no evidence that someone exploited this bug to issue malicious certificates.

More information about the incident can be found in the following resources [120, 121].

## A.1.25. Symantec (2017-2018)

### A.1.25.1. Description of the incident

On January 14th, 2017, security researcher Andrew Ayer discovered the issuance of several certificates for the domain `example.com` without the permission of the domain owner (ICANN), as well as questionable certificates containing the word `test` in the domain name or in the Subject field [122]. These certificates were issued by Symantec's RA, CrossCert. This incident lead to a detailed investigation [123, 124] that discovered and confirmed further 127 wrongly issued certificates, as well as other potential faulty issuances by CrossCert [125]. Moreover, poor CA practice in Symantec's RA network was discovered [124] which did not comply with the Baseline Requirements (BR) such as:

- use of domain names not owned by either Symantec or CrossCert without an approval from domain owners;
- typos and mistakes in domain names;
- absence of a legible Certificate Practice Statement (CPS);
- poorly trained personnel performing certificate issuance.

### A.1.25.2. The reasons and mitigation

During several investigations and audits [124, 125, 126], it became clear that Symantec had entrusted several of their RAs with the ability to issue certificates without the appropriate or necessary supervision and oversight. Symantec shut down their RA program and made plans to evaluate every certificate issued by their RAs [124]. However, major browser vendors lost the trust in Symantec after this incident. Google announced that starting with Chrome 66, they had removed the trust in Symantec certificates issued prior to June 1st, 2016 [127]. Mozilla stated that Firefox 60, which was expected to enter Beta on March 13th, removed the trust in Symantec certificates issued prior to June 1st, 2016 [128].

More information about the incident can be found in the following resources [123, 124, 127, 128].

## A.1.26. Certinomis (2018)

### A.1.26.1. Description of the incident

Certinomis is a French certification authority. In April 2019, security researcher Andrew Ayer discovered that Certinomis had issued 14 certificates for the domain [mediatheque-lecannet.fr](#) [129]. At this time, this domain had not been registered at any time since January 2017 [130]. Certinomis responded that this incident was caused by a human error, and could not be reproduced in responding to other certificate requests. However, Mozilla started additional investigations, gathered information about other security incidents involving Certinomis, and presented the results on the Wiki page [131]. Some of those incidents were [132]:

- **StartCom Cross-signing (2017):** Certinomis had created cross-signature of two intermediate certificates for StartCom in 2017. At that time, StartCom had been distrusted after the series of WoSign incidents described in Sec. A.1.20.
- **Lack of Communication or Response to Mozilla's requests:** Certinomis was ignoring Mozilla's requests or was not responding in timely manner.
- **Audit Issues (2015-2018):** a three-year gap in assessment reports.
- **CPS compliance:** The 25th of November, 2018 version of Certinomis' Certificate Practice Statement (CPS) did not comply with the Baseline Requirements (BR).
- **Non-BR-Compliant OCSP Responders:** Certinomis' OCSP responders were violating the BRs by returning *good* in response to a request for an unknown certificate [133].

### A.1.26.2. The reasons and mitigation

On May 9th, 2019, Certinomis reported all issues, explaining how they addressed them [131]. However, on May 13th, 2019, 174 pre-certificates with *unknown* OCSP status were discovered, proving that Certinomis had issued certificates by not following BRs.

Mozilla found even more unresolved security issues and made a decision to remove the Certinomis Root CA from Mozilla root store [134]. Additionally, they decided to treat any cross-signature of the existing root CA as a policy violation that will result in the immediate addition of the cross-certificate to OneCRL [132].

More information about the incident can be found in the following resources [129, 130, 131, 133,

[\[132\]](#).

## A.1.27. Kazakhstan Root CA (2019)

### A.1.27.1. Description of the incident

In July 2019, The Kazakhstan government issued a fraudulent root certificate in order to perform MitM attacks to intercept encrypted traffic of Kazakhstan's citizens [\[135, 136\]](#). Internet Service Providers (ISPs) in Kazakhstan were trying to convince their customers to install the government-issued root certificate on their devices. Interception of the users' traffic was first detected on July 17th, however, only certain sites were intercepted. According to Sundara Raman et al. [\[135\]](#), at least 37 domains were affected, including Google, Twitter, and Facebook domains.

### A.1.27.2. The reasons and mitigation

This incident affected some, but not all, of Internet users in Kazakhstan. Mozilla blocked the usage of the Kazakhstan root CA certificate and encouraged Kazakhstan citizens to use virtual private network (VPN) software, or the Tor Browser to access the Web [\[137\]](#). Google had taken similar steps by blocking the root CA certificate and, moreover, added it to a blocklist in the Chromium source code blocking the certificate in Chromium based browsers [\[138\]](#).

More information about the incident can be found in the following resource [\[135, 136, 137, 138\]](#).

## A.2. Discussion of incidents

### A.2.1. Scenarios

It is challenging to compare every case presented in the previous section, because the availability of information on each incident varies, and history of most incidents is quite unique. However, it is possible to outline the main kinds of CA incident scenarios (based on NIST ITL Bulletin for July 2012):

- **Impersonation:** An attacker impersonates some specific company or person to the registration authority and is issued a certificate for that company or person. Examples of this scenario are VeriSign and Microsoft (Sec. [A.1.1](#)), Thawte (Sec. [A.1.2](#)), StartSSL (Sec. [A.1.3](#)), Comodo and CertStart (Sec. [A.1.4](#)), Comodo (Sec. [A.1.16](#)), Symantec (Sec. [A.1.21](#)), Comodo and PositiveSSL (Sec. [A.1.22](#)), Comodo (Sec. [A.1.23](#)) incidents.
- **RA compromise:** An attacker corrupts the RA to be able to authorise the issuance of fake certificates by the CA. One of the example of such scenario is Comodo (Sec. [A.1.6](#)) case.
- **CA System Compromise:** An attacker corrupts the CA itself to be able to use the certificate issuance system and issue fake certificates. However, in this scenario, the attacker does not obtain the copy of CA's private key, but is able to use it for certificate issuance. Examples of this scenario are Diginotar (Sec. [A.1.8](#)), StartSSL (Sec. [A.1.7](#)) and NIC of India (Sec. [A.1.15](#)) incidents.
- **CA Signing Key Compromise:** An attacker is able to get a copy of CA's private key to sign arbitrary fake certificates and CRLs.
- **Inefficient internal policy and controls:** Absence of established CA policies or processes may result in improper action. Examples of this scenario are Pos Digicert Sdn.Bhd. (Sec. [A.1.10](#)),

Trustwave (Sec. A.1.11), Cyberoam (Sec. A.1.12), TURKTRUST (Sec. A.1.13), CNNIC (Sec. A.1.17), Symantec (Sec. A.1.18), StartSSL and WoSign (Sec. A.1.20), Symantec (Sec. A.1.25), Certinomis (Sec. A.1.26), Kazakhstan Root CA (Sec. A.1.27).

- **Human/Technical error:** A human error or a bug in a code can be the reason for an incident even in a CA with clear policies. Examples of such scenario are ANSSI (Sec. A.1.14), SK ID Solutions (Sec. A.1.19), GoDaddy (Sec. A.1.24) incidents.

## A.2.2. Prevention and incident response techniques

Over the years, IT community and browser vendors have established methods which can prevent or minimize the impact of a CA related security incident:

- **Certificate Revocation:** In the case of compromising the key for which the certificate has been issued, the CA has a mechanism of revoking the certificate. Revocation information is published, either using a Certificate Revocation List (CRL) or via the Online Certificate Status Protocol (OCSP).
- **Certificate Pinning:** This is a mechanism that allows user agents (e.g. web browsers) to check that only authorized CAs are issuing certificates for specific web sites. Certificate pinning mechanism calculates the hash of the public key contained in a certificate, and checks it against the hash value of the public key contained within the browser. Even though this mechanism can be used only for a small number of the domains, it works for the most popular web domains in the world. Certificate pinning was added to Chrome 13 [139], it helped to detect the MitM attack during Diginotar 2011 incidents (Sec. A.1.8), and it noticed TURKTRUST (Sec. A.1.13), ANSSI (Sec. A.1.14), NIC of India (Sec. A.1.15) and CNNIC (Sec. A.1.17) rogue certificates for Google domains.
- **Certificate Transparency:** This initiative [88] was a response to the 2011 attack on DigiNotar and other Certificate Authorities. These attacks showed that the lack of transparency in the way CAs operated was a significant risk to the Public Key Infrastructure used in the Web. Certificate Transparency is a mechanism, where public append-only logs of issued certificates are maintained, and anyone can query them to see what certificates have been included and when. The Certificate Transparency program is supported by Apple and Google. Certificate Transparency helped to detect the mississue of certificates by Symantec (Sec. A.1.18), WoSign (Sec. A.1.20) and Certinomis (Sec. A.1.26).
- **CA/Browser Forum and Baseline Requirements:** The Certification Authority Browser Forum, also known as the CA/Browser Forum, is a voluntary organization of certification authorities, browser, PKI and OS vendors that promotes industry guidelines governing the issuance and management of digital certificates [140]. Even though it is a voluntary organisation, major browser vendors and CAs follow the Baseline Requirements published by it. Baseline Requirements audits brought WoSign's (Sec. A.1.20) and Symantec's (Sec. A.1.25) poor CA issuance practices to the public eye.
- **Established cybersecurity community:** No incident prevention mechanism is perfect. Thanks to the community of security researchers, penetration testers and white hackers, vulnerabilities found in Thawte (Sec. A.1.2), StartSSL (Sec. A.1.3), Comodo and PositiveSSL (Sec. A.1.22), and Symantec (Sec. A.1.21) incidents were never exploited by malicious parties.

## A.3. Incidents affecting availability

Integrity and availability are the main security and functionality properties that we associate with and expect from CAs' processes. The result of all incidents described in Sec. A.1 has been the breach of integrity of the offered services. Some of these incidents may have also caused a loss of availability, when browser vendors have removed their trust to certificates issued by certain CAs, but this loss was caused mainly by the mitigation actions, not by the incident itself.

There have definitely been losses of availability, i.e. outages of trust services; indeed, they take place with some regularity even in Estonia (e.g. on [May 9th and 10th, 2019](#), [September 6th and 9th, 2019](#), [December 16th, 2019](#), [March 5th, 2020](#), [January 5th, 2021](#), [May 21st, 2021](#); this list is definitely not exhaustive). Usually, they only last for a few hours, and the community tends to forget them quickly.

There exists a [list](#) collecting significant events of DNSSEC outage. The page states that the median duration of a DNSSEC is eight days. The main reasons for outages do not appear to be malicious. Rather, they are caused by incorrect updates.

The Wikipedia entry on [Internet outages](#) gives a list of incidents. None of the included incidents are related to the non-functionality of trust infrastructure.

# B. Appendix: Proposed trust models

## B.1. Speaking about beliefs and trust

A number of (at least slightly) different trust models have been proposed in the research literature, standards, regulations; or have emerged organically. In this chapter, we describe and analyze them in the context of key ownership. The statements that we either want to *trust*, or do not want to *trust*, are answers to the question "does the public key  $pk$  belong to the agent  $A$ ?" The (positive) answers to these questions are represented by *certificates*. We let  $\mathcal{C}(pk,A)$  denote that agent  $A$  indeed controls  $pk$ .

Speaking about agents controlling public keys is suitable for discussing both authentication and signing. In the case of authentication, an agent proves knowledge of a private key that corresponds to a certain public key, which is linked to a certain natural person (or a legal entity) by a certificate. A signature is created using a private key, and can be verified with the corresponding public key, which is again linked to a certain natural person (or a legal entity) by a certificate.

There is one formal difference between the requirements for these two scenarios:

- For authentication, we only need to know which key belongs to which user *right now*.
- For verifying a signature, we need to know which key belonged to which user *at a certain time in the past*.

Hence, the trust models for authentication and signing will be essentially the same, differing only in the trusted statement, i.e. does the claim hold about the present, or a certain point of time in the past.

Differently from *claims*, the trust *relations* denote the trust that the agent is having right now, i.e. in the present timepoint. If agent  $A$  has trusted agent  $B$  yesterday, but the trust has been lost today, we assume that also the yesterday's statements of  $B$  are not trusted by  $A$  anymore. On the other hand, if  $A$  believes that  $B$  controlled the public key  $pk$  yesterday, then learning that  $B$  lost the control today, will by itself not change the beliefs about yesterday.

In a trust relationship, one agent trusts another one **with respect to a certain class of statements**. We consider the following basic trust relations, to formally state that an agent may trust another agent. The trust is with respect to statements about control of public keys.

- Weak trust (denoted  $\dashrightarrow$ ) means that only the agent's statements about his *own* public key are trusted. The ownership of the corresponding private key can be proved cryptographically. This type of trust is needed for a CA to issue a certificate to an agent.
- Strong trust (denoted  $\rightarrow$ ) means that the agent's statements about *other agents'* public keys are trusted. A strong trust is needed for a CA to certify another CA.
- Trust with responsiveness is denoted by a double arrow (e.g.  $\Rightarrow$  denotes the strong trust  $\rightarrow$  with responsiveness). This means that at any time in the future the agent will be available to tell whether he believes that  $\mathcal{C}(pk,A)$  holds or not. This property is important for signatures, and also for authentication if the keys can be revoked. For example, while the relation  $A \dashrightarrow B$  permits  $B$  to have presented a different public key to some agent  $C$  while hiding that public key

from  $A$ , the relation  $A \Rightarrow\!\!> B$  states that  $B$  honestly tells to  $A$  whether the key that he showed to  $C$  is indeed his. Responsiveness of a CA may be supported technically by some kind of OCSP service, which we will not model explicitly. When a CA certifies another CA, it should also believe in the other CA's responsiveness.

A *trust assumption* is a propositional formula about trust relationships, i.e. a formula that combines statements of the form  $A \rightarrow B$  (with arbitrary kinds of arrows) using the propositional connectives (i.e. conjunction, disjunction, implication, negation). A *trust model* is a set of trust assumptions that must be accepted by the agents, in order to make the trust infrastructure work. Assuming that a set of trust assumptions  $\mathcal{M}$  holds, it may be possible to deduce that further assumptions also hold. We want to be able to deduce  $A \Rightarrow\!\!> B$  for all agents  $A$  and  $B$  in the considered domain. We let  $\models \mathcal{M} \models A \Rightarrow\!\!> B$  denote the ability to make such deduction from the trust model  $\mathcal{M}$  for the agents  $A$  and  $B$ . We may also have any other claim at the right hand side of  $\models$ , meaning that this claim can be deduced from the trust model.

In this chapter, we will present various trust models, describing the trust assumptions that they contain. We do this description pictorially, using stick figures to represent agents / users. In the images below, a stick figure represents a single user, i.e. a natural person or a legal entity. A rounded rectangle surrounding multiple users denotes that these users belong to the same domain, i.e. they are covered by the same CA. The agents playing the role of  $A$  (the trustees) are labeled by a question mark (?), and the agents playing the role of  $B$  (the trustees) are labeled by an exclamation point (!).

There are two kinds of trust assumptions. The first kind has to be *consciously* accepted by the trustee. The second kind holds, because it is enforced by technology and cryptography. In the descriptions below, we only depict the first kind of assumptions, as the differences between them can make one trust model superior to another one.

## B.2. Trust models without weak responsiveness

We first introduce two simple trust relationships, neither of which is strong enough to give us  $A \Rightarrow\!\!> B$ . However, they serve as a good introduction to our notation.

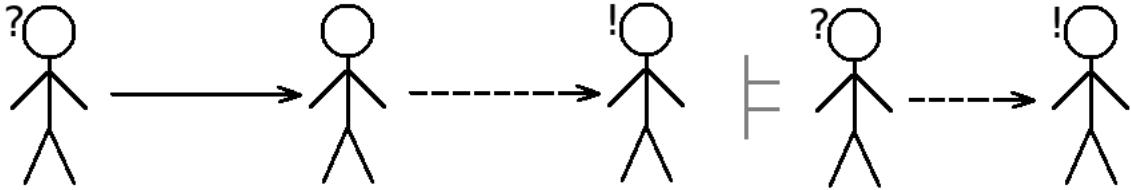
**Direct Trust.** The simplest solution for Alice and Bob who know each other is to create their own self-signed certificates and confirm them in person. Each entity makes statements only about their own keys, so it suffices to have a weak trust. Bob is free to present different public keys to different agents, and we are not claiming responsiveness. This situation is depicted below; it is at the same time both the trust assumption, and the guarantee provided by the trust model.



**Indirect Trust.** Suppose that Alice had established trust with Bob through direct trust, and obtained a trusted certificate of Bob. In a similar way, Bob obtained a trusted certificate of Chris. If Alice and Chris want to communicate, but cannot meet in person in order to exchange their public keys, then Bob may be able to help. In addition to the axioms of direct trust, we will have a rule for

indirect trust. The idea is that, if Alice believes that Bob's public key is  $pk_B$ , and Bob believes that Chris's public key is  $pk_C$ , then he can convince Alice of that as well, so that *direct trust* between Alice and Chris can be established.

Now Alice has to trust that Bob reports the public key of Chris honestly, which requires a higher level of trust, since a malicious Bob can impersonate Chris by reporting a fake public key  $pk_C$ . Moreover, even if Bob is honest, Alice must still believe that Bob was not deceived by someone who managed to impersonate Chris. Therefore, here we need strong trust. The trust assumption is depicted below, to the left of  $\models$ . The formula to the right of  $\models$  shows, what may be deduced from this trust assumption.



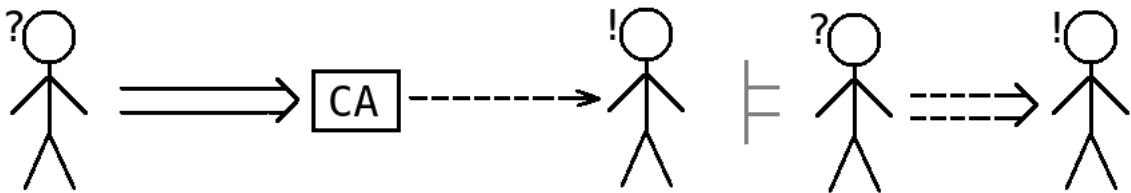
Even if Bob is always available, he still cannot prove that a public key *does not belong* to Chris, since Bob is not necessarily aware of that. Hence, it would be unreasonable to assume responsiveness. This problem could be solved if Bob would have some privilege of assigning public keys to agents, so having a key that is not confirmed by Bob is equivalent to not having it. This leads to CA models, which we discuss in the next section.

## B.3. Classical PKI

In classical PKI, there is a special party called a CA (Certificate Authority). The CA tells, who holds or does not hold which public key, either currently or in the past. The other parties must trust the CA.

Besides the trustworthiness, it is also necessary that the CA actually tells whether a key is controlled by an agent or not, i.e. the CA should be responsive.

In addition, and similarly to indirect trust, whenever a CA issues a certificate for an agent, the CA must *trust* that agent to a certain extent. Namely, when the CA confirms that the public key  $pk$  belongs to Bob, then CA must be convinced that Bob is indeed in control of the corresponding secret key. Here a weak trust from CA towards Bob is sufficient. With the help of CA, a weak trust with responsiveness can be established between Alice and Bob.



We note that a CA might not be aware of Bob's certificates that he has received from some other, independent CA-s. Here we assume that responsiveness covers a particular *domain* of users, and queries are made only about the  $\mathcal{C}(pk, B)$ -statements that hold within the domain.

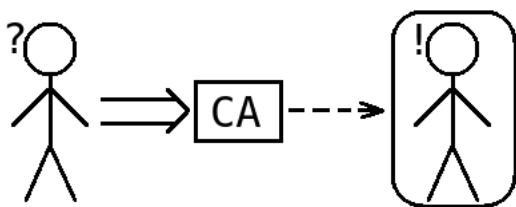
This basic model with a certification authority responsible for a domain of users can be set up and extended in a number of ways. The following classification is quoted from [141], where such authority is called a *trust anchor*.

### B.3.1. Dedicated Domain PKI

In this model, digital certificates are associated to a single trust anchor. In this case, the trust anchor serves a single domain i.e. it is a dedicated anchor.

— [141]

This is exactly the classical model with a single CA described above. The weak trust from CA to the target user means that the CA needs to believe in the user's identity before issuing to them a certificate.

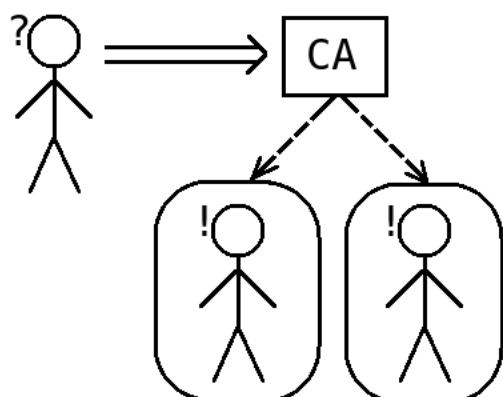


### B.3.2. Shared Domain PKI

In this model, digital certificates are also associated to a single trust anchor but, in this case, the trust anchor serves multiple domains i.e. it is a shared anchor.

— [141]

Here we are still assuming a single trust anchor, but there are multiple domains trusting the same anchor. The difference is, that the public keys of one domain will not necessarily convince the agents in another domain. We do not include distinct domains in our model, so we can just assume that there are several isolated worlds, each of which follows the same model as in the case of dedicated domain PKI.



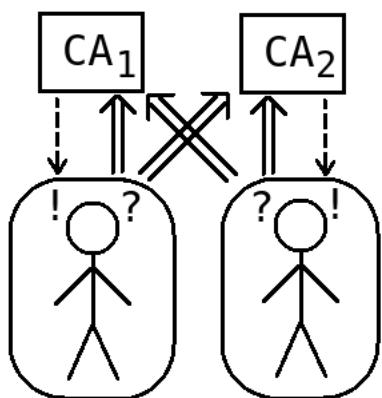
### B.3.3. Mutual exchange

Mutual exchange trust is a trust model based on the direct mutual exchange of digital certificates between different components.

This model relies on digital certificates from different trust anchors. As there is no single trust anchor, organizations use the trust anchor of their choice (typically, according a set of well-defined criteria).

— [141]

In this model, it is sufficient to corrupt any of the trust anchors in order to corrupt some of the certificates. However, the document [141] seems to consider several PKI anchors as a *good* property in terms of trustworthiness, because corrupting a single anchor will not compromise *all the certificates*.

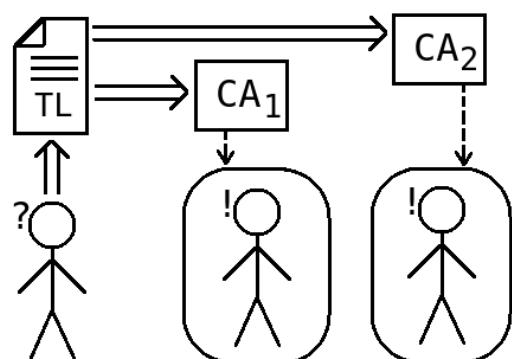


#### B.3.4. Domain trusted lists

This model relies on a list containing the trusted certificates and/or trust anchors complying with a common domain policy. As a result, organizations are free to choose their preferred trust anchor from that list.

— [141]

This model seems even less reliable than the previous one because the trust list, which might be corrupted, is an additional attack vector. At the same time, the party responsible for maintaining the list is a single point of failure in terms of availability as well. The document [141] agrees that this setting has lower trustworthiness than the previous three. In the following figure, the statement "the trusted list TL trusts the CA" (leftmost arrow) means that "the CA is included into the TL", which implies that the entity that generated the TL had to trust the CA.



### B.3.4.1. eIDAS trust list

Some more details about trust lists can be found in [142]. A trust list is typically managed either by a single user or by a trust provider. Both approaches have drawbacks, since a user may not be able to evaluate all risks properly, but at the same time blindly trusting the provider is not a good option either. As an example, let us see how the eIDAS list of trusted providers can be obtained.

- The trusted providers for different EU countries are listed on [this page](#). In order to use that list, we need to be sure that the webpage itself is not under attacker's control, so we need to verify its own certificate.
- The issuer of the certificate is Thawte RSA CA 2018. We need check that Thawte in turn can be trusted as a CA.
- The issuer of the Thawte's CA certificate is DigiCert Global Root CA.
- The certificate of DigiCert Global Root CA is self-signed. That certificate should be known in advance to our web browser.
- We have now verified the full certificate chain which tells us that the domain <https://esignature.ec.europa.eu> is owned by an entity called COMMISSION DE L'UNION EUROPEENNE-COMMISSIE VAN DE EUROPESE UNIE. Since they are *not* certified as a CA, we need to understand what this entity actually is, and we need to trust that entity with respect to the content of the trusted list.

We see that the full verification of origin of eIDAS trusted list eventually requires the verifier to have trust assumptions similar to what it would have in some hierarchical model (which we consider in more detail in Sec. B.4.1.2). Should we consider the solution as bad as the hierarchical model and claim that the DigiCert Global Root CA is a single point of failure? To some extent, it is, as the user who is searching for the trusted list on the web for the first time may get a malicious version of the trusted list. However, the users who have already been assured in the eIDAS trusted list once, can keep it locally in their computers for further reference, and continue trusting it even if the Root CA gets corrupted.

## B.4. Mesh PKIs (with multiple/distributed CAs)

The idea of Mesh PKI is to establish a sort of *indirect agreement* on certificates as in Sec. B.2, but using CA-s instead of parties.

### B.4.1. Without Thresholds

The following descriptions are based on [142]. First of all, we give a model for a general unstructured mesh of CA-s and then adapt it to meshes that follow a particular structure.

#### B.4.1.1. Mesh PKI

In the cross-certification model, also known as mesh model, two CAs cross-certify each other once they agree to trust and rely on each other's issued public key certificates as if they had generated them themselves. Pairs of CAs exchange cross-certificates and enable users from one administrative

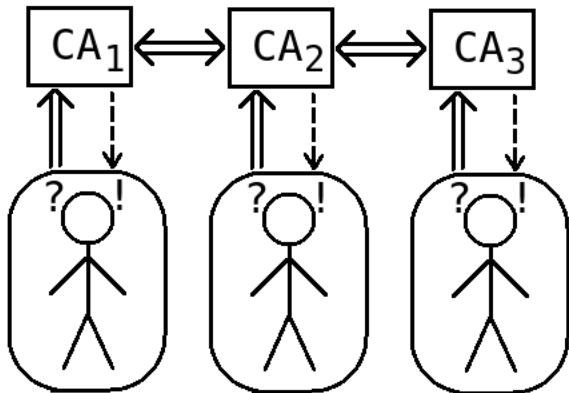
domain to interact electronically and securely with users from the other.

— [142]

In this model, the users trust only their own CA, who in turn trusts some other CA to which it is connected in a mesh network. Due to cross-certification, a user can trace a certificate from an unknown CA back to a local trusted CA. The chain of all CA-s on this path tracing the certificate back to the local CA must be "trusted". The trust to other CA-s does not come directly by the user, but through cross-certificates from CAs, meaning that the path must belong to the mesh graph. The Mesh model can be related to the trusted lists of Sec. <>sec:classical>:

From an inter-domain interoperability perspective, the provider trust list essentially replaces the cross-certificate pair in the Mesh model. The user trusts the issuer of the list, adopts the list, and then the trust is extended to the CAs conveyed within it.

— [142]

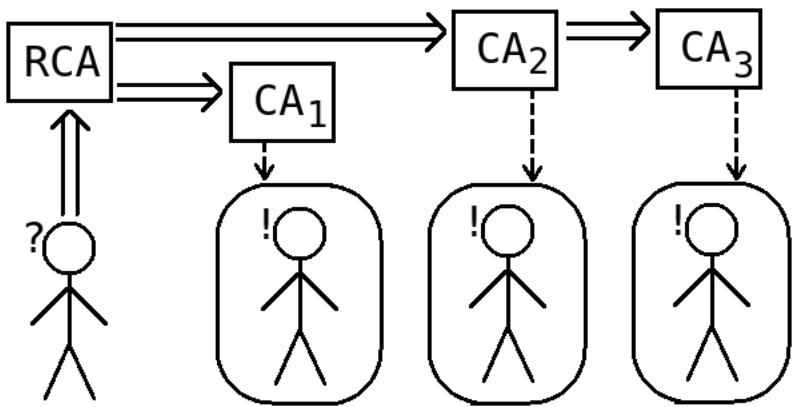


#### B.4.1.2. Hierarchical PKI

The first attempt to solve the problem of having multiple interconnected trust islands was the hierarchical PKI structure that is managed by a Root Certification Authority (RCA). Trust is established in a tree-like fashion and flows from top to bottom. The RCA public key is the fundamental point of trust, or trust anchor, for evaluating certificate acceptability. In this model the path construction procedure is very simple, as a single path exists from any end entity up to the RCA.

— [142]

This model can be viewed as a certain mesh topology with unidirectional trust propagation. Each user initially trusts only the Root CA (RCA). Each CA (including the RCA) trusts the CAs that are its immediate descendants in the topology, since it will "approve" the statements made by them.

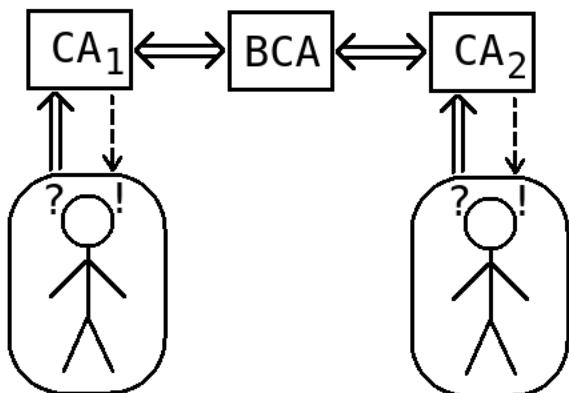


#### B.4.1.3. Bridge CA

Bridge CA (BCA) trust model was first introduced by the U.S. Federal Government as a way to facilitate the interconnection of CAs through a cross-certification process. Each user only trusts its own CA which in turn trusts the bridge that finally trusts the remote CA, so that each member needs only to maintain a single cross-certification with the BCA, and then it is automatically able to build certification paths across all spokes.

It must be noted that the BCA is not intended to be used as a trust anchor by the users of the PKI. It simply acts as a gateway between isolated CAs. Even so, BCA is responsible to map certificate policies and guarantee PKI equivalences adequately. Therefore, users must rely on it regarding these mappings.

— [142]



#### B.4.1.4. Bridge Validation Authority (VA)

The Bridge Validation Authority (BVA) trust model is a further step to the BCA in which the central entity is not a CA but a Validation Authority (VA). VAs are trusted third parties that offer online services on certificate validation. In general terms, they are responsible of building the certification path, evaluating the quality of the certificates, validating their

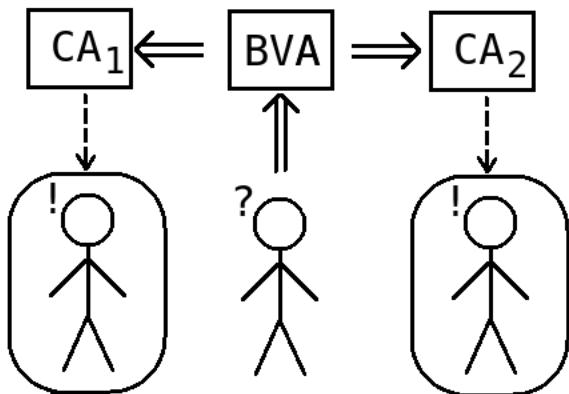
status, and ensuring they are trustworthy.

In the BVA model, the element that links multiple PKI islands is a VA that gathers and classifies the status information of certificates from multiple domains. BVA becomes the trust anchor for users and admits liabilities for the certificates it works with.

— [142]

In this trust model, the central bridge party of the previous point is considered as a special party—the Validation Authority (VA). The VA controls the trustworthiness of the CAs. The VA is not a CA by itself, so it does not issue signed certificates. Rather, it is an online service answering requests of relying parties [143].

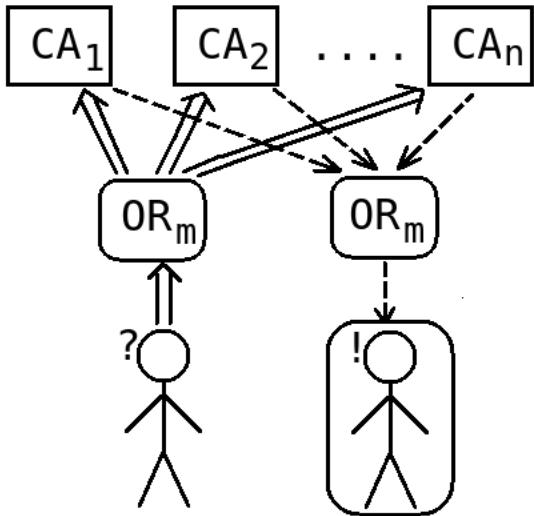
The main difference between BCA and BVA is that BVA is treated as a trust anchor for the users, while BCA is not. The trust model thus becomes more similar to a trusted list model, where instead of consulting an abstract trust list the user consults the BVA.



#### B.4.2. With Thresholds

A single CA could be distributed over several distinct machines. Since we want to achieve both integrity (corrupting a small number of machines should not make it possible to issue a fake certificate), and availability (corrupting a small number of machines should not halt the system), we are looking for a threshold certification approach.

In general, threshold PKI allows the distribution of a shared private key among CA components in such a way, that no single machine actually knows the value of that key. Hence, instead of trusting the CA, one only has to trust that at least a certain number of components are not corrupted. For example, if  $(n,m)$ -threshold sharing (read: "  $m$  out of  $n$  ...") is used, then a user believes that at least  $m$  out of the  $n$  CA-s are not corrupted (represented by a node  $OR_m$  in the following figure). Also, for the CAs to find out users' public keys, at least  $m$  CAs must have weak trust in them. Here the trustee belongs to the collaborative domain of the  $n$  CA-s, which are on a more abstract level treated as a single CA.



For example, using (3,2)-threshold sharing, there would be three CAs. A certificate of a user must have signatures from at least two of them. In order to follow the standards that prescribe a single signature on a certificate, the three CAs will likely use some form of *threshold signatures* [144]. In typical threshold signing schemes, each CA produces a *signature share* and sends it to the user. If the user has obtained two (not three) signature shares, then it can combine them into a signature. The system with (3,2)-threshold sharing gives us the following properties.

- If at most one server stops, the service continues working.
- If at most one server is under the control of an attacker, the latter still cannot cause fraudulent certificates to be issued. Indeed, two signature shares are needed for a certificate, and an honest server is not going to help the attacker.

There exist various examples of threshold signatures, and one suitable solution seems to be [145]. Postquantum-secure threshold signing algorithms are more complicated, and we may rather want to rely on information theory and certain assumptions than on computationally hard cryptographic problems. We probably also want to achieve a signature that follows some standard format.

### B.4.3. Guardtime KSI

As an alternative to a threshold signature (where the secret key is basically not held by anyone), we could consider so-called *keyless signatures*. Guardtime KSI is a platform for generating such signatures, which are linked to timestamps. The idea is, that hashes of signed messages are published together with the signers' names in a special kind of distributed ledger. In the following, we give a general overview of the system, based on [146, 147, 148] Particular details may depend on the implementation.

First of all, a document-signing user authenticates to a gateway, which is a special server responsible for accepting a signing request. We stress that this user has no signature key. Hence there are also no certificates for signature keys, and the claims to be trusted are somewhat different. Instead of the claims "Agent A controls the public (signature) key  $pk$ ", which we denoted  $\mathcal{C}(pk, A)$ , we now argue about trusting the claims of the form "Agent A wanted to get the document  $D$  signed".

The authentication gateway interacts with other special servers called the aggregators, who collect signature requests from many users and compute their hashes. These aggregator servers may in

turn have several layers of hierarchy, collecting hashes of the previous layers and thus constructing a hash tree. Eventually, the root of the hash tree (let us denote it  $h$ ), computed from the requests of numerous users, is sent to the *core cluster* of servers, which are responsible for publishing  $h$  in some trusted medium, e.g. in a periodically published newspaper. The user gets a special *signature token*, which together with the document and  $h$  is sufficient to verify whether the document was signed by that user.

Importantly, KSI is *not* an alternative to PKI, as it only allows keyless *signing*, but some keys are still needed for *authentication*. Moreover, since a corrupted gateway could potentially impersonate an honest user, a keyless signature must contain a *proof of authentication* [148], so some kind of PKI signature may still be necessary to enforce non-repudiation. As discussed in [148], an alternative solution is to assume a distributed scheme wherein a set of valid tokens received from  $t$  different gateways is needed to represent a valid signature, thus getting a trust model similar to threshold signing.

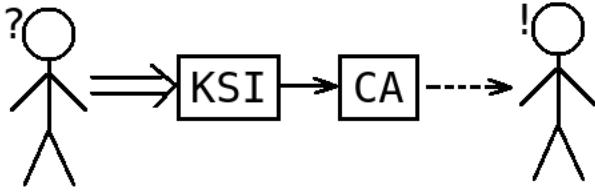
We also need to put certain trust into the aggregation servers and the core cluster, which may depend on the particular implementation. This kind of trust is similar to "trusting a blockchain", e.g. EBSI (European Blockchain Services Infrastructure), so if the Guardtime KSI is considered "trusted" by eIDAS regulation, then the fine details of the implementation of the ledger probably do not matter. In any case, using a KSI would give us the following benefits.

- Each signature is linked to a timestamp, which cannot be modified.
- Since there are no signing keys, there is no need for revocation.
- The PKI keys that are used for the authentication (and also for the proof of authentication) may still leak and need to be revoked using ordinary PKI revoke measures, e.g. assuming the availability of certain Online Certificate Status Protocol (OCSP) responders who confirm the validity of the keys [148]. However, such revocation would not affect the existing keyless signatures due to timestamping, as it can be proven that a document was signed *before* the key had leaked.

We see that we still need to maintain a PKI in the background, but timestamps would make key leakage less damaging. Even if Guardtime KSI fails for some reason, it does not make the situation worse than it would be without KSI.

A simple benefit that we could get from a KSI is to enhance the classical PKI model with a distributed ledger that serves as a trusted log. Any certificate issued by the CA would be stored in a blockchain, and a *keyless* signature is provided. Upon certificate status request, a user would need to consult the KSI, checking whether the keyless signature of the CA is correct. From the technical point of view, all OCSP responders would need to be able to access and use the KSI.

Let us discuss, what the trust model for such a setting would be. If a CA is corrupted, it can write garbage values to the blockchain anyway, so we cannot say that it is sufficient to rely on the KSI if the CA gets corrupted. The useful thing that we get from the blockchain is the responsiveness of CA, which is now due to technology, not due to trust. That is, if the signature is not yet in the blockchain, then it does not exist for the world. Hence, in order to get responsiveness, we can either trust that CA itself is responsive, or trust that the KSI is responsive, which means that the service indeed prints all the hashes it is supposed to print, and the newspaper continues to be published.



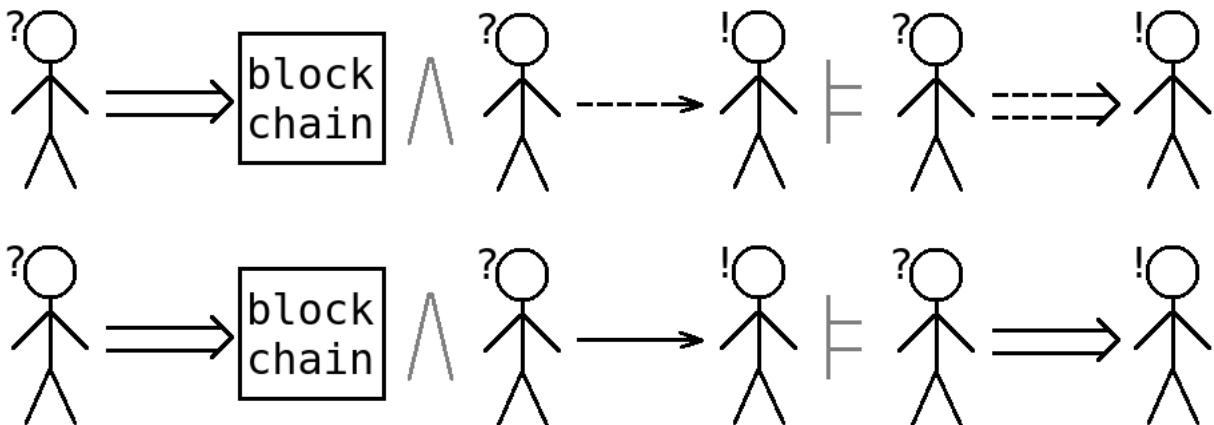
#### B.4.4. Web of Trust on a Distributed Ledger

The principles of direct and indirect trust (Sec. B.2) can be generalized to a network of agents. An example is PGP (Pretty Good Privacy), which provides a variety of functionalities like authentication, signing, and public key encryption. There is no certificate issued to PGP keys by a designated CA. Instead, the user himself has to somehow establish the trust that the used public key corresponds to the private key of the intended individual. Trust can be direct or indirect. These types of trust alone do not give us any responsiveness.

Solutions like ClaimChain [149] or [the DKMS](#) (Decentralized Key Management System) by Evernym describe an architecture of a distributed PKI based on a distributed ledger. The ledger serves as a trusted event log, guaranteeing that the logged claims cannot be repudiated. For simplicity, we omit the trust assumptions required for the ledger to work, as the precise assumptions depend on a particular implementation. We just assume that it implements an ideal trusted event log. The system implements functionality that allows issuing decentralized identifiers (DID) to users, and linking them to PKI credentials.

##### B.4.4.1. Trust model

A useful property that we get from a trusted log is the increased difficulty to make false claims, since all statements will be kept in the history of the ledger. That is, if Bob reports the claim  $\mathcal{C}(pk,C)$  to Alice, then Alice expects that Bob reports  $\mathcal{C}(pk,C)$  on the blockchain as well. The agents cannot read the public keys from the blockchain directly, but they can verify whether a certain agent has claimed  $\mathcal{C}(pk,C)$ . If Bob has not logged anything, then everyone assumes that Bob has not claimed  $\mathcal{C}(pk,C)$ . This adds responsiveness to the initial Web of Trust model, and we get the following trust model (recall that the operation  $\wedge$  means conjunction):



##### B.4.4.2. Example

Let us see how DID-s work in practice. Suppose that Alice wants to convince someone that she has a Ph.D. degree obtained from Green University. The system does not need to know that the physical person Alice exists in the real world. Instead, there is a user named e.g. *pumpkin* (a true DID will

actually be a random number). The trusted event log contains the following records:

- $pk_A$  is the public key of *pumpkin*;
- $pk_B$  is the public key of the user *university*;
- $pk_B$  has been used to sign that *pumpkin* has obtained Ph.D. degree in computer science;

Now anyone can believe that  $pk_A$  belongs to someone calling herself *pumpkin*, who has a Ph.D. degree in computer science, on assumption that the user *university* is eligible for issuing such degrees. If the system is fully decentralized and there is no trusted authority, by whom Green University could be registered when registering the DID *university*, then only those users who know the Green University in person can be convinced that it is the user *university*, similarly to the direct trust model. The trust can be propagated using indirect trust. It is probably fair to assume that there are sufficiently many users knowing Green University in person, and impersonations will be detected. It is more difficult to prove that *pumpkin* is Alice. It is particularly difficult if she and her close friends collaboratively lie that *pumpkin* is Bob, while Bob is not in the system and cannot even argue against it.

In order to answer the statement "which agent holds which public key", it is not enough to answer "which DID holds which public key". We also need to link the DID to an actual natural person or a legal entity, i.e verify that *pumpkin* is indeed the physical person Alice, and that *university* is indeed Green University. In general, strong linking to the real world requires some trust to certain authorities who are responsible for identifying the users who join the system.

#### B.4.4.3. Summary

An ideal trusted log system (if designed and implemented correctly, which is not so easy) could potentially create a strong correlation between PKI credentials and an artificial identifier. However, this technology only describes how the trust moves inside the system, but it does not cover the link to the real world. Technically, the relation  $\rightarrow$  is easier to establish with a centralized CA than with a decentralized blockchain, where the user would need to register at each stakeholder controlling the distributed ledger. We need a trusted mediator between the blockchain and the trustee.

An alternative solution is to move citizens into a fully virtual world, where issuing a new PKI credential would solely depend on correct older credentials, similarly to presenting an old passport in order to receive a new one, so that no visual observation would ever be needed. However, without any connections to the physical world, we would not be able to identify a physical person e.g. when he commits a crime, as his virtual identity would be completely separate. Thus in a governmental setting, we would anyway need a way to link a virtual identity to a physical person, and a reasonable solution would not be fully decentralized.

#### B.4.5. ESSIF

In the previous section, we described in general a Web of Trust model based on a distributed ledger. In this section, we discuss a particular system ESSIF [150] which specifies some important details that were missing in the general description, namely, how the DID-s are linked to the actual natural persons and legal entities. The ESSIF system is built upon a permissioned ledger EBSI [151], which is able to control who actually writes information to the ledger.

In the following, we will give an overview of ESSIF. We do not dive into details of implementing the ledger, and, as in the previous section, we assume for simplicity that it can be fully trusted. We also assume that the integrity and availability guarantees of the ledger can be taken for granted.

We study ESSIF in order to find out, whether an idealized system of this kind would fulfill our needs. The system allows certain entities to issue to users Verifiable Credentials (VC) such as a driving license. While we are only interested in the key management, and VC-s are out of the scope of this document, we still need a particular type of VC, called the Verifiable ID, which links a DID to an actual natural person or legal entity.

#### B.4.5.1. Roles

ESSIF includes the following roles [152].

##### B.4.5.1.1. Issuer

This term refers to a party that creates and issues Verifiable Credentials (e.g. Verifiable ID-s or Verifiable Attestations) to Holders.

— [152]

In general, the issuer does not have to be accredited, and a verifier should choose whether to trust the issuer or not. Governmental institutions should not trust arbitrary issuers; e.g. the Police need to check that the driving license is issued by a legitimate driving school. To make this decision easier, ESSIF assumes that there is a special registry of accredited trusted issuers, which says who can issue which kind of credentials.

##### B.4.5.1.2. Holder

This term refers to a party who is "holding" (or storing) Verifiable Credentials that have been issued to them by an Issuer.

— [152]

Even if a holder himself cannot be trusted, his verifiable credentials can, if they are issued by a trusted issuer. The holder needs to be trusted by the issuer at the time when a VC is being issued, and the trust may in turn be based on some other VC.

##### B.4.5.1.3. Verifier

This term refers to a party who requests/verifies Verifiable Credentials (e.g. Verifiable IDs or Verifiable Attestations), such as to provide a service.

— [152]

If the holder wants to make use of his VC-s, he should trust the verifier to whom he presents the VC, e.g. when he logs in to a bank.

#### B.4.5.1.4. ESSIF Onboarding Service (EOS)

This term refers to a function/service an organization can perform to onboard Natural Persons and Legal Entities to the EBSI ESSIF ledger (DID Registry) if it has the required authorization based on a classical identification means (that provide the required level of assurance as defined by relevant regulations).

— [152]

EOS is the entity that links physical Alice to the DID *pumpkin*. Alice may have several DID-s, but all of them should be linked to her. On the first onboarding, she gets a special kind of VC called a Verifiable ID, which is linked to the first of her registered DIDs. At this point, Alice should be able to prove that she is indeed the natural person Alice, so we need to assume that there exist some credentials "outside of ESSIF system", such as passport. EOSs need to be fully trusted; if an attacker controls an EOS, then impersonation is possible. We can think of it as the Police Department who is trusted to link a physical person to a DID. The accredited EOS entities are listed in a special trust register EOSR. In the newer ESSIF documents, EOS is also called a Registration Authority (RA), and the corresponding registry is called TRAR.

This description of EOS seems to contradict the claim that ESSIF is a framework for decentralized Self-Sovereign Identity (SSI). However, if we look deeper into the documentation of ESSIF, we can find the following remark:

As per remark that on the EBSI page on the CEF, it is stated 'without relying on centralized authorities' >> it should be clearly explained that this refer to the user being self-sovereign in using a wallet, generated DIDs.... but then and in context of ESSIF and support of "legally enforceable exchanges" and "supporting the Single Digital market" some registration (in a system that protects privacy) is needed.

— [153]

Hence, we further assume that EOSs satisfy the role description and do identify Alice in terms of some existing centralized system. It is not relevant to us whether ESSIF is actually SSI or not.

#### B.4.5.2. Trust model

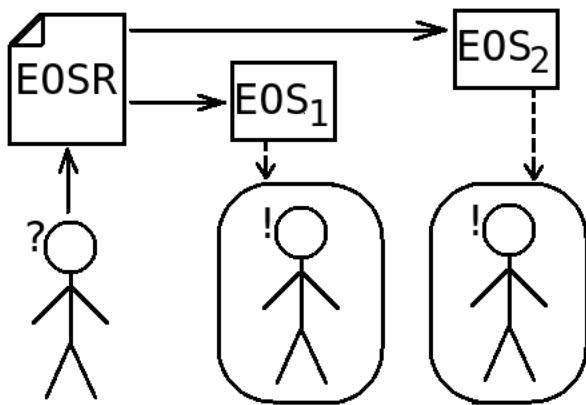
Let us now describe the trust model. First of all, each agent has to trust the EOSs who have the right to issue DIDs to natural persons and legal entities. If even a single EOS is compromised, we get the problem of impersonation. Such EOS entities are listed in a separate registry EOSR, which can be viewed as a kind of trusted list.

We get a trust model that is somewhat similar to the trusted list model, despite being based on a decentralized ledger. The difference comes from the working principles of EOSs and EOSR. Let us see how an ESSIF Trusted Registry (which also involves the EOSR) is defined.

A Trusted Registry is a (decentralized) data storage that contains vital information for the functioning of ESSIF. They are reliable data sources (single source of truth) that facilitate the establishment of trust between different parties within ESSIF based on different governance structures and policy frameworks (rules of conduct) for different functions and domains.

#### — [Nodes & Ledgers \(DID & Revocation/Endorsement Registries\) ESSIF v2 \(archived\)](#)

Coming back to the trusted list model of the classical PKI, we can think of EOSR as a trusted list stored on a distributed ledger. A distributed trusted registry allows us to relax the requirement of responsiveness from the EOSR and EOSs, since all their claims are now stored on a responsive ledger.



An EOS differs from a classical CA in the sense that it does not issue signed certificates directly to the users, but instead splits the statement about an agent controlling a public key into two statements. These are the statements about a DID controlling a public key (the EOS writes such statements to the DID registry), and about an agent holding a DID (evidence for such statements is given directly to the user in the form of a VC). Compared to classical PKI, this VC can be treated as a certificate issued by CA, and the difference is that VC alone is not yet the full certificate, but it has to be combined with information from the DID registry to actually get the public key.

#### B.4.5.3. Discussion

ESSIF is a complex framework that consists of many components that would be hard to re-implement and maintain locally. Nevertheless, for the purposes of our proposed construction in Sec. 3, we can pick up some useful ideas from the ESSIF model without actually implementing a distributed ledger. Instead, we can bring it to the classical CA model.

- ESSIF allows several DID-s, which basically allow an agent to have several credentials. We can have different CAs for different types of credentials, e.g. Mobile-ID and Smart-ID. If the CA for Smart-ID fails (either stops working, or has an integrity breach), then the users could temporarily switch to Mobile-ID.
- Information that has to be trusted (such as trusted lists and public keys) can be copied to several servers and/or secret-shared. For example, if an attacker impersonates a CA by showing a fake public key on behalf of the CA, the user can consult another server to double-check the correctness of the public key.

- In ESSIF, each agent consults the DID registry in order to learn whether a public key is still valid, or revoked. In a similar manner, a user consults a CA for the OCSP service. Such a service is sensitive to a denial-of-service attack, so we need to ensure the availability like in the previous point.

When proposing the construction in Sec. 3, we considered the possibility of splitting the statements about agents controlling public keys into statements about DIDs controlling public keys, and agents holding DIDs. We have not found any immediate advantages from such splitting in the context of Verifiable IDs (it would be useful for more general types of VCs). A disadvantage is, that splitting may lead to unexpected vulnerabilities that we have not had before, so we need to carefully revise the use cases. Let us consider at least some of such vulnerabilities.

- Splitting introduces an additional point of failure since it suffices to convincingly lie about only one of the two statements to successfully impersonate a user.
- If the VC is fully controlled by Alice, and a public key cannot be linked to Alice without the VC, then Alice can potentially repudiate her signatures by hiding the VC and saying that the DID is not hers. Thus, the signature must include either a copy of the VC of Alice, or at least some proof that the DID belongs to Alice.
- A Verifiable ID should always be issued together with the fresh DID, and should not be issued to previously registered DID-s. Otherwise, Alice and Bob may obtain two distinct Verifiable IDs that are related to the same DID. An ESSIF business scenario [154] actually considers this situation legitimate. Hence we need to disallow it, if we do not want that the same public key could belong to different natural persons, even if these persons agreed to share the same key voluntarily.

While these particular vulnerabilities are easy to mitigate, there could be other ones. We want to stress that one should be very careful when designing a DID-based solution.

Another interesting feature of SSI is, that each agent, including natural persons, is responsible for generating their own private keys. This could potentially avoid problems caused by the ID card manufacturer, like a weak key generation or generating the keys outside the ID card chip. We have not thoroughly analysed, what actually happens if we push the responsibility of key generation to the lay user. Eventually, the user still relies on the mobile phone manufacturers and service providers who create the random number generators (hardware) and the corresponding algorithms (software).

## B.5. Qualitative comparison

Let us now try to compare the trust models that we have seen. We will compare them from the perspective of usefulness, i.e. how well they match with our likely use-cases involving trust services. We will also compare them from the perspective of cost. In this section, we put forth a couple of scenarios covering the main use-cases of trust services. We will then discuss, how well different trust models support these scenarios.

### B.5.1. Signing a document

**Scenario.** In this scenario, parties  $A$  and  $B$  are in the process of signing a document  $D$ . Party  $A$  has received a data item  $\sigma$  from  $B$ , which the latter claims to be his signature on  $D$ . Party  $A$  is able to verify that cryptographically,  $\sigma$  is a signature on  $D$  with a public key  $pk$ . She also wants to make sure that  $pk$  is controlled by  $B$ , i.e.  $A$  will be later able to prove to some other party  $C$ , that  $\mathcal{C}(pk, B)$ .

#### B.5.1.1. Classical PKI

In classical PKI (including threshold PKI),  $A$  asks a CA whether the public key of  $B$  is  $pk$ . She is sure (i.e. she trusts) that  $C$  can do the same and will get the same answer.

**Verdict:** Suitable.

#### B.5.1.2. Web of Trust

In the plain web of trust model (based on direct and indirect trust, without additional constructions like blockchains), it may happen that the party  $C$  has not received any public keys from  $B$ , or has received a different key. Theoretically,  $A$  and  $C$  could have agreed beforehand on the public key of  $B$  e.g. during a step for establishing indirect trust, but there are no guarantees that  $C$  does not forget it.

**Verdict:** Not suitable.

#### B.5.1.3. Generic SSI

In generic SSI (without globally trusted third parties), the plain web of trust model is enhanced with a blockchain that allows keeping all claims of parties in a trusted log, so that  $A$  can inform  $C$  about the earlier claim made by  $B$ . The problem here is, that SSI solutions link  $pk$  to a DID which is not linked to  $B$ . Hence the signature can only be presented to  $C$  if he is convinced, and will not forget that this DID is owned by  $B$ . This is similar to remembering  $pk$  in the plain web of trust model. Nevertheless, this model is stronger if we can treat DID as the party identity, e.g. if  $B$  is a well-known university whose DID is known to everyone. So the party  $A$  may decide to some extent whether  $C$  will link the given DID to  $B$  at any point of time in the future.

**Verdict:** Suitable only if DIDs are seen as the true names of parties.

#### B.5.1.4. ESSIF

In ESSIF, a DID is linked to  $B$  using a special kind of verifiable credential (VC) issued by a trusted third party (EOS). Such a VC would need to be given by  $B$  to  $A$ , and presented to  $C$  as a part of the signature. This would work similarly to a classical PKI signature, and an EOS can be treated

similarly to a CA.

**Verdict:** Suitable.

### B.5.2. Signature verification

**Scenario.** In this scenario, party  $A$  holds a document  $D$ , presumably signed by  $B$  using the public key  $pk$ . She wants to verify the signature  $\sigma$ . The party  $A$  needs to convince herself that  $pk$  belongs to  $B$ .

#### B.5.2.1. Classical PKI

In classical PKI,  $A$  is going to consult the CA.

**Verdict:** Suitable.

#### B.5.2.2. Web of Trust

In the plain web of trust model, it may happen that the party  $A$  has not received any public keys from  $B$  at all, or has received keys different from  $pk$ .

**Verdict:** Not suitable.

#### B.5.2.3. Generic SSI

In generic SSI, the plain web of trust model is enhanced with a blockchain that allows keeping all claims of parties in a trusted log. Hence,  $A$  can check whether  $B$  has earlier claimed to anyone that he controls  $pk$ . The problem here is, that SSI solutions link  $pk$  to a DID which is not linked to  $B$ , and  $A$  is not necessarily convinced that DID is owned by  $B$ .

**Verdict:** Not suitable.

#### B.5.2.4. ESSIF

In ESSIF, a DID is linked to  $B$  using a special kind of verifiable credential (VC) issued by a trusted third party (EOS). Such a VC would need to be presented to  $A$  as a part of the signature. This would work similarly to a classical PKI signature, and an EOS can be treated similarly to a CA.

**Verdict:** Suitable.

### B.5.3. Authentication

**Scenario.** In this scenario, party  $B$  wants to authenticate itself to party  $A$ , presumably to set up a secure channel between them and to use the privileges that  $A$  affords to  $B$ .

For authentication between  $A$  and  $B$ , all considered trust models are acceptable, even the plain web of trust model. If  $A$  and  $B$  have not pre-shared their keys (e.g. using some other authentication methods), then the authentication will likely fail, but this is the expected behaviour.

Let us now add more details to this scenario. Suppose that a client  $C$  wants to establish connection with a server  $S$ . Server  $S$  needs to find out who  $C$  is, i.e. a certificate of  $C$  has to be found and

presented. We are interested in how the details of the scenario vary when varying the origin of  $S$  and  $C$ . Although all trust models are in principle sufficient, we need to think which of them are actually *implementable* and have *reasonable trust assumptions* for different origins of  $S$  and  $C$ . Obviously, we only care for the cases where at least one of  $S$  and  $C$  is Estonian.

#### B.5.3.1. Both the server and the client are Estonian

In this case, the interaction fully depends on our internal trust model, and we can choose any trust model that we like. We can even make use of ESSIF for internal communication.

#### B.5.3.2. The server is Estonian, the client is from EU

If  $S$  is Estonian and  $C$  is from EU, then the following trust models can probably be made to work.

- Server  $S$  may itself issue a certificate to  $C$  (this corresponds to the *shared domain* model in Sec. B.3.2). This can be a complicated procedure, something similar to issuing e-residence, but weaker.
- Both  $S$  and  $C$  may be using ESSIF.
- If  $C$  is certified by some foreign CA, then  $S$  may accept  $C$ 's certificate if one of the following holds about that CA:
  - It has performed a *mutual exchange* (Sec. B.3.3) of the certificates with  $S$ .
  - If belongs to some European *trusted list* like eIDAS.
  - It is certified by some trusted European CA (corresponding to the *hierarchical* and *bridge* models in Sec. B.4.1.2 and Sec. B.4.1.3).

#### B.5.3.3. The server is Estonian, the client is outside of EU

If  $C$  is outside of the EU, then the server will probably aim for a particular domain of  $C$ , since it is unlikely to trust the entire world. There are no global trusted lists and global RCA that would be a single point of failure for the entire world. It is likely that the Estonian server does not want to issue certificates to non-European clients, since identifying them may be too complicated (unless there is already positive experience with non-European e-residents). Let  $C$  be certified by some foreign CA. The server can accept  $C$ 's certificate if one of the following holds about that CA:

- Server  $S$  has performed a *mutual exchange* (Sec. B.3.3) of certificates with CA.
- A *mesh model* of trust (Sec. B.4.1.1) is in use where certificates from remote domains propagate on the basis of indirect trust between CAs.

#### B.5.3.4. The client is Estonian, the server is from EU

In this case  $S$  needs to be convinced that  $C$  can be trusted. As  $S$  can be from anywhere in the EU, we do not want to perform the *mutual exchange* (Sec. B.3.3) and send all local certificates to an external domain, as it would require too many connections. One of the following models is preferable.

- $C$  could obtain a certificate directly from some European CA (corresponding to the *shared domain* model of trust, Sec. B.3.2). This can be used in addition to already existing locally issued certificate. That is, in addition to Smart-ID etc. there could be some external European ID.

- $C$  could join ESSIF. This is in principle similar to the previous case.
- $C$  could be certified inside our internal trust model by an Estonian CA. This CA needs to undergo one of the following:
  - Be certified by some European CA (*hierarchical* model, Sec. B.4.1.2).
  - Be certified by some European *bridge* that connects multiple domains (Sec. B.4.1.3).
  - Be added to some European *trusted list* (Sec. B.3.4) like the one mandated by eIDAS.

#### B.5.3.5. The client is Estonian, the server is outside of EU

Similarly to previous case,  $S$  needs to be convinced that  $C$  can be trusted. Differently from previous case, here we target a particular  $S$ , as we are unlikely to be able to convince every server in the world: there are no global trusted lists and global RCAs that are trusted by everyone. We also probably cannot assume a bridge, since it is difficult to make an international bridge trusted by all involved domains. One of the following models is preferable.

- $C$  could obtain a certificate directly from some non-European CA (*shared domain* model of trust, Sec. B.3.2). There exist some globally recognized trust anchors, e.g. Google or Microsoft, but not every  $S$  might accept them.
- $C$  could be certified inside our internal trust model by an Estonian (or European) CA. This CA needs to undergo one of the following:
  - Perform *mutual exchange* (Sec. B.3.3) of the certificates of  $C$  with  $S$ .
  - Be part of a *mesh model* of trust (Sec. B.4.1.1), where certificates from remote domains propagate on the basis of indirect trust between CAs.

## B.6. Quantitative comparison

In this section we try to estimate the size of the effect of some components of trust infrastructure misbehaving, i.e. no longer satisfying the trust assumptions. Such misbehaviour may affect either integrity or availability of the trust infrastructure. To quantify the effect, we ask:

- How many attempts to find someone's public key, or to check whether a public key belongs to a party will fail by returning a wrong answer? (Integrity loss)
- How many attempts to find someone's public key, or to check whether a public key belongs to a party will fail by not returning anything at all? (Availability loss)

The number of failing queries obviously depends on the number of queries made in total. We may be more interested in the fraction of queries thus failing.

### B.6.1. Costs

There are many kinds of costs related to a trust infrastructure. We can characterize them as follows.

- Costs of the services that comprise the physical part of the trust infrastructure. These are split into

- Costs of setting up the necessary services (measured in monetary units);
- Costs of running the necessary services (measured in monetary units per time unit).
- Costs of establishing strong trust between components. We can treat it as the cost of cross-certification between components. A cross-certification reduces the possibility of abuse (by avoiding trivial attacks where a component is malicious from the beginning), but does not fully prevent it. In the current analysis we measure it in the number of needed cross-certificates.
- Availability losses resulting from the failure of pieces of infrastructure. The cost materializes as the inability of a party to determine the public key of another party either now or in the past. It can be measured by characterizing the set of pairs of parties, where the first one was unable to determine the public key of the second one. There are a number of sensible measures applicable to this set, as well as a number of sensible assumptions that can and should be made about this set. As we see below, we will simplify these measures and assumptions as much as possible, in order to make the analysis more tractable.
- Integrity losses resulting from the failure of pieces of infrastructure. The cost materializes as a party learning the public key of another party, which actually isn't that other party's public key. Similarly to availability losses, it can be characterized by the set of pairs of parties, where the first one learned the public key of the second party incorrectly. Again, several measures are assumptions make sense, and we make them as simple as possible for the purposes of our analysis.

We recognize that not all parties are equal, and faking one signature or masquerading as a certain person may be much more profitable than for some other person. We think that it is very hard to properly model these inequalities, especially if we want to do it quantitatively. Rather, we assume that with the increase in importance comes the increase in the capability of weathering the failures, and this cancels out the higher interests to attack this particular person.

Beside the costs of setting up the system, and costs of the attacks, there are also costs for the attacker to interfere with the components of the system. If the system has homogeneous topology, then we may characterize an attack by the fraction of the system that the attacker has taken over. If there are components with different roles, then we can also state, which components the attacker has compromised.

### B.6.2. Attack model

We consider the setting where a user  $i$  is making a query about some other user's public key, i.e. wants to learn whether  $\mathcal{C}(pk, j)$  holds for a certain user  $j$  and a public key  $pk$ . Each user is directly trusting a particular CA (or several different CA-s). The CA-s are indexed as  $CA_1, \dots, CA_n$ . We call the set of all users trusting a certain  $CA_k$  the *domain* of  $CA_k$ , denoted  $dom(k)$ . Depending on the trust model, the domains may intersect. We consider sets of queries  $Q_{i,j}$ , where a query  $q \in Q_{i,j}$  is made by a user  $i' \in dom(i)$  about  $\mathcal{C}(pk, j)$  for some  $j' \in dom(j)$ . It is possible that  $i=j$ .

Assume that the attacker has managed to corrupt  $k$  components in the system, and is now waiting for user queries. Let us quantify the attack impact in terms of the expected fraction of queries that will fail due to the attack (e.g. until the corruption will be detected and fixed). We consider the success in breaking the availability as well as the integrity.

Define the following quantities:

- $n$  is the total number of CAs;
- $k$  is the number of corrupted CAs and other components of the trust infrastructure;
- $t$  is the threshold parameter, when using  $(n,t)$ -threshold cryptography (relevant for threshold models only);
- $c_X$  is the cost of setting up and running the service  $X$ , where  $X$  can be one of CA, VA (validation authority), TL (trusted list), BC (blockchain), KSI, EOS, EOSR (the latter two are relevant only to ESSIF);
- $s_X$  is the cost of setting up (but not running) the service  $X$ , where  $X$  can range over the same services as above.

In the analysis, we assume that the query is equally likely to come from the domain any particular CA, i.e. with the probability  $1/n$ . We also assume that the query is equally likely targeted towards any particular CA's domain, i.e. again with the probability  $1/n$ .

In practice, the impacts of an attack can only be stronger for a non-uniform distribution over CAs, as the attacker may always corrupt the CA that affects most of the users. In the worst case, all the users belong to a single CA domain, and adding more CAs has no effect because their domains are empty. Since the worst-case user distribution falls back to classical PKI in most models (except the threshold PKI) and hence is not interesting, we will be considering uniform distribution of users over the CAs.

### B.6.3. Analysis results

The impacts of the attack, as well as costs of setting up and running the trust infrastructure are summarized in the following table.

Trust model	Impact on availability	Impact on integrity	Setup and running costs	No. of cross-certs
Dedicated domain	1	1	$c_{CA}$	0
Shared domain	1	1	$c_{CA}$	0
Mutual exchange	$k / n$	$k / n$	$n \cdot c_{CA}$	$n \cdot (n-1)$
Trusted lists	$k / n$	$k / n$ , if CA is corrupt 1, if TL is corrupt	$n \cdot c_{CA} + c_{TL}$	$n$
Hierarchical	Between $k / n$ and 1	Between $k / n$ and 1	$n \cdot c_{CA}$	$n-1$
Bridge CA	$(1-1/n)+k/n^2$ , if BCA is corrupt $k / (n-1)$ , otherwise	1	$n \cdot c_{CA}$	$2 \cdot (n-1)$
Bridge VA	$(1-1/n)+(k-1)/n^2$ , if BVA is corrupt $k / n$ , otherwise	1	$n \cdot c_{CA} + c_{VA}$	$n$

Trust model	Impact on availability	Impact on integrity	Setup and running costs	No. of cross-certs
General mesh	Between $k / n$ and $(1 - 1/n) + k/n^2$	1	$n \cdot c_{CA}$	Between $2 \cdot (n-1)$ and $n \cdot (n-1)$
Threshold	0, if $k \leq n-t$ 1, otherwise	0, if $k \leq t$ 1, otherwise	$n \cdot c_{CA}$	$n \cdot (n-1)$
Cold standby	1 (if $k=1$ )	1 (if $k=1$ )	$c_{CA} + s_{CA}$	0
Hot balancing	1 / 2 (if $k=1$ )	1 / 2 (if $k=1$ )	$2 \cdot c_{CA}$	0
Active $\times$ 2	1 (if $k=1$ )	0 (if $k=1$ )	$2 \cdot c_{CA}$	2
Active $\times$ 3	0 (if $k=1$ )	0 (if $k=1$ )	$3 \cdot c_{CA}$	6
KSI enhancement	0	the same as in the original model	$\dots + c_{KSI}$	$\dots + n$
Web of Trust	from 0 to 1	from 0 to 1	$c_{BC}$	0
ESSIF	$k / n$	$k / n$ , if an EOS is corrupt 1, if EOSR is corrupt	$c_{BC} + n \cdot c_{EOS} + c_{EOSR}$	0

Let us discuss how these numbers were obtained.

#### B.6.3.1. Dedicated domain

In a classical model with a single CA, we have a single point of failure. Corrupting the CA fully breaks both availability and integrity.

#### B.6.3.2. Shared domain

In a shared domain model, we still have a single CA trusted by several user domains, which in our analysis is treated as a single logical domain. Corrupting the CA fully breaks both availability and integrity.

#### B.6.3.3. Mutual Exchange

In this trust model, each user has chosen one of the  $n$  CA-s to trust. Consider a single query made by a certain user. If the attacker has corrupted  $k$  out of  $n$  CA-s, then with probability  $k/n$  the set of corrupted CA-s includes the CA that this particular user trusts. Such a corruption breaks both availability and integrity. Since CA-s do not trust each other, corrupting  $k$  of them does not have impact on the other  $n-k$  of them.

Every pair of domains needs to somehow cross-certify each other before agreeing to accept each other's certificates, thus the number of cross-certificates given in the table.

#### B.6.3.4. Trusted Lists

The CA-s in a trusted list do not cross-certify each other, so if one of them fails, the remaining CAs will not help in approving the credentials issued by the failed CA. Hence, we can assume that the availability and integrity are the same as for the mutual exchange for a single query. In addition, if the attacker corrupts a trusted list provider, he will be able to fraudulently reply all queries that rely on that list.

Regarding the number of cross-certificates: each CA needs to certify itself to the trusted list.

#### B.6.3.5. Hierarchical PKI

In this model, the nature of corrupted CAs is very important. If an attacker corrupts a single CA, it affects all CAs that have trusted it, i.e. all descendants of that CA in the hierarchy.

- The worst case is, when the attacker corrupts the RCA, which is the common anchor of trust for all the CAs. In terms of integrity, any statement can be faked. In terms of availability, all other CAs need to refer to RCA to prove their trustworthiness, so there is also no availability without RCA.
- The best case is, when the attacker corrupts only the leaf nodes, since it only spoils the queries concerning the corrupted CAs. A query is related to a corrupted CA with the probability  $k/n$ .

Regarding the number of cross-certificates: every CA that is not RCA must certify itself to the parent CA.

#### B.6.3.6. Bridge CA

If the attacker corrupts the BCA, then all connections between CA-s are lost, and from that point onward, a user can make queries only to its local CA. Moreover,  $k$  of these CAs (including the BCA) will be corrupted as well. With probability  $n / n^2 = 1/n$ , the query that a user makes is related to the local CA of that user, which is uncorrupted and thus available with probability  $(1-k/n)$ . The product of these two quantities is the probability that an arbitrary query will be answered. The complement of this value is the impact on availability, if the BCA is corrupted.

If the attacker corrupts a non-BCA, then he only disrupts the availability of the users belonging to that CA. However, in terms of integrity, corrupting any CA affects everyone due to mutual trust.

Every CA needs to be certified by the BCA, and it in turn needs to certify the BCA, thus the number of cross-certificates.

#### B.6.3.7. Bridge VA

While the BVA is formally not a CA, we can analyse it similarly as a gateway between CAs. Similarly to the previous case, the corruption of BVA breaks the connection between CAs. We thus get numbers that are similar to the previous case. But as the VA is not a CA, then only  $(k-1)$  CAs are corrupted, if VA is corrupted.

The number of cross-certificates is smaller than in the previous case, because we assume that the BVA is trusted by the CA-s as an authority and does not need separate cross-certification.

### B.6.3.8. General mesh

The structure of the mesh is very important for the impact on availability.

- In the worst case, i.e. when the mesh is a star (i.e. a central hub with spokes), the attacker may corrupt the central node, resulting each single remaining CA to be a separate unconnected component. The users of the  $k$  corrupted CAs will not get any of their queries answered, and the others now can only communicate within their own domains. This is analogous to the Bridge CA model with a removed bridge.
- In the best case, the removed  $k$  CA-s are not critical, and only their local users are affected. This is again similar to Bridge CA model when the bridge is not affected.

The impact on integrity is full in any case due to propagated trust.

All connected CA-s need to cross-certify each other, and there are from between  $2(n-1)$  and  $n(n-1)$  directed connections.

### B.6.3.9. Threshold

In  $(n,t)$ -threshold sharing schemes, the attacker that corrupts less than  $t$  components, can affect neither the integrity nor the availability of the system. But if he corrupts at least  $t$  components, then we have no guarantees anymore. A good property of this model is that the guarantees do not depend on the distribution of queries.

All CAs need to cross-certify each other, as in the full mesh model.

### B.6.3.10. Double CA

In the SPoF analysis [1], Cybernetica discussed some particular solutions based on two CAs. Different variants correspond to different generic trust models. Let us discuss the impact of corruption on different variants. As the number of CAs is two, and corrupting both of them will fully impact both availability and integrity, let us estimate the impact for  $k=1$ .

#### B.6.3.10.1. Cold standby

Here the second CA is just a reserve for the first one. If the first CA gets corrupted, all its certificates are cleared, and the reserve CA is put into use. Until the corruption is detected, the attacker may influence as many queries as for a single CA (dedicated domain and shared domain models). As mentioned in [1], the goal of this solution is to reduce the time it takes to resolve the situation, but not to prevent abuse. Here we only count the setup cost of the reserve CA, since we need to run it only when the first one fails.

#### B.6.3.10.2. Hot balancing

Here the credentials are uniformly split between the two CA-s. This is an instance of the mutual exchange model with  $k=1$  and  $n=2$ , and corruption of a single CA affects one half of the queries.

#### B.6.3.10.3. Active-active

Here all credentials need to be approved by both CA-s. While the failure of one CA will stop the system, it will not be enough to affect the integrity. This set-up is the same as the  $(2,2)$ -threshold

scheme.

#### B.6.3.10.4. Active-active-active

If we increase the number of CA-s to three, and require that any credential is be approved at least by two out of three CAs, then we get a simple (3,2)-threshold scheme. Now a single corruption cannot affect neither availability nor integrity, but maintaining three CAs requires more resources than two CAs.

#### B.6.3.11. KSI

KSI can support any of the previous trust models, letting all CAs use keyless signatures for their certificates. The main advantage of KSI is, that we can guarantee availability. Since corrupted CAs are free to issue fake certificates, it does not provide additional guarantees for integrity. The effort required to break KSI depends on its implementation and is out of the scope of this analysis. We just assume that it is hard, and do not take it into account.

Regarding the number of cross-certificates: KSI needs to certify all  $n$  CAs of the underlying system.

#### B.6.3.12. Web of Trust

We cannot provide a generic analysis of such models, since they can be very different. For example, all CA models can be seen as special cases of the indirect trust model, where a CA is seen as a "friend" of the users who belong to its domain. For example, if Bob is the only person who belongs to two mostly non-overlapping social groups, then he can be seen as a Bridge CA. If a teacher Alice arranges the sharing of credentials between her students, then she can be treated as a classical CA.

On the other hand, a special case of indirect trust is the direct trust, for which the attacker's impact is 0 since Alice and Bob exchange credentials using a separate channel (like meeting physically). Hence, any impact between 0 and 1 is possible in this model.

#### B.6.3.13. ESSIF Model

This model assumes a permissioned ledger, so the impact is easier to estimate than for the generic web of trust model. Assuming that the blockchain does not fail, this model can be treated similarly to the trusted list model, where  $n$  now denotes the number of EOS services. The effort required to break the blockchain depends on its implementation and is out of the scope of this analysis.

## B.7. High-level comparison from the legal perspective

In this section, we will do a very high-level comparison of different trust models in terms of legal aspects. We will *not* do a formal legal analysis, but rather mention the questions that should be considered when doing such an analysis. We give a more detailed analysis for the scheme we propose in Chapter 4.

We will only briefly discuss compliance with eIDAS and GDPR here. For the description of a full system, we would likely need to take into account a variety of laws, such as contract law, consumer protection law, etc., as described in Appendix C.

### B.7.1. Classical PKI

The classical PKI models (or Mesh PKI models without thresholds) seem to be already covered by eIDAS, and the particular trust relations between CAs do not seem to be relevant. There are certain providers that belong to the eIDAS trust list, and they in turn may issue certificates to subordinate CAs. However, if we want to avoid a single point of failure, then all CAs cannot rely on a single member of the trust list, as compromising that member will compromise the entire system with its numerous CAs.

There seem to be no issues with GDPR, as long as one can only ask from a CA whether the claim  $\mathcal{C}(pk, A)$  is true, but cannot query for the values of  $pk$  and  $A$ .

### B.7.2. Mesh PKI with thresholds

On the abstract level, all the physical CAs participating in the mesh comprise a single CA, which should be approved by eIDAS. It is not clear if it is sufficient that every participating CA alone satisfies the eIDAS, but it looks reasonable since if all components of a CA are honest, then so is the single CA that they implement, regardless of the used threshold sharing scheme. It is less clear whether a distributed signature is accepted by eIDAS, or we need a more complicated procedure to make it accepted. Apparently eIDAS does provide a deep treatment of technical details, and just requires that the signature format is compatible with an accepted standard signature scheme.

In terms of GDPR, we can potentially do even better than in the case of classical PKI, since the identifying data does not even need to be stored on a single CA, but can be distributed among multiple CAs using some secret sharing scheme, so that no single CA sees the data. However, the cryptographic details of this approach can be more difficult to implement efficiently.

### B.7.3. KSI

This service can provide better availability by logging certificates onto a blockchain. We are not sure if something like KSI is currently regulated by eIDAS. Perhaps it is not relevant in the case when we are using KSI just as a support for a CA that makes it more reliable.

In relation to GDPR, we need to be careful that any information that is stored in a distributed ledger is not identifying. A generic solution is to provide every committed hash with some randomness (salt) in such a way that only the owner of the salt may reveal what has actually been logged. It is plausible that a more detailed legal analysis and comparison of KSI against GDPR already exists, since it would benefit the technology and service providers.

### B.7.4. SSI

The SSI approach can theoretically be out of the scope of GDPR since it is using artificial decentralized identifiers (DIDs) that are not linked to anything else. However, for practical solutions, we still need a verifiable credential (VC) that links a DID to a natural person or a legal entity. This seems fine, as whenever Alice needs to prove her identity to Bob, she herself presents to him the VC, which can be treated similarly to presenting a classical PKI certificate. The underlying blockchain should also satisfy eIDAS.

ESSIF aims to be compatible with both eIDAS and GDPR:

ESSIF aims to be compliant with GDPR as well as aligned with eIDAS to ensure that ESSIF can benefit from existing legal frameworks, allowing ESSIF to provide digital evidences providing support to legal enforceability.

— [152]

However, it looks like the version 1 of eIDAS is not yet sufficient for SSI solutions, and we need to wait for version 2. Also, SSI should probably be rather studied in the context of [Sovrin Governance Framework \(10.04.2022\)](#).

# C. Appendix: Applicable laws

## C.1. Introduction

This Annex provides an overview of the legal framework that affects current and future trust models for the electronic identification system and infrastructure in Estonia. The legal environment that governs the operation of any identity system consists of three different levels of legal rules [155]:

1. general law — applies generally to all business and personal activities, not written with identity systems in mind (e.g. contract law, tort law, privacy law, consumer protection law etc);
2. generic identity system law — law written specifically to govern identity systems generally, typically apply to all identity systems within a jurisdiction, e.g. Virginia's Electronic Identity Management Act [156], or the Draft Provisions on the Cross-border Recognition of IdM and Trust Services developed by the UN Commission on International Trade Law (UNCITRAL) [157]);
3. individual identity system rules — the set of system-specific rules written to govern the operation of a particular identity system. Within this level, there are two subcategories of identity systems:
  - a. private sector identity systems — the legal rules are typically contract-based, often referred to as a trust framework or system rules, and apply only to those system participants who have contractually agreed to be bound to them (e.g. [the SAFE-BioPharma Trust Framework \(15.11.2021\)](#), [the Sovrin Governance Framework \(15.11.2021\)](#)).
  - b. government identity systems — the legal rules are often embodied in a law or regulation enacted by the government, and thus automatically apply to all those who participate in the identity system (e.g. the Identity Documents Act in Estonia, the Aadhaar Act in India). Some government identity systems also use contract-based trust frameworks, such as the [Trusted Digital Identity Framework \(TDIF\) \(26.11.2021\)](#) for the Australia national federated identity system.

## C.2. Applicable law

The focus of this section is defining the legal rules applicable to trust frameworks used in the electronic identity management system and related infrastructure established in Estonia for the public services. In the following sections, we shall map the relevant legal acts in the three different levels described above.

### C.2.1. Individual identity system rules

Under Estonian national law, the Estonian government identity system is regulated by the following legal acts:

1. Identity Documents Act (IDA) [11]
2. Electronic Identification and Trust Services for Electronic Transactions Act (EITSETA) [158]
3. Emergency Act (EA) [12]

#### 4. Statutes of the Estonian Information System Authority (SEISA) [159]

The legal acts are further supported by contractual measures concluded between the relevant stakeholders in the Estonian identity system (e.g. the different [Conditions for Use of Certificates by SK ID Solutions \(29.11.2021\)](#)).

### C.2.2. Generic identity system law

In the European Union (EU), the eIDAS Regulation [2] was adopted in 2014 to [ensure \(link checked 29.11.2021\)](#) that people and businesses can use their own national electronic identification schemes (eIDs) to access public services available online in other EU countries. It has been implied that the eIDAS Regulation functions as an individual identity system [155 p. 3]. The authors of this paper do not support this view.

The eIDAS Regulation establishes the cross-border recognition of national electronic identification schemes notified by the Member States [160 p. 9]. Therefore, the eIDAS Regulation, as it currently stands, does not create a separate identity system but merely facilitates the mutual recognition of existing national identity systems if the relevant Member States have requested for this. This is supported by Recitals 12 and 13 of the eIDAS Regulation:

One of the objectives of this Regulation is to remove existing barriers to the cross-border use of electronic identification means used in the Member States to authenticate, for at least public services. This Regulation does not aim to intervene with regard to electronic identity management systems and related infrastructures established in Member States. The aim of this Regulation is to ensure that for access to cross-border online services offered by Member States, secure electronic identification and authentication is possible.

— Recital 12 [2]

Member States should remain free to use or to introduce means for the purposes of electronic identification for accessing online services. They should also be able to decide whether to involve the private sector in the provision of those means. Member States should not be obliged to notify their electronic identification schemes to the Commission. The choice to notify the Commission of all, some or none of the electronic identification schemes used at national level to access at least public online services or specific services is up to Member States.

— Recital 13 [2]

In essence, these Recitals confirm that the eIDAS Regulation does not interfere with national identity systems as such. However, several authors have pointed out that the eIDAS Regulation uses legal terms that are difficult to interpret and thus cause conflicting approaches and

implementations throughout the Member States. It will require further legal analysis to determine the extent to which the eIDAS Regulation can set binding legal requirements to national identity systems in Member States who have notified their electronic identification schemes.

For these reasons, we categorise the eIDAS Regulation as a type of generic identity system law, belonging to the second level of legal rules governing the operation of any identity system. This classification of eIDAS Regulation will be followed throughout the rest of the document.

In addition to the eIDAS Regulation, there are ongoing projects for introducing new legislative drafts and different kinds of soft law, which may influence the interpretation of current applicable law or even become new law in the future. For example:

1. Council of Europe Convention 108 Draft Guidelines on Digital Identity [161],
2. UNCITRAL Draft Provisions on the Use and Cross-border Recognition of Identity Management and Trust Services [157],
3. European Commission Proposal for a regulation establishing a framework for a European Digital Identity (EDIR) [162].

### C.2.3. General law

There are several legal acts both nationally, at EU level and internationally that cover different aspects of a national identity system. These acts may apply to the Estonian national identity system only partially, depending on their scope of application. Considering the variety of subject matters that a national identity system needs to address, relevant legal acts from different domains may come to play:

1. Data protection (e.g. Personal Data Protection Act [163], General Data Protection Regulation [4], Convention 108 [164]);
2. Cybersecurity (e.g. Cybersecurity Act [165], Directive on security of network and information systems [166], Directive on attacks against information systems [167], Budapest Convention [168]);
3. Interoperability (e.g. Public Information Act [169] Single Digital Gateway Regulation [170]);
4. Other.

## D Lisa: Väljapakutud arhitektuuri kvantitatiivne analüüs

Käesolevas lisas kirjeldame täpsemalt, kuidas arhitektuurikirjelduses toodud kvantitatiivne analüüs tehtud on. Seejuures anname hinnanguid veidi üldsemate mudelite jaoks. Võrdleme ründaja edukust järgmiste usaldusmudelite korral:

- *Üksik CA* on lihtne usaldusmudel üheainsa TSP ja tema osaks oleva CA-ga. Valisime selle mudeli võrdluseks, sest et seda kasutatakse praegu.
- *1-sert TL* on tavaline usalduslisti mudel  $n$  erineva CA-ga, kus kasutaja saab endale sertifikaadi täpselt ühelt CA-lt.
- *m-sert TL* on tavaline usalduslisti mudel  $n$  erineva CA-ga, kus kasutaja saab endale sertifikaadi  $m$  erinevalt CA-lt. Teenuse osutamiseks sobib ükskõik milline neist  $m$  sertifikaadist.
- *(3,2)-lävi* on väljapakutud arhitektuuri peatükis kirjeldatud põhilahendus. Kvantitatiivses analüüsides me ei erista I ja II usaldusklassi TSP-sid, sest need määradavad ainult seda, kas CA-d võivad sattuda ühe ja sama ründaja kontrolli alla. Käesolevas analüüsides on aga meil ühe ja sama ründaja kontrolli all olevate CA-de arv niikuinii üheks parameetriks. Eeldame, et süsteemis võib olla  $n \geq 3$  erinevat CA-d ning kasutaja sertifikaatide kolmik võib olla väljastatud erinevate CA-de kombinatsioonide poolt.
- *(2,1)-lävi* on väljapakutud arhitektuuri peatükis kirjeldatud põhilahenduse lihtsustatud versioon. Eeldades, et süsteemis võib olla  $n \geq 2$  erinevat CA-d, saame täpselt samu hinnanguid mis *2-sert TL* mudeli korral.

Ründaja edukust käideldavuse (AV) ja tervikluse (INT) murdmisel hindame arvuga vahemikus  $0 \dots 1$ , mis väljendab mõjutatud kasutajate osakaalu alates mitte kellestki (0) kuni kõigini (1). Olgu süsteemid  $n$  erinevat CA-d, millest ründajal õnnestus enda kontrolli alla saada  $k$  tükki. Tabelites D.1 ja D.3 toodud avaldised põhinevad järgmistel arvutustel:

- Olgu kasutajal  $m$  sertifikaati. Arvutame tõenäosuse, et ründaja suudab kõigi nende väljaandmist mõjutada. Olgu ründaja kontrolli all  $k \geq m$  erinevat CA-d. Siis tõenäosus, et kõik  $m$  sertifikaati on väljastatud nende  $k$  ülevõetud CA poolt, on järgmine:

$$\frac{k}{n} \cdot \frac{k-1}{n-1} \cdots \frac{k-m+1}{n-m+1} = \frac{k(k-1)\cdots(k-m+1)}{n(n-1)\cdots(n-m+1)} \in \Theta\left(\frac{1}{n^m}\right).$$

- Olgu kasutaja käes 2 sertifikaati. Arvutame tõenäosuse, et ründaja suudab neist vähemalt väljaandmist mõjutada. Oletame, et ründaja kontrolli all on  $k$  erinevat CA-d. Pöörame nüüd tähelepanu ühele neist kahest sertifikaadist.

- Tõenäosusega  $\frac{k}{n}$  on see “halb”. Rünne on õnnestunud.
- Tõenäosusega  $\frac{n-k}{n}$  on see “hea”. Ründe õnnestumiseks peab nüüd teine sertifikaat olema kindlasti “halb”. Kui  $(n-1)$ -st ülejäänud CA-st on  $k$  tükki ründaja kontrolli all, siis juhtub see tõenäosusega  $\frac{k}{n-1}$ .

Kokku saame tõenäosuseks

$$\frac{k}{n} + \frac{n-k}{n} \cdot \frac{k}{n-1} = \frac{(n-1)k + (n-k)k}{n(n-1)} = \frac{k(2n-k-1)}{n(n-1)} \in \Theta\left(\frac{1}{n}\right).$$

- Olgu kasutaja käes 3 sertifikaati. Arvutame tõenäosuse, et ründaja suudab neist vähemalt ühe väljaandmist mõjutada. Oletame, et ründaja kontrolli all on  $k$  erinevat CA-d. Pöörame nüüd tähelepanu kahele neist kolmest sertifikaadist.

- Kui vähemalt üks neist on “halb” (selle tõenäosus on  $\frac{k(2n-k-1)}{n(n-1)}$ , nagu leidsime ülalpool), siis on rünne õnnestunud.
- Kui mõlemad on “head” (selle tõenäosus on  $1 - \frac{k(2n-k-1)}{n(n-1)}$ ), siis võib rünne ikkagi õnnestuda juhul, kui kolmas on halb. Kui  $(n-2)$ -st ülejäänenud CA-st on  $k$  tükki ründaja kontrolli all, siis juhtub see tõenäosusega  $\frac{k}{n-2}$ .

Kokku saame tõenäosuseks

$$\begin{aligned} \frac{k(2n-k-1)}{n(n-1)} + \left(1 - \frac{k(2n-k-1)}{n(n-1)}\right) \cdot \frac{k}{n-2} \\ = \frac{k(3n^2 - 3nk - 6n + 3k + k^2 + 2)}{n(n-1)(n-2)} \in \Theta\left(\frac{1}{n}\right). \end{aligned}$$

- Kui kasutaja peab esitama kaks sertifikaati kolmest, siis on halb juht see, kui neist sertifikaatidest vähemalt kahe väljaandmine on toimunud ründaja kontrolli all. Arvutame tõenäosuse, et see nii on. Pöörame nüüd tähelepanu ühele neist kolmest sertifikaadist.

- Tõenäosusega  $\frac{n-k}{n}$  on see sertifikaat “hea”. Ründe õnnestumiseks peavad nüüd mõlemad ülejäänenud sertifikaadid olema “halvad”. Nagu eelnevalt analüüsime: kui  $(n-1)$ -st CA-st on  $k$  tükki ründaja kontrolli all, siis juhtub see tõenäosusega  $\frac{k}{n-1} \cdot \frac{k-1}{n-2}$ .
- Tõenäosusega  $\frac{k}{n}$  on see sertifikaat “halb”. Ründe õnnestumiseks peab nüüd vähemalt üks ülejäänenud kahest sertifikaadist samuti olema “halb”. Nagu eelnevalt analüüsime: kui  $(n-1)$ -st CA-st on  $(k-1)$  tükki ründaja kontrolli all, siis juhtub see tõenäosusega  $\frac{k-1}{n-1} + \frac{(n-1)-(k-1)}{n-1} \cdot \frac{k-1}{n-2}$ .

Kokku saame tõenäosuseks

$$\begin{aligned} \frac{n-k}{n} \cdot \frac{k}{n-1} \cdot \frac{k-1}{n-2} + \frac{k}{n} \cdot \left( \frac{k-1}{n-1} + \frac{(n-1)-(k-1)}{n-1} \cdot \frac{k-1}{n-2} \right) \\ = \frac{k(k-1)}{n(n-1)} \cdot \left( \frac{n-k}{n-2} + 1 + \frac{(n-1)-(k-1)}{n-2} \right) \\ = \frac{k(k-1)(3n-2k-2)}{n(n-1)(n-2)} \in \Theta\left(\frac{1}{n^2}\right). \end{aligned}$$

Tervikluse hindamisel eeldame, et sertifikaadi võltsimine annab ründajale võimaluse ükskõik kellena esineda. Seega on terviklus alati kas 0 või 1. Tervikluse ründamiseks peaks ründaja suutma luua piisava arvu võltsitud sertifikaate, mis on  $(3, 2)$ -lävimudeli korral 2 ning iga teise analüositud mudeli korral 1.

Tablis D.2 on toodud mudelite omadused, mis võivad olla seotud nende realiseerimise ja ülalpidamise kuludega. Täpsemat rahalist hinnangut me selles osas anda ei oska.

**Kasutajate jagamine sõltumatuteeks domeenideks.** Kui CA-de arv  $n$  on stabiilne, siis võib jagada kasutajad  $n$ -ks mittekattuvaks domeeniks, millest igaühes kasutatakse vaid üht konkreetset CA-d. Näiteks, kui ründaja saab enda kontrolli alla Tartu elanikke teenindava

Tabel D.1: Kvantitatiivne ründaja edukus erinevate eelduste ja mudelite korral.

Rikutud CA-de arv		0	1	2	$k > 2$	sõltuvus $n$ -ist
Üksik CA	AV	0	1	1	1	1
	INT	0	1	1	1	1
1-sert TL	AV	0	$\frac{1}{n}$	$\frac{2}{n}$	$\frac{k}{n}$	$\Theta(1/n)$
	INT	0	1	1	1	1
2-sert TL	AV	0	0	$\frac{2}{n(n-1)}$	$\frac{k(k-1)}{n(n-1)}$	$\Theta(1/n^2)$
	INT	0	1	1	1	1
$m$ -sert TL $(m > 2)$	AV	0	0	0	$\frac{k(k-1)\cdots(k-m+1)}{n(n-1)\cdots(n-m+1)}$	$\Theta(1/n^m)$
	INT	0	1	1	1	1
(3,2)-lävi	AV	0	0	$\frac{6}{n(n-1)}$	$\frac{k(k-1)(3n-2k-2)}{n(n-1)(n-2)}$	$\Theta(1/n^2)$
	INT	0	0	1	1	1
(2,1)-lävi	AV	0	0	$\frac{2}{n(n-1)}$	$\frac{k(k-1)}{n(n-1)}$	$\Theta(1/n^2)$
	INT	0	1	1	1	1

Tabel D.2: Erinevate mudelite omadused, mis võivad olla seotud kuludega.

	sertifikaate kasutaja kohta	CA-de arv	muudatused RP loogikas
Üksik CA	1	1	ei
1-sert TL	1	$\geq 2$	ei
2-sert TL	2	$\geq 2$	ei
3-sert TL	3	$\geq 3$	ei
(3,2)-lävi	3	$\geq 3$	jah

CA, siis ei saa ta selle abil vältida Tallinna elanike sertifikaate.  $2$ -sert TL mudelite korral peab kasutajal olema 2 erinevat sertifikaati, nii et kasutajad saab jagada  $\binom{n}{2} = n(n - 1)/2$  domeeniks. (3,2)-lävi ja 3-sert TL mudelite korral peab kasutajal olema 3 erinevat sertifikaati, nii et kasutajad saab jagada  $\binom{n}{3} = n(n - 1)(n - 2)/6$  domeeniks.

Tabelis D.3 on analoogiliselt Tabeliga D.1 toodud eri mudelite käideldavus- ja terviklushinnangud, arvestades et kasutajad on ära jagatud domeenidesse vastavalt neid teenindavatele CA-dele. Terviklushinnangute arvutamisel peame nüüd  $m$ -sert TL mudelite korral arvutama töenäosust, et vähemalt üks  $m$  sertifikaadist on "halb". (3,2)-lävimudeli korral on tervikluse ründamine sama keeruline kui käideldavuse ründamine (ründaja kontrolli all peab olema vähemalt 2 sertifikaati 3-ist). Sarnaselt varasemaga on (2,1)-lävimudel samaväärne 2-sert TL mudeliga.

Kui süsteemis on kokku  $n$  CA-d, siis on  $(n, t)$ -lävimudelite korral kõik kasutajad ühes domeenis. Samas võime uurida alternatiivset jaotust, kus üks domeen vastab konkreetsele  $t$ -st CA-st koosnevale komplektile. See annab kokku  $\binom{n}{t}$  erinevat domeeni. Domeeni kasutaja tohib esitada vaid nende konkreetsete CA-de väljastatud sertifikaate. Selline rangem jaotus lubab parandada terviklust veelgi, kuid vähendab selle eest käideldavust, sest nüüd on igale kasutajale rangelt määratud  $t$  CA-d, kellelt ta tohib saada sertifikaate. Kui ründaja on üle

Tabel D.3: Kvantitatiivne ründaja edukus erinevate eeldustega korral, eeldusel, et kasutajad on jaotatud  $n$  domeeniks, kus iga domeen võib kasutada ühe konkreetse CA teenust.

Rikutud CA-de arv		0	1	2	$k$	sõltuvus $n$ -ist
Üksik CA	AV	0	1	1	1	1
	INT	0	1	1	1	1
1-sert TL	AV	0	$\frac{1}{n}$	$\frac{2}{n}$	$\frac{k}{n}$	$\Theta(1/n)$
	INT	0	$\frac{1}{n}$	$\frac{2}{n}$	$\frac{k}{n}$	$\Theta(1/n)$
2-sert TL	AV	0	0	$\frac{2}{n(n-1)}$	$\frac{k(k-1)}{n(n-1)}$	$\Theta(1/n^2)$
	INT	0	$\frac{2}{n}$	$\frac{2(2n-3)}{n(n-1)}$	$\frac{k(2n-k-1)}{n(n-1)}$	$\Theta(1/n)$
3-sert TL	AV	0	0	0	$\frac{k(k-1)(k-2)}{n(n-1)(n-2)}$	$\Theta(1/n^3)$
	INT	0	$\frac{3}{n}$	$\frac{6(n-2)}{n(n-1)}$	$\frac{k(3n^2-3nk+3k+k^2+2-6n)}{n(n-1)(n-2)}$	$\Theta(1/n)$
(3,2)-lävi	AV	0	0	$\frac{6}{n(n-1)}$	$\frac{k(k-1)(3n-2k-2)}{n(n-1)(n-2)}$	$\Theta(1/n^2)$
	INT	0	0	$\frac{6}{n(n-1)}$	$\frac{k(k-1)(3n-2k-2)}{n(n-1)(n-2)}$	$\Theta(1/n^2)$
(2,1)-lävi	AV	0	0	$\frac{2}{n(n-1)}$	$\frac{k(k-1)}{n(n-1)}$	$\Theta(1/n^2)$
	INT	0	$\frac{2}{n}$	$\frac{2(2n-3)}{n(n-1)}$	$\frac{k(2n-k-1)}{n(n-1)}$	$\Theta(1/n)$

Tabel D.4: Kvantitatiivne ründaja edukus  $(n, t)$ -lävimodelite korral, eeldusel et kasutajad on jaotatud  $\binom{n}{t}$  domeeniks, kus iga domeen võib kasutada konkreetse CA teenust.

Ülevõetud CA-de arv		0	1	2	3
(3,2)-lävi	AV	0	$\frac{2}{3}$	1	1
	INT	0	0	$\frac{1}{3}$	1
(2,1)-lävi	AV	0	$\frac{1}{2}$	1	1
	INT	0	$\frac{1}{2}$	1	1

võtnud kasvõi ühe  $t$ -st CA-st, siis on domeeni käideldavus koheselt rikutud. Kui ründaja saab kontrolli alla  $k$  erinevat CA-d, siis:

- Käideldavus on rikutud neis domeenides, kus oli kasutatud vähemalt üks neist CA-dest. Nagu arutasime eespool (fikseerides  $k$  asemel  $t$ ), on selle sündmuse tõenäosus  $\frac{t}{n}$ , kui  $k = 1$ , ning  $\frac{t(2n-t-1)}{n(n-1)}$ , kui  $k = 2$ .
- Terviklus on rikutud neis domeenides, kus kõik CA-d on riündaja kontrolli all. Tingimusel  $k \geq t$  on selle sündmuse tõenäosus  $\frac{1}{n}$  kui  $k = 1$  ning  $\frac{2}{n(n-1)}$  kui  $k = 2$ .

(3, 2)- ja (2, 1)-lävimodelite korral saame kätte tabelis D.4 toodud väärtsused.

## E Appendix: Logic for describing trust models

### E.1 Syntax

Our language is an instance of mathematical logic. The syntax of a logic consists of *atomic propositions*, and means of combining them into *propositions*. Depending on the nature of atomic propositions and combinators (i.e. on the subject matter of our language), other syntactic categories may need to be introduced.

In our setting, we want to talk about various entities — users, relying parties, certification authorities, etc. They exchange messages with each other, possibly signed with their public keys. We will call both the entities and public keys *agents*, and won't distinguish between them. “Agent” is the only additional syntactic category that we need. Let  $\mathbf{A}$  be the set of all agents of the system.

In our language, the atomic propositions have the form  $\mathcal{C}_{i,j}$ , where  $i, j \in \mathbf{A}$ . Informally, the truth of the proposition  $\mathcal{C}_{i,j}$  means, that agent  $j$  controls agent  $i$ . Controlling means, that if agent  $j$  does something, then it actually was agent  $i$  that made him to do it, and the agent  $i$  should also be thought of as doing that thing. The formal meaning is given by the semantics of the language. We think of  $j$  as a public key and  $i$  as an entity that knows and controls the corresponding private key (which noone else knows).

As usual, the set of propositions of our language is defined inductively. First, all atomic propositions  $\mathcal{C}_{i,j}$  are propositions, and so are the constants `true` and `false`. Second, if  $\phi$  and  $\psi$  are propositions, and  $i, j \in \mathbf{A}$ , then the following are also propositions:

- $\neg\phi, \phi \wedge \psi, \phi \vee \psi, \phi \Rightarrow \psi;$
- $\mathcal{B}_i \phi, \mathcal{I}_{i,j} \phi, \mathcal{X} \phi, \mathcal{Y} \phi, \mathcal{A}_i \phi.$

Here the first bullet point lists the usual connectives of propositional logic — negation, conjunction, disjunction, and implication of propositions. The second bullet point lists the *modalities* that we use to describe the interactions between agents and/or the environment, as well as the passage of time. Intuitively, the modalities mean the following:

- $\mathcal{B}_i \phi$  means that the agent  $i$  *believes* that the proposition  $\phi$  is (currently) true. If an agent believes that  $\phi$  is true, then this does not mean that  $\phi$  is actually true; its truth just seems likely for the agent.
- $\mathcal{I}_{i,j} \phi$  means that the agent  $j$  is currently *telling* agent  $i$  that the proposition  $\phi$  is true. Such telling does not imply that either of them believes that  $\phi$  is true.
- $\mathcal{X} \phi$  means that the proposition  $\phi$  will definitely be true at the *next moment in time*. We define (and make it precise in the semantics) that the time moves in discrete steps. Due to these discrete steps, we also write  $\mathcal{X}^n \phi$  for non-negative integers  $n$ , meaning that  $\phi$  will definitely be true after  $n$  steps. This write-up is merely a syntactic sugar; we define  $\mathcal{X}^0 \phi \equiv \phi$  and  $\mathcal{X}^{n+1} \phi \equiv \mathcal{X} \mathcal{X}^n \phi$ . Note that  $n$  is not a variable, but an integer literal (or: constant).
- $\mathcal{Y} \phi$  means that the proposition  $\phi$  was true at the previous moment in time. We similarly write  $\mathcal{Y}^n \phi$  to denote that  $\phi$  was true  $n$  steps ago.
- $\mathcal{A}_i \phi$  means that the agent  $i$  is able to make it so that  $\phi$  will be true at the next moment in time. It does not have to be a concious effort (or lack of effort) on behalf of that agent;

it just means that if the agent decides that  $\phi$  should be true at the next time moment, then it will be true. In particular, the implication  $\mathcal{X} \phi \Rightarrow \mathcal{A}_i \phi$  will hold for all agents  $i$  and propositions  $\phi$ . We also define the notation  $\mathcal{A}_i^0 \phi \equiv \phi$  and  $\mathcal{A}_i^{n+1} \phi \equiv \mathcal{A}_i \mathcal{A}_i^n \phi$ .

A certification infrastructure is meant to allow agents to find out, which agents are controlled by which ones at which time moments. “Finding out” means forming a belief in it. We see that we have the syntactic components to state that a belief exists, and that agents can work towards establishing it, by the means of one agent telling another one something about control.

## E.2 Expressing trust relationships

Formalizations of trust have been studied for its different intuitive meanings. It may mean one agent trusting another agent to perform certain actions [171,172]. But this is not the viewpoint of trust that we take in this report. In this report, we are interested in one agent trusting another one to not lie about the truth value of certain propositions [173]. Note that trust is only with respect to certain propositions, not all of them.

As Liau [173] discusses, in order to say that an agent  $i$  *trusts* an agent  $j$  with a proposition  $\phi$ , it is sufficient for the following two propositions to be true:

- $\mathcal{B}_i(\mathcal{B}_j \phi \Rightarrow \phi)$  — agent  $i$  believes that agent  $j$  is correct about  $\phi$  being true;
- $\mathcal{B}_i(\mathcal{I}_{i,j} \phi \Rightarrow \mathcal{B}_j \phi)$  — agent  $i$  believes that agent  $j$  will not lie to him about  $\phi$  being true.

Let us introduce syntactic sugar  $\mathcal{T}_{i,j} \phi$ , meaning the conjunction of these two propositions.

A *weak trust* (denoted  $\dashrightarrow$ ) means that only the agent’s statements about his *own* public key are trusted. The ownership of the corresponding private key can be proved cryptographically. This type of trust is needed for a CA  $i$  to issue a certificate to an entity  $j$ . We have  $i \dashrightarrow j$  if we have  $\mathcal{T}_{i,j} \phi$  for all statements  $\phi \equiv \mathcal{Y}^n \mathcal{C}_{k,j}$  for a natural number  $n$  and an agent  $k$ . I.e. if we say that “ $i$  weakly trusts  $j$ ” or that “we require / the system requires  $i$  to weakly trust  $j$ ” then we mean that all such statements  $\mathcal{T}_{i,j} \phi$  must be true.

A *strong trust* (denoted  $\rightarrow$ ) means that the agent’s statements about *other agents’* public keys are trusted. A strong trust is needed for a CA to certify another CA. The relation  $i \rightarrow j$  can also be expressed as  $\mathcal{T}_{i,j} \phi$  for all formulas  $\phi$  of a certain shape; but now the shape of  $\phi$  is different. We obviously want the formulas  $\phi$  to include all propositions of the shape  $\mathcal{Y}^n \mathcal{C}_{k_2,k_1}$  — agent  $j$  is trusted on whether the agent  $k_1$  controls the agent (public key)  $k_2$  —, but this is not the only kind of statement that we want to include in the strong trust. We also want to talk about *trust chains* — agent  $j$  tells agent  $i$ , which other agents can be trusted with respect to the control of public keys. In this case, agent  $i$  has to trust agent  $j$  that the judgement of the latter is a good one. Hence we want  $\phi$  to also range over formulas of the form  $\mathcal{T}_{i,k_3} \mathcal{Y}^n \mathcal{C}_{k_2,k_1}$ .

We also want to have longer trust chains, meaning that agent  $i$  has to trust agent  $j$  in deciding, which other agents can be trusted in deciding which other agents can be trusted, etc., with respect to the control of the public keys. Hence the statement  $\phi$  ranges over all formulas of the form

$$\mathcal{T}_{i,k_l} \mathcal{T}_{i,k_{l-1}} \cdots \mathcal{T}_{i,k_3} \mathcal{Y}^n \mathcal{C}_{k_2,k_1}$$

for all lists of agents  $k_1, \dots, k_l$ .

*Trust with responsiveness* is denoted by a double arrow (e.g.  $\Rightarrow$  denotes the strong trust  $\rightarrow$  with responsiveness). This means that the agent will be available at any time in the future to actually tell whether he believes that  $\mathcal{C}_{pk,j}$  holds or not. This property is important for signatures, and also for authentication if the keys can be revoked. For example, while the

relation  $A \dashrightarrow B$  allows that  $B$  has presented a different public key to the agent  $C$  and is now silent about it, the relation  $A \Rightarrow B$  assumes that  $B$  honestly tells to  $A$  whether the key that he showed to  $C$  is indeed his. Responsiveness of a CA may be supported technically by some kind of OCSP (Online Certificate Status Protocol) service, which we will not model explicitly. When a CA certifies another CA, it should also believe in the other CA-s responsiveness.

Responsiveness is achieved by adding to the statement  $\mathcal{T}_{i,j} \phi$  the belief that the last responsible agent will actually tell agent  $i$  whether the control of one agent over another exists or not. This responsiveness of the agent is judged by the next agent in the trust chain, and the goodness of the judgement is judged by all preceding agents. Hence weak trust with responsiveness —  $i \Rightarrow j$  — means

$$\mathcal{T}_{i,j} \mathcal{Y}^n \mathcal{C}_{k,j} \wedge \mathcal{B}_i (\mathcal{A}_i \mathcal{I}_{i,j} \mathcal{Y}^{n+1} \mathcal{C}_{k,j} \vee \mathcal{A}_i \mathcal{I}_{i,j} \mathcal{Y}^{n+1} \neg \mathcal{C}_{k,j})$$

for all  $n \in \mathbb{N}$  and  $k \in A$ .

Strong trust with responsiveness is more laborious to specify. We first define the formulas  $\psi[k_1, k_2, \dots, k_l; \phi]$  representing responsiveness; they state that agent  $i$  believes that he is capable from obtaining the agents  $k_l, \dots, k_1$  a chain of claims that agent  $i$  can use as the basis of belief that the proposition  $\phi$  holds. Alternatively, agent  $i$  is able to obtain from (a subset of) these agents a chain of claims implying that support for  $\phi$  is not forthcoming. Combining  $\psi[k_1, \dots, k_l; \phi]$  with (strong) trust towards  $\phi$  via the trust chain  $j, k_l, \dots, k_1$  will then give us  $i \Rightarrow j$ . We have

$$\begin{aligned} \psi[k_1; \phi] &\equiv \mathcal{A}_i \mathcal{I}_{i,j} \mathcal{Y} \phi \vee \mathcal{A}_i \mathcal{I}_{i,j} \neg \mathcal{Y} \phi \\ \psi[k_1, \dots, k_l; \phi] &\equiv \mathcal{A}_i \mathcal{I}_{i,k_l} \neg \mathcal{T}_{i,k_{l-1}} \cdots \mathcal{T}_{i,k_1} \phi \vee (\mathcal{A}_i \mathcal{I}_{i,k_l} \mathcal{T}_{i,k_{l-1}} \cdots \mathcal{T}_{i,k_1} \phi \wedge \psi[k_1, \dots, k_{l-1}; \phi]) . \end{aligned}$$

We now define strong trust with responsiveness —  $i \Rightarrow j$  — as the holding of the formulas

$$\mathcal{T}_{i,j} \mathcal{T}_{i,k_l} \mathcal{T}_{i,k_{l-1}} \cdots \mathcal{T}_{i,k_3} \mathcal{Y}^n \mathcal{C}_{k_2, k_1} \wedge \mathcal{B}_i \psi[k_3, \dots, k_l, j; \mathcal{Y}^n \mathcal{C}_{k_2, k_1}]$$

for all  $n \in \mathbb{N}$  and agents  $k_1, \dots, k_l$ .

### E.3 Axioms and inference rules

In order to show the truth of a proposition, it must be *derivable* from the *axioms* of the logic by means of the *inference rules*. We write  $\vdash \phi$  to denote that  $\phi$  can be derived like that.

As next, let us discuss the axioms and inference rules of our language. First, as usual, we include as axioms all tautologies of propositional calculus, i.e. all propositions in our language that would be true, no matter what the truth values are for atomic propositions, and for the applications of modalities. We also include the inference rule *modus ponens*: for all propositions  $\phi$  and  $\psi$ , if  $\vdash \phi$  and  $\vdash (\phi \Rightarrow \psi)$ , then  $\vdash \psi$ .

Let us now discuss axioms related to individual modalities. There exist a number of “standard” axioms in modal logic, capturing the essence of different kinds of modalities (e.g. knowing, believing, permissions and obligations, computability and derivability, etc.). Let us list some of them, where  $\mathcal{M}$  denotes an arbitrary modality.

(K)  $\mathcal{M}\phi \wedge \mathcal{M}(\phi \Rightarrow \psi) \Rightarrow \mathcal{M}\psi$ . This axiom states that it is possible to do logical arguments under  $\mathcal{M}$ . We see that it is very similar to *modus ponens*.

(4)  $\mathcal{M}\phi \Rightarrow \mathcal{M}\mathcal{M}\phi$ . This axiom captures some sort of reflection in the positive sense.

(5)  $\neg \mathcal{M}\phi \Rightarrow \mathcal{M}\neg \mathcal{M}\phi$ . This axiom also captures reflection, but in the negative sense.

- (D)  $\mathcal{M}\phi \Rightarrow \neg\mathcal{M}\neg\phi$ . This axiom states some kind of consistency under  $\mathcal{M}$ : it is not the case that  $\mathcal{M}$  holds for both  $\phi$  and  $\neg\phi$ . An equivalent way (if (K) is also assumed) of stating (D) is  $\neg\mathcal{M}\text{false}$ .

There is also a generic derivation rule called *necessitation*: from  $\vdash\phi$  we may infer  $\vdash\mathcal{M}\phi$ . I.e. tautologies may be put under  $\mathcal{M}$ .

Individually, the modalities we've introduced have to satisfy the following:

- $\mathcal{B}_i$  satisfies axioms KD45 and the necessitation rule. This is the standard way of axiomatizing beliefs.
- $\mathcal{I}_{i,j}$  satisfies axioms KD and the necessitation rule. I.e. if agent  $j$  tells agent  $i$  something, then agent  $i$  will also learn everything that logically follows from it. This effectively means that agent  $i$  is able to do the logical derivations. Hence also the (D) axiom — agent  $i$  will notice when he is told logically inconsistent things, and will not accept them.
- $\mathcal{X}$  and  $\mathcal{Y}$  satisfy KD and the necessitation rule, meaning that logic works in both future and in the past, and there are no inconsistencies.
- $\mathcal{A}_i$  does not satisfy K, because agent  $i$  may be able to cause some proposition  $\phi$  to hold at the next time moment, but, behaving differently, he may alternatively be able to cause  $\neg\phi$  to hold. However,  $\mathcal{A}_i$  satisfies a weaker derivation rule, which we phase as an inference rule:

$$\frac{\vdash\phi \Rightarrow \psi}{\vdash\mathcal{A}_i\phi \Rightarrow \mathcal{A}_i\psi} \text{ (AImp)}$$

i.e. if agent  $i$  can cause  $\phi$ , then he also causes everything that logically follows from it. Also, the necessitation rule is applicable to  $\mathcal{A}_i$  — tautologies will happen.

A number of axioms relate different modalities and atomic propositions. We have already discussed the intuitive meaning of  $\mathcal{C}_{i,j}$ ; we will now make it formal through an axiom:

$$\mathcal{I}_{k,i}\phi \wedge \mathcal{C}_{i,j} \Rightarrow \mathcal{I}_{k,j}\phi . \quad (\text{CI})$$

In this axiom, agent  $i$  is a public key. The axiom states that if agent  $i$  says something, but it was controlled by agent  $j$ , then it was actually agent  $j$  that said this thing. Besides the modality of agent  $i$  telling something to some other agent, we also have other modalities that have something to do with that agent. It may be difficult to state, what a public key believes, or what it is able to cause. Still, it may make sense to state that anything a controlled agent believes or causes, actually stems from the controlling agent. Thus we also introduce the following axioms:

$$\mathcal{B}_i\phi \wedge \mathcal{C}_{i,j} \Rightarrow \mathcal{B}_j\phi , \quad (\text{CB})$$

$$\mathcal{A}_i\phi \wedge \mathcal{C}_{i,j} \Rightarrow \mathcal{A}_j\phi . \quad (\text{CA})$$

Obviously, the future and the past are tightly bound together — the past of the future is now, and vice versa. This is expressed by the following axioms:

$$\phi \Rightarrow \mathcal{X}\mathcal{Y}\phi , \quad (\text{TF})$$

$$\phi \Rightarrow \mathcal{Y}\neg\mathcal{X}\neg\phi . \quad (\text{TP})$$

These axioms express that while the future is not predetermined, the past is. The axiom TF states that if  $\phi$  holds now, then in all possible futures (making a single step ahead),  $\phi$  definitely

held in the previous step. In the other direction (axiom TP), if we make one step to the past, then there is some future where  $\phi$  holds, i.e. it is not the case that  $\neg\phi$  holds in all futures.

We think of  $\mathcal{A}_i$  as the agent  $i$  restricting the set of possible futures where we can end up. We express this by the axiom

$$\mathcal{X} \phi \Rightarrow \mathcal{A}_i \phi, \quad (\text{FA})$$

i.e. if  $\phi$  holds in all possible futures, then an agent is able to (vacuously) cause  $\phi$ .

## E.4 Semantics

We give our logic a typical possible-worlds semantics, where the meaning of each modality is given by an accessibility relation. A *frame* is a tuple  $\mathcal{F} = (\mathcal{W}, V, \mathbf{B}, \mathbf{I}, \mathbf{X}, \mathbf{A})$ , where  $\mathcal{W}$  is a set of (*possible*) *worlds*, and other components are relations of certain type between the worlds and the agents. Namely,

- $V \subseteq \mathcal{W} \times \mathbf{A} \times \mathbf{A}$ ,
- $\mathbf{B} \subseteq \mathbf{A} \times \mathcal{W} \times \mathcal{W}$ ,
- $\mathbf{I} \subseteq \mathbf{A} \times \mathbf{A} \times \mathcal{W} \times \mathcal{W}$ ,
- $\mathbf{X} \subseteq \mathcal{W} \times \mathcal{W}$ ,
- $\mathbf{A} \subseteq \mathbf{A} \times \mathcal{W} \times 2^{\mathcal{W}}$ , where  $2^{\mathcal{W}}$  denotes the *power set* (i.e. the set of all subsets) of  $\mathcal{W}$ .

These relations define the truth values of all propositions in any world  $\mathbf{w} \in \mathcal{W}$ . In order for the axioms and inference rules to hold, these relations cannot be arbitrary, though. Let us first describe how the truth value of a proposition is determined, and discuss the requirements on the relations afterwards.

The write-up  $\mathcal{F}, \mathbf{w} \models \phi$  means “in frame  $\mathcal{F}$  and in a world  $\mathbf{w}$  among the possible worlds of  $\mathcal{F}$ , the proposition  $\phi$  holds”. We also write  $\mathcal{F} \models \phi$ , if  $\mathcal{F}, \mathbf{w} \models \phi$  for all worlds  $\mathbf{w}$  of  $\mathcal{F}$ . Moreover, we write  $\models \phi$ , if  $\mathcal{F} \models \phi$  holds for all *well-formed* frames  $\mathcal{F}$ , i.e. those frames where the relations satisfy the conditions we describe below. We have

- $\mathcal{F}, \mathbf{w} \models C_{i,j}$ , iff  $(\mathbf{w}, i, j) \in V$ ;
- $\mathcal{F}, \mathbf{w} \models \text{true}$ ;
- $\mathcal{F}, \mathbf{w} \models \neg\phi$ , iff  $\mathcal{F}, \mathbf{w} \models \phi$  cannot be derived according to these (syntax-directed) rules that we are currently stating;
- $\mathcal{F}, \mathbf{w} \models \phi \wedge \psi$ , iff  $\mathcal{F}, \mathbf{w} \models \phi$  and  $\mathcal{F}, \mathbf{w} \models \psi$ ;
- $\mathcal{F}, \mathbf{w} \models \phi \vee \psi$ , iff either  $\mathcal{F}, \mathbf{w} \models \phi$  or  $\mathcal{F}, \mathbf{w} \models \psi$  (or both);
- $\mathcal{F}, \mathbf{w} \models \phi \Rightarrow \psi$ , iff it is not the case that  $\mathcal{F}, \mathbf{w} \models \phi$ , but  $\mathcal{F}, \mathbf{w} \models \psi$  cannot be derived;
- $\mathcal{F}, \mathbf{w} \models B_i \phi$  iff  $\mathcal{F}, \mathbf{w}' \models \phi$  holds for all  $\mathbf{w}' \in \mathcal{W}$ , where  $(i, \mathbf{w}, \mathbf{w}') \in \mathbf{B}$ ;
- $\mathcal{F}, \mathbf{w} \models I_{i,j} \phi$  iff  $\mathcal{F}, \mathbf{w}' \models \phi$  holds for all  $\mathbf{w}' \in \mathcal{W}$ , where  $(i, j, \mathbf{w}, \mathbf{w}') \in \mathbf{I}$ ;
- $\mathcal{F}, \mathbf{w} \models X \phi$  iff  $\mathcal{F}, \mathbf{w}' \models \phi$  holds for all  $\mathbf{w}' \in \mathcal{W}$ , where  $(\mathbf{w}, \mathbf{w}') \in \mathbf{X}$ ;
- $\mathcal{F}, \mathbf{w} \models Y \phi$  iff  $\mathcal{F}, \mathbf{w}' \models \phi$  holds for all  $\mathbf{w}' \in \mathcal{W}$ , where  $(\mathbf{w}', \mathbf{w}) \in \mathbf{X}$  (as we state below, there is going to be exactly one such  $\mathbf{w}'$ );

- $\mathcal{F}, \mathbf{w} \models \mathcal{A}_i \phi$ , if  $(i, \mathbf{w}, \mathbf{W}) \in \mathbf{A}$  for the set  $\mathbf{W} = \{\mathbf{w} \in \mathcal{W} \mid \mathcal{F}, \mathbf{w} \models \phi\}$ .

For the axioms to hold, the relations  $\mathbf{B}$ ,  $\mathbf{I}$ ,  $\mathbf{X}$  and  $\mathbf{A}$ , as well as the valuation  $V$  cannot be completely arbitrary. We need the following *well-formedness* conditions for each single relation:

- For each  $i \in A$ , the relation  $\mathbf{B}(i, \cdot, \cdot)$  needs to be such, that the the axioms KD45 are satisfied for the modality  $\mathcal{B}_i$ . It is well-known, how such relation looks like. The relation  $\mathbf{B}(i, \cdot, \cdot) \subseteq \mathcal{W} \times \mathcal{W}$  is such, that there exists a partitioning  $\mathcal{W}_1 \dot{\cup} \mathcal{W}_2 \dot{\cup} \dots = \mathcal{W}$ , and a non-empty subset  $\mathcal{V}_k \subseteq \mathcal{W}_k$  for each of the parts, such that  $\mathbf{B}(i, \mathbf{w}, \mathbf{w}')$  iff there exists some  $k$ , such that  $\mathbf{w} \in \mathcal{W}_k$  and  $\mathbf{w}' \in \mathcal{V}_k$ . Intuitively, agent  $i$  cannot distinguish between different worlds in the same part. For the part  $\mathcal{W}_k$ , he considers the worlds in  $\mathcal{W}_k \setminus \mathcal{V}_k$  to be implausible.
- For each  $i, j \in A$ , the relation  $\mathbf{I}(i, j, \cdot, \cdot)$  needs to be such, that the axioms KD are satisfied for the modality  $\mathcal{I}_{i,j}$ . For their satisfiability, it is necessary and sufficient to require that for each  $\mathbf{w} \in \mathcal{W}$  there exists some  $\mathbf{w}' \in \mathcal{W}$ , such that  $\mathbf{I}(i, j, \mathbf{w}, \mathbf{w}')$  holds. Intuitively, when agent  $j$  tells something to agent  $i$ , then he is narrowing down the set of possible worlds that he is claiming the agents to be in. Agent  $j$  is unable to tell something contradictory to agent  $i$ , i.e. he cannot narrow the set of possible worlds down to the empty set.
- Similarly, for the axioms KD to hold for  $\mathcal{X}$  and  $\mathcal{Y}$ , we require that for each  $\mathbf{w} \in \mathcal{W}$  there exists  $\mathbf{w}' \in \mathcal{W}$ , such that  $\mathbf{X}(\mathbf{w}, \mathbf{w}')$  holds. We also require the opposite: for each  $\mathbf{w} \in \mathcal{W}$  there exists  $\mathbf{w}' \in \mathcal{W}$ , such that  $\mathbf{X}(\mathbf{w}', \mathbf{w})$  holds. Moreover, in this case, the world  $\mathbf{w}'$  must be unique. In this way, the axioms TP and TF will be satisfied.
- For each  $i \in A$ , the relation  $\mathbf{A}(i, \cdot, \cdot)$  must be such, that the inference rule AImp is valid for  $\mathcal{A}_i$ . This weakening operation in AImp suggests some kind of upwards closure requirement for  $\mathbf{A}(i, \cdot, \cdot)$ . Namely: if  $\mathbf{w} \in \mathcal{W}$ ,  $\mathbf{W}, \mathbf{W}' \subseteq \mathcal{W}$ ,  $\mathbf{W} \subseteq \mathbf{W}'$ , and  $\mathbf{A}(i, \mathbf{w}, \mathbf{W})$  holds, then also  $\mathbf{A}(i, \mathbf{w}, \mathbf{W}')$  has to hold. Effectively, each  $\mathbf{A}(i, \mathbf{w}, \cdot)$  is determined by the minimal sets  $\mathbf{W}$ , such that  $\mathbf{A}(i, \mathbf{w}, \mathbf{W})$  holds.

The different relations also have to be related to each other in certain ways, such that the axioms involving several modalities and/or atomic propositions are satisfied. Namely:

- Suppose that  $\mathcal{F}, \mathbf{w} \models \mathcal{C}_{i,j}$ , i.e.  $(\mathbf{w}, i, j) \in V$ . In this case, for each  $k \in A$  and  $\mathbf{w}' \in \mathcal{W}$ , the relationship  $\mathbf{I}(k, j, \mathbf{w}, \mathbf{w}')$  must imply that  $\mathbf{I}(k, i, \mathbf{w}, \mathbf{w}')$  holds. Similarly,  $\mathbf{B}(j, \mathbf{w}, \mathbf{w}')$  implies  $\mathbf{B}(i, \mathbf{w}, \mathbf{w}')$  and  $\mathbf{A}(i, \mathbf{w}, \mathbf{W})$  implies  $\mathbf{A}(j, \mathbf{w}, \mathbf{W})$  for any  $\mathbf{W} \subseteq \mathcal{W}$  (notice the opposite locations of  $i$  and  $j$  in the condition involving  $\mathbf{A}$ , compared to other named relations).
- For any  $i \in A$ ,  $\mathbf{w} \in \mathcal{W}$ , and  $\mathbf{W} \subseteq \mathcal{W}$ : if  $\mathbf{X}(\mathbf{w}) \subseteq \mathcal{W}$  denotes the set of all such  $\mathbf{w}' \in \mathcal{W}$ , such that  $\mathbf{X}(\mathbf{w}, \mathbf{w}')$  holds, then  $\mathbf{A}(i, \mathbf{w}, \mathbf{W})$  must imply  $\mathbf{A}(i, \mathbf{w}, \mathbf{W} \cap \mathbf{X}(\mathbf{w}))$ . I.e. the minimal sets defining  $\mathbf{A}(i, \mathbf{w}, \cdot)$  must be subsets of  $\mathbf{X}(\mathbf{w})$ .

If the frame  $\mathcal{F}$  satisfies all these conditions, then we say that it is well-formed. We can now state the main theorem of this Appendix:

**Theorem (soundness).** If  $\vdash \phi$ , then  $\models \phi$ .

The proof of the theorem proceeds with induction over the derivation tree of  $\vdash \phi$ . We show that all axioms, and all inference rules are sound.

It would also be interesting to state the opposite theorem: “if  $\models \phi$  then  $\vdash \phi$ ” (**completeness**). However, we do not know whether our axioms and inference rules match with our well-formedness conditions sufficiently well for this theorem to be valid.