CYBERNETICA

Krüptograafiliste algoritmide elutsükli uuring

Uuringuaruanne

Versioon 4.0

3. juuni 2015. a.

65 lk

Dok. A-101-1

Sisukord

1 Sissejuhatus

See aruanne on loogiline jätk 2011. ja 2013. aastal koostatud krüptograafiliste algoritmide elutsükli uuringutele [14, 17]. Kui kaks esimest aruannet seadsid eesmärgiks anda võimalikult täielik ülevaade kogu praktikas kasutatavast krüptograafiast, siis seekord olid fookuses konkreetsemad küsimused. Krüptograafia on juba piisavalt arenenud ja stabiilne valdkond, kus suuri matemaatilisi läbimurdeid esineb harva. Küll aga leitakse pidevalt vigu teostustes, samuti tulevad järjest peale uued lahendused (elliptkõverad, laiemate võimalustega elektroonilise identiteedi protokollid, postkvantkrüptograafia jne). See nõuab teemaga pidevat kursisolemist ning teadmuse kogumist, et osata infosüsteemide kaasajastamisel ja arenduste plaanimisel uute ohtude ja võimalustega arvestada.

Seekordne ülevaade keskendub valitud kitsamatele valdkondadele.

- Jaotises 2 täiendatakse 2013. aasta uuringut vahepeal aset leidnud sündmuste valguses. Olulisi matemaatilisi ja krüptograafilisi läbimurdeid pole selle aastaga toimunud, põhilisi probleeme tekitasid SSL/TLS protokollistiku vead. Üle on vaadatud ka pangalinkide olukord, ENISA võtmepikkuste soovitused ja TrueCrypti tootetoe lõppemisega seotud temaatika.
- Jaotise 3 moodustab elliptkõverate krüptograafia ülevaade: sissejuhatus elliptkõveratesse ning neile kõveratele tuginevasse krüptograafiasse, elliptkõverate turvaomadused, nende toetus standardites ning tark- ja riistvarasüsteemides.
- Jaotis 4 hindab räsifunktsioon SHA-1 murdumisega seotud riske, eriti avaliku võtme sertifikaatide ja sertifitseerimisteenuse võltsimise seisukohast. Jaotises antakse ülevaade räsifunktsioonide murdmise teooria ja praktika hetkeseisust ning hinnatakse võltsitud sertifitseerimisteenuse loomiseks vajalikku rahalist investeeringut.
- Jaotis 5 käsitleb krüptograafilisi protokolle, mis võivad leida rakendust praeguste elektroonilise identiteedi lahenduste (ID-kaart, mobiil-iD) laiendamisel, näiteks lähiväljaside võimalustega.
- Jaotises 6 töötatakse välja kuluanalüüsi metoodika krüptograafiliste algoritmide muutmisega kaasnevate kulude hindamiseks.
- Jaotis 7 annab ülevaate krüptograafilistest algoritmidest, mis jäävad turvaliseks ka pärast kvantarvutite teostamist.
- Jaotis 8 teeb aruandest kokkuvõtteid ning annab soovitusi krüptograafiliste algoritmide kasutamiseks.

Aruandes esinevat erialaterminoloogiat ning sagelikasutatavaid lühendeid katab Andmekaitse ja infoturbe seletussõnastik[1].

[1] http://akit.cyber.ee/

2 Riigi põhitaristus kasutatavate krüptolahenduste turvalisus

Järgnevas jaotises anname ülevaate pärast aruande [17] ilmumist toimunud arengutest riigi põhitaristus olulist rolli omavate krüptoprotokollide ja nende teostuste teadaolevas turvalisuses. Ilmselt ületas 2014. aastal leitud teostusvigade tõsidus tunduvalt protokollispetsifikatsioonides leitud puudusi.

2.1 SSL/TLS

Transport Layer Security (TLS) [55] ja tema eelkäija *Secure Sockets Layer (SSL)* on levinuim protokollistik kahe osapoole vaheliseks autentimiseks ja võtmevahetuseks ning sellele järgnevaks turvaliseks suhtluseks. Aruande [17] jaotistes 3.2.1 ja 3.5 andsime ülevaate selle protokolli ja tema teostuste teadaolevast turvalisusest 2013. aasta seisuga. Järgnevalt anname ülevaate 2014. aastal selgunud asjaoludest.

Protokollivead

Kolmikkätlusrünne (*Triple Handshake Attack*)

TLS-i võtmevahetusprotokolli põhiline sõnumivoog lubab *kliendil* ja *serveril* veenduda teineteise identiteedis (kasutades selleks teineteise sertifikaate ja/või teisi autentimismehhanisme) ning kokku leppida sümmeetriline võti, mida ainult nemad kahekesi teavad ja edaspidise suhtluse turvamiseks kasutavad. Lisaks põhivoole on TLS-il olemas mitmeid teisi sõnumivoogusid, mis osapoolte vahelist suhtlemist hõlbustada võivad. Näiteks on katkenud TLS-seansi jätkamiseks olemas eraldi alamprotokoll, mis nõuab osapooltelt vähem arvutusi kui terve võtmevahetusprotokolli uuesti jooksutamine. Samuti on võimalik juba pikalt kasutusel olnud seansivõti välja vahetada, jooksutades selleks taas võtmevahetusprotokolli.

Nende alamprotokollide detailid ja võimalikud variandid mõjutavad üksteist nii, et ründaja saab viia protokolli osapooli olekutesse, mida võib pidada edukaks ründeks [43]. Ründestsenaariumis võtab klient kõigepealt ühendust ründaja kontrolli all oleva serveriga, mis seejärel ühendub ausa serveriga. Kummaski seansis jääb klient esialgu identifitseerimata. Ründaja hoolitseb selle eest, et nende seansside teatud atribuudid oleksid võrdsed, nii et neid võiks katkenud TLS-seanssi jätkates segi ajada. Ründaja põhjustabki seansside katkemise ning nende jätkamise järgselt kasutatava seansivõtme vahetuse ühes kliendi identifitseerimisega. Kuna kaks seanssi on segiaetavad, siis arvab aus server lõpuks, et ta suhtleb ausa kliendiga. Aus klient arvab aga, et ta suhtleb endiselt ründaja kontrolli all oleva serveriga (kuigi ta loomulikult ei tea, et ründaja seda serverit kontrollib). Viimasel seansivõtmevahetusel saab aus klient uue serverisertifikaadi, kuid selline sertifikaadimuu-

tus on võtmevahetusprotokollis lubatud. Bhargavan jt. [43] näitavad, et selline rünne on veebibrauserite vastu reaalselt kasutatav.

Kirjeldatud ründe tõkestamiseks tuleb parandada seda, kuidas katkenud TLS-seansse identifitseeritakse [42], või keelata serveri sertifikaadi muutus võtmevahetusprotokolli ajal. Sertifikaadimuutuse mitteaktsepteerimine on klientprogrammi ülesanne ning on 2014. aasta kevade jooksul realiseeritud kõigis peamistes veebibrauserites (Firefox, Chromium, Internet Explorer, Safari) [43, jaotis IX].

POODLE

Eelmises aruandes [17, jaotis 3.5.1] kirjeldasime ründeid SSL/TLS transpordiprotokolli vastu, mida on võimaldanud asjaolu, et protokollispetsifikatsioon ei järgi teoreetilisi soovitusi, kuidas plokkšifri abil luua turvalist kanalit. 2014. aastal on leitud veel üks selline rünne. POODLE-rünne [85] lubab ründajal dekrüpteerida saladusi (näiteks seansiküpsiseid), mida turvalise kanali kaudu saadetakse. Sarnaselt BEAST-ründele [59] nõuab POODLE, et ründaja suudaks mõjutada, kui pikad on rünnatava veebibrauseri poolt väljasaadetavad päringud ja nende osad. Erinevalt BEAST-ist ei vaja aga POODLE-rünne, et ründajal oleks kontroll veebilehitseja ja serveri vahelise võrguliikluse üle. Erinevad on ka protokollinõrkused, mida need kaks rünnet ära kasutavad. Kui BEAST-il oli selleks IV-aheldatud režiimi CBC nõrkus [33], siis POODLE kasutab ära seda, et avateksti täidise (*padding*) sisu on SSL versiooni 3 transpordiprotokollis [65] spetsifitseerimata (v.a. selle viimane bait, mis peab võrduma täidise pikkusega) ning täidis ise on autentimata. Seega on POODLE-rünne kasutatav ainult siis, kui veebibrauserist klient ja server on omavahelise võrguliikluse turvamiseks millegipärast valinud vananenud protokolli SSLv3. See võib aga juhtuda, kui esialgsel võtmevahetusel takistab ründaja neil uuema protokolliversiooni kokkuleppimist. Seega annab POODLE-rünne veel ühe põhjuse SSLv3 protokolli toetamisest loobumiseks.

TLS-protokolli kõigis versioonides on täidise sisu täielikult spetsifitseeritud ning POODLE-rünne ei tohiks nende protokollide vastu rakendatav olla. Hiljuti on aga selgunud, et mõned TLS-protokolli teostused ei kontrolli transpordiprotokolli kaudu saabunud teadete täidise vastavust TLS-spetsifikatsioonile [75]. Seetõttu on ka need teostused POODLE-ründe suhtes haavatavad. Nn. *Postel'i seadus* — „ole . . . liberaalne selles, mida sa teistelt vastu võtad“ — krüptograafiliste protokollide teostamisel ei kehti.

Vead teostustes ja kasutuses

Heartbleed

Heartbleed [25] on TLS-i südamelöögilaienduse [96] teostusviga OpenSSL-teegi versioonides 1.0.1 kuni 1.0.1f. OpenSSL-teegi teistes harudes (1.0.0 ja 0.9.8) seda viga ei esine [25]. Viga avalikustati 7. aprillil 2014 ja parandati samal päeval välja lastud OpenSSL-teegi versioonis 1.0.1g.

TLS südamelöögilaiendus on mõeldud TLS-kanali pikemaajaliseks elushoidmiseks ajal, kui andmeid saata ei tule. Kanali elushoidmisest huvitatud osapool saadab teisele osapoolele teate ning ootab sama teadet teiselt osapoolelt tagasi. Teate vorming sisaldab lisaks kordamist ootavale baidijadale eraldi ka teate pikkust. Pahatahtlik osapool võib konstrueerida teate, mille väidetav pikkus on suurem kordamist ootavast baidijadast. OpenSSL-teegi vigastes versioonides ei kontrollitud, kas väidetav pikkus ei ole liiga suur. Selle asemel saadeti tagasi sõnum, mille pikkus oli võrdne vastuvõetud teate väidetava pikkusega. Selline

vastussõnum sisaldas lisaks korratavale baidijadale ka sellele järgneva mälupiirkonna sisu. Järgnevas mälupiirkonnas võis olla ka väärtusi (näiteks privaatvõtmete osi), mida ei oleks tohtinud teisele osapoolele saata. Lihtsasti jälgitav seletus Heartbleed-veale on toodud näiteks allikas [86].

ChangeCipherSpec injection

ChangeCipherSpec on teade, mille TLS-protokolli osapooled vahetavad pärast õnnestunud võtmevahetusprotokolli, andes sellega teineteisele teada, et nad võtavad nüüd värskelt kokkulepitud võtme (ning šifri ja selle töörežiimi) kasutusse. TLS-i spetsifikatsioon [55] näeb ette, et *ChangeCipherSpec*-teade võib järgneda ainult edukale võtmevahetusele [74]. OpenSSL-teegi versioonid, mis on varasemad kui 1.0.1h, 1.0.0m või 0.9.8za, on aga valmis *ChangeCipherSpec*-teateid töötlema ka varem, näiteks ajal, kui šiffer ja kasutusrežiim on kokku lepitud, aga võtme genereerimiseks kasutatav salajane juhuslik väärtus veel mitte [70]. Nii võib juhtuda, et võti genereeritakse väärtusest, mida ründaja suudab ära arvata. Ründaja, kes suudab kontrollida võrguliiklust TLS-võtmevahetust alustavate osapoolte vahel, võib neile mõlemale saata *ChangeCipherSpec*-teate enne, kui osapoolte vaheline kanal turvatud on. Sel juhul hakkavad osapooled kasutama ründajale teada olevat võtit. Viga on parandatud OpenSSL-teegi versioonides 1.0.1h, 1.0.0m ja 0.9.8za, mis lasti välja 5. juunil 2014, samal ajal kui käesoleva teostusvea kirjelduski.

Winshock

Winshock [109] on klassikaline puhvri ületäitumisviga Microsofti veebibrauserite ja -serverite kasutatavas *Schannel*-teegis, kus puhvri loomisel leitakse tema pikkus ühel viisil, temasse sisu kopeerimisel kopeeritava sisu pikkus teisel viisil ning ei kontrollita, kas teine neist väärtustest pole suurem kui esimene [51]. Viga võis avalduda teise osapoole sertifikaadis leiduva signatuuri dekodeerimisel transpordivormingust, kui sertifikaat oli pahatahtlikult vormindatud ja signatuur oli moodustatud mõne elliptkõvera signatuuriskeemiga. Vea tõsidust võimendas asjaolu, et *Schannel*-teek ei järginud täpselt TLS-i spetsifikatsiooni, mis näeb ette, et kui TLS-i klient saadab serverile oma sertifikaadi, ilma et viimane oleks seda küsinud, tuleb ühendus sulgeda. Selle asemel võttis *Schannel*-teeki kasutav server küsimata saadud sertifikaadi vastu ja asus seda töötlema [19].

Viga avalikustati 11. novembril 2014 ja parandati samal päeval Microsofti väljalastud turvapaigaga [26].

Goto fail

Apple'i TLS-protokolli teostavas teegis *Secure Transport*, mis oli kaasas iOS-i ja OS X-i teatud versioonidega, jäeti kontrollimata TLS-võtmevahetusprotokolli käigus serverilt saadud signatuur kokkulepitud seansivõtmel, kui TLS-i versioon oli väiksem kui 1.2 ja kasutusel oli Diffie-Hellmani võtmevahetus [73]. Viga esines iOS-i versioonides 6.x, mis on vanemad kui 6.1.6, versioonides 7.x, mis on vanemad kui 7.0.6, ja OS X-i versioonides 10.9.x, mis on vanemad kui 10.9.2 [27].

OpenSSL-i versioonialandusrünne

OpenSSL-teegi versioonides, mis on vanemad kui 1.0.1i, valib server kasutatavaks protokolliks TLS versiooni 1.0, juhul kui esimene teade kliendilt (*ClientHello*) on väga killustu-

nud, seda isegi juhul, kui klient ja server mõlemad toetavad ka TLS-i uuemaid versioone. Sel viisil on kliendi ja serveri vahelist võrguliiklust kontrollival ründajal võimalik vältida uuemate TLS-versioonide kasutamist. Alates teegi versioonist 1.0.1i vana protokolliversiooni enam kasutusele ei võeta, vaid fragmenteerunud ClientHello-teate saamisel suletakse ühendus [23].

Puudulik sertifikaatide kontroll

2014. aastal on selgunud, et mitmete populaarsete teekide ja rakenduste vanemad versioonid ei kontrolli TLS-i võtmevahetusprotokollis piisavalt põhjalikult, kas teise osapoole (tüüpiliselt serveri) poolt esitatud sertifikaat seda osapoolt piisavalt hästi identifitseerib. See on võimaldanud kliendi võrguühendust kontrollival ründajal petta see klient ühenduma ründaja kontrolli all oleva serveriga.

Apple'i iCloud Data Access, mis oli kaasas iOS-i versioonidega, mis on varasemad kui 8.1, lubas veebisaitidel ennast tuvastada endasigneeritud (*self-signed*) sertifikaatidega [18]. Vabavaraliste kiirsõnumirakenduste (Pidgin jt.) kasutatava *libpurple*-teegi versioonid, mis on varasemad kui 2.10.10, ei kontrollinud, kas kõigi vahepealsete sertifikaatidega sertifikaadiahelas tohib uusi sertifikaate välja anda [20]. Sarnaste kontrollidega oli probleeme ka GnuTLS-teegi versioonides, mis on vanemad kui 3.2.12 või 3.1.22 [81]. Mitmeid andmevahetusprotokolle teostava *libcurl*-teegi versioonid 7.27.0 kuni 7.35.0 (teatud konfiguratsioonides) jätsid HTTPS-protokollis serveri numbrilise IP-aadressi kasutamise korral kontrollimata, kas serveri nimi vastab sertifikaadis olevale [22, 21].

2.2 Pangalink

Pangalink (iPizza) on 1990ndate aastate lõpul Eesti pankade poolt kasutusele võetud firmapärane standard, mis võimaldab teenuse andjail autentida kasutajaid internetipanga kaudu. Samuti võimaldab see kasutajatel internetipangas kaupmeestele maksta. Pangalingist ilmus 2014. aasta oktoobris uus versioon ja Pangaliitu ühinenud pangad toetavad seda. Kaupmeestele muutub pangalingi uue versiooni kasutamine kohustuslikuks 2015. aasta lõpus [24]. Oluliseks eesmärgiks pangalingi uue versiooni väljatöötamisel oli protokolli turvalisuse tõstmine – protokolli eelmise versiooni spetsifikatsiooni ja kasutusel olevate implementatsioonide vastu oli teada mitmeid ründeid [80, 88, 17].

Siinse aruande koostajad ei ole teadlikud põhjalikest turvaanalüüsidest pangalingi uuele versioonile. Spetsifikatsiooni põhjal võib aga oletada, et eelmise versiooni jaoks teada oleva vahendusründe suhtes on uuem versioon turvaline, sest panga poolt allkirjastatud kinnituses kasutaja identiteedi kohta on nüüd ühe väljana olemas ka allkirja kontrollija – kasutaja identiteedist huvitatud asutuse – tunnus.

2.3 TUPAS

TUPAS on Soome finantsettevõtete liidu *Finanssialan Keskusliitto* loodud protokoll [62], mille abil saavad teenuseandjad autentida kasutajaid internetipanga kaudu. Protokolli kasutab Eestis Nordea pank. Protokollist TUPAS on 2013. aasta lõpus ilmunud uus versioon, mis erineb eelmisest [61] küll ainult panku identifitseerivate koodide nimekirja poolest.

Aruandes [17] heitsime me ette Eesti pankade teenustes jätkuvat räsifunktsioonide MD5 ja SHA-1 kasutamist sõnumiautentimiskoodide arvutamisel. 2014. aasta lõpul ei ole mei-

Primitiiv	Keskpikk	Ülipikk
Sümmeetriline	128	256
Räsifunktsioon	256	512
RSA	3072	15360
Diskreetne logaritm	3072	15360
ECDL	256	512

Tabel 1. ENISA 2014. aasta parameetrite soovitused

le teada, et selles osas oleks midagi muutunud ja et räsifunktsioon SHA-256 kasutusele võetud oleks.

2.4 Võtmepikkuste soovitused ja primitiivide murdmise täpsustatud ressursitarve

Novembris 2014 üllitas ENISA (*European Union Agency for Network and Information Security*) oma soovitused krüptograafiliste algoritmide, võtmepikkuste ja muude parameetrite kohta [100], mis täiendavad ECRYPT II aruannet [31].

Üldjoontes jäävad ECRYPT II soovitused kehtima, kuid ENISA on oma aruandes pikendanud ajahorisonti, lisades keskpika (kuni 10 aastat) ning ülipika (30-50 aastat) perspektiivi. Keskpikas perspektiivis kasutusele võetavate uute süsteemide jaoks loetakse sobivaks turvatase, mis vastab 128-bitisele sümmeetrilisele tuvatasemele. See tähendab, et ENISA ei soovita uutes süsteemides enam kasutada algoritme RSA2048 ja SHA2-224. RSA puhul on keskpikas perspektiivis soovitatav võtmepikkus vähemalt 3072 bitti.

ENISA täpsemad soovitused võtmepikkuste ja teiste parameetrite kohta on toodud tabelis 1. Sümmeetriliste primitiivide parameetrid on plokk- ja jadašifrite võtmepikkused; RSA, diskreetse logaritmi ja elliptkõverate diskreetse logaritmi (ECDL) puhul on tabelis vastavate moodulite pikkused. Räsifunktsioonide korral on tabelis 1 toodud soovitatav väljundjada pikkus.

Sümmeetrilistest algoritmidest soovitab ENISA uutes süsteemides kasutamiseks šifreid AES ja Camellia, kuid mitte enam šifrit Blowfish. Räsifuktsiooni SHA-3 standardimisprotsess pole 2014. aastaga ikka veel lõppenud, seega on ka ENISA aruanne äraootaval seisukohal, soovitades selle räsifunktsiooni juurutamisega alustada pärast ametliku standardi staatuse saavutamist.

Alates aruande [17] väljaandmisest 2013. aasta lõpul pole olulisi krüptoanalüütilisi läbimurdeid toimunud. See tähendab, et aruandes toodud hinnangud erinevate primitiivide murdmiseks vajalikule arvutiressursile on jätkuvalt üldjoontes õiged. Üllataval kombel pole toimunud ka arvutusvõimsuse märgatavat odavnemist, pigem vastupidi. Aruandes [17] kasutasime Schneieri ja Walkeri hinnangut, et 2^{61} operatsiooni (üks protsessoriaasta) maksab umbes $\$350$. 2015. aasta märtsis maksab Amazoni pilvtöötluspakett m3.medium aastas $\$372$.[2] Selle eest saab ühe kuni $3,3$ GHz protsessori, mis teeb aastaseks arvutusvõimsuseks

$$3,3 \cdot 10^9 \cdot 60 \cdot 60 \cdot 24 \cdot 365 \approx 2^{56,53}$$

[2] `http://aws.amazon.com/ec2/pricing/`, viimati vaadatud 25. märtsil 2015.

operatsiooni. Kasutades aruande [17] hinnanguid, on ühe RSA1024 mooduli murdmiseks vaja 2^{73} ning SHA-1 kollisiooni leidmiseks 2^{74} operatsiooni. Nende arvutuste hinnad on siis täpsustatult vastavalt

$$\frac{2^{73}}{2^{56{,}53}} \cdot \$372 \approx \$33\,760\,000 \quad \text{ja} \quad \frac{2^{74}}{2^{56{,}53}} \cdot \$372 \approx \$67\,520\,000\,.$$

Rõhutame veelkord, et leitud väärtused on ligikaudsed ning nende usaldusväärsus sõltub sisendandmete täpsusest. Kuna aruandes [17] kasutatud Schneieri ja Walkeri hinnang osutus Amazoni pilvetöötluspaketi jõudluse mõttes liiga optimistlikuks, on siinses uuringus arvutused uuesti tehtud ning RSA1024 ja SHA-1 murdmiseks vajaliku eelarve hinnangut suurusjärgu võrra ülespoole parandatud.

Samuti märgime, et meie arvutused põhinevad laiatarbe arvutusvõimsuse maksumusel. Eririistvara, nt FPGA-de (*Field Programmable Gate Array*), kasutamisel võib ründe kogumaksumus olla mitme suurusjärgu võrra väiksem.[3] [4]

2.5 Kettakrüptolahendused

Pärast Windows XP ametliku eluea lõppu aprillis 2014 teatasid krüptotoote TrueCrypt arendajad, et ka nemad ei arenda oma toodet enam edasi. Ametliku põhjendusena toodi vajaduse puudumine, kuivõrd Windowsi hilisemad versioonid pakuvad operatsioonisüsteemi integreeritud BitLocker lahendust. Samuti on teiste operatsioonisüsteemide krüptolahenduste tugi juba ammu sobiv igapäevaseks kasutamiseks (nt OSX-il FileVault2 ning Linuxil dm-crypt koos LUKS-konteineriga).

TrueCrypti toe lõppemise teade sisaldas küll väidet TrueCrypti muutumisest ebaturvaliseks, kuid see on arvatavasti pigem tema loojate viis juhtida tähelepanu vajadusele kasutada aktiivse toega krüptotarkvara. 2015. aastal avaldati sõltumatu auditiaruanne [32], milles kirjeldatu põhjal ei leitud auditi käigus TrueCrypti lähtekoodist olulisi puudujääke ega tagauksi. Kõige suuremaks probleemiks osutus Windows API viga juhuarvude genereerimisel, kuid selle vea esinemise tõenäosus on väike.

Seega pole tungivat krüptograafilist põhjust TrueCrypti kasutamisest loobuda, pigem on probleemiks tootetoe puudumine. Missioonikriitilistes rakendustes soovitame seetõttu kasutada operatsioonisüsteemide tootjate poolt pakutavaid lahendusi.

[3] `http://nsa.unaligned.org/`

[4] `http://blog.ptsecurity.com/2014/12/4g-security-hacking-usb-modem-and-sim.html`

3 Elliptkõverate krüptograafia

Üks uuringu [17] olulisemaid järeldusi on, et paljude klassikaliste avaliku võtme krüptoprimitiivide turvalisuse tagamiseks tuleb kasutatavad moodulid ja võtmed valida üsna pikad. Näiteks peaks nii RSA algoritmi kui Diffie-Hellmani võtmevahetuse ehitama vähemalt 2048-bitistele moodulitele.

Niisugune nõue toob aga endaga kaasa probleeme teostuses: pikenevad krüptogrammid ning aeglustuvad neid moodustavad ja kasutavad rakendused. Seetõttu on krüptograafiakogukond otsimas alternatiive klassikalistele avaliku võtme algoritmidele ja üheks kõige huvitavamaks alternatiiviks on selles vallas elliptkõverate (varasemas kirjanduses tuntud ka kui *elliptilised kõverad*) algebrale tuginevad krüptosüsteemid, mille kirjeldamine ongi praeguse jaotise eesmärk.

Järgnevas eeldame, et lugeja on kursis algebraliste struktuuride (eriti lõplike korpuste ja rühmade) teooriaga ülikooli standardse sissejuhatava abstraktse algebra kursuse tasemel [91]. Ülevaate kirjutamisel elliptkõveratest oleme tuginenud Joseph H. Silvermani ettekandele [99].

3.1 Elliptkõverad

Elliptkõvera defineerimiseks fikseeritakse kõigepealt mingi (kas lõplik või lõpmatu) korpus $\mathbb{F}$ ning kaks elementi $a, b \in \mathbb{F}$ nii, et diskriminant

$$4a^3 + 27b^2 \neq 0. \tag{1}$$

Elliptkõveraks (Weierstraßi kujul) nimetatakse võrrandi

$$y^2 = x^3 + ax + b$$

lahendite (x, y) hulka. Tingimus (1) kindlustab, et polünoomil $x^3 + ax + b$ on üle korpuse $\mathbb{F}$ kolm erinevat juurt ja et vastav elliptkõver osutub mittesingulaarseks. Põhjustel, mis selguvad veidi hiljem, lisatakse elliptkõvera punktide hulka ka formaalne lõpmatuspunkt $\mathcal{O}$. Kokkuvõtteks on elliptkõvera formaalne definitsioon

$$\mathcal{E} = \{(x, y) : \ y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

Ajaloomärkusena mainime, et elliptkõveratel on ellipsitega ühist ainult väga kaudselt. See on ainult väike osa suuremast elliptiliste integraalide ja funktsioonide teooriast, mille arendamise algse tõuke andis 18. sajandil kerkinud ellipsi kaare pikkuse mõõtmise probleem. Muuhulgas ei ole ellips elliptkõver.

See, milline elliptkõverana defineeritud punktihulk välja näeb, sõltub paljuski baaskorpuse $\mathbb{F}$ omadustest. Näiteks üle reaalarvude korpuse saame ühe kolmandat järku joone, mille näide võrrandi $y^2 = x^3 - 5x + 8$ jaoks on toodud joonisel 1.

Tasub tähele panna, et saadav joon on sümmeetriline x-telje suhtes.

Lõplikes korpustes muutub pilt tundmatuseni. Joonisel 2 on sama võrrandi $y^2 = x^3 - 5x + 8$ lahendite hulk üle korpuse $GF(37)$.

Näeme, et igasugune geomeetriline intuitsioon kõvera osas kaob, kuid punktihulk on endiselt sümmeetriline.

Weierstraßi kuju ei ole ainus mõeldav alternatiiv, uuritud on ka

- Montgomery kõveraid defineeriva võrrandiga $by^2 = x^3 + ax^2 + x$,
- Edwardsi kõveraid defineeriva võrrandiga $x^2 + y^2 = 1 + dx^2y^2$ (kus $d \in \mathbb{F} \setminus \{0, 1\}$),
- Koblitzi kõveraid defineeriva võrrandiga $y^2 + xy = x^3 + ax^2 + 1$ (kus $a = 0$ või $a = 1$) jt.

Ühel kujul defineeritud elliptkõverat on vahel võimalik teisendada ka teisele kujule, aga mitte alati.

Osutub, et elliptkõvera punktide hulgal on võimalik defineerida liitmistehe, mis rahuldab kõiki meile tuttavaid liitmise omadusi – on assotsiatiivne ja kommutatiivne, leidub nullelement ning liitmise pöördtehe lahutamine. Teisisõnu, see on kommutatiivne ehk Abeli rühm, mis võimaldab omakorda ehitada krüptograafilisi protokolle.

Elliptkõvera punktide hulga liitmistehte üldine formaalne definitsioon ning omaduste tõestamine on praeguse aruande jaoks liiga keeruline; huvitatud lugeja leiab täpsemaid detaile monograafiast [48]. Siinkohal selgitame liitmise üldist tööpõhimõtet, tuginedes tema geomeetrilisele interpretatsioonile reaalarvuliste kõverate korral.

Olgu meil antud kaks erinevat punkti $P, Q \in \mathcal{E} \setminus \{O\}$. Tõmbame läbi nende punktide sirge, leiame selle sirge kolmanda lõikepunkti antud kõveraga (see on alati olemas) ning defineerime summaks $P + Q$ selle lõikepunkti peegelduse x-teljest (mis asub samuti kõveral, sest kõver on sümmeetriline x-telje suhtes). Kogu operatsiooni illustreerib joonis 3.

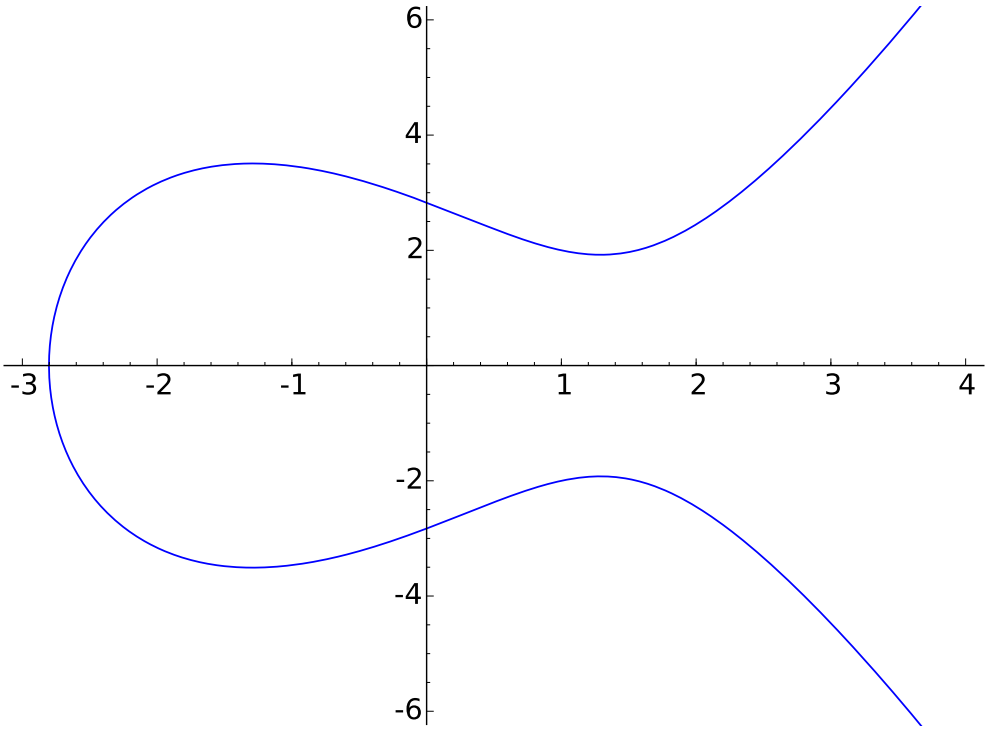
Kui tuleb liita kaks võrdset punkti, leitakse kõvera lõikaja asemel lihtsalt puutuja selles punktis. Punkt O loetakse asuvaks lõpmatuses y-telje sihis ning ta käitub punktide liitmise suhtes nagu nullelement. Elliptkõvera punkti vastandelemendiks on temaga x-telje suhtes sümmeetriline punkt.

Standardse algebralise traditsiooni kohaselt tähistatakse

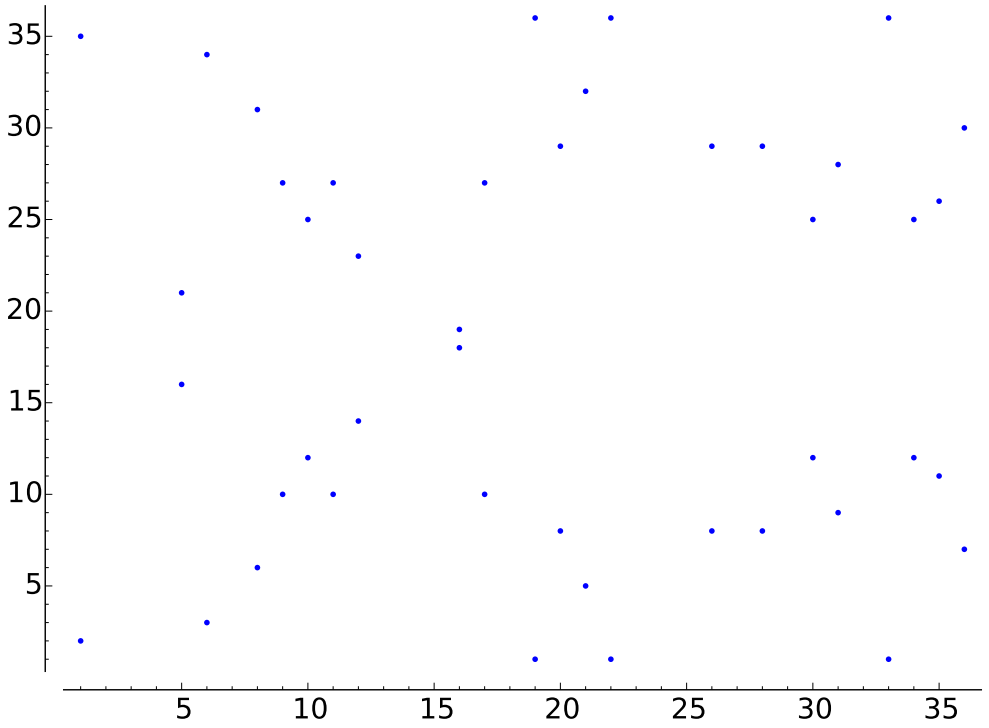
$$P + P = 2P,\ P + P + P = 3P, \ldots, \underbrace{P + P + \ldots + P}_{n} = nP \quad \text{jne.}$$

3.2 Diskreetse logaritmi probleem elliptkõveratel

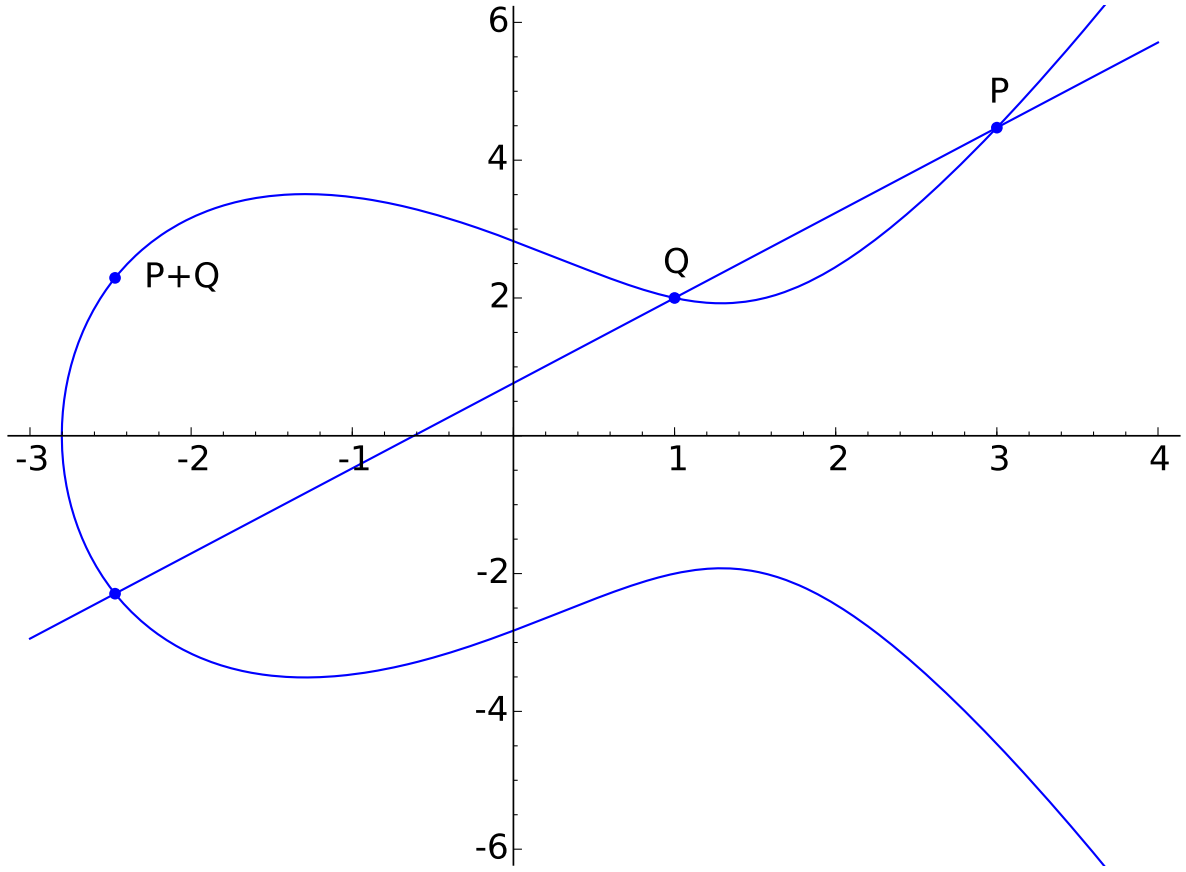
Mitmed klassikalised asümmeetrilised krüptograafilised primitiivid on defineeritavad üle suvalise (kommutatiivse) rühma, näiteks Diffie-Hellmani võtmevahetus, ElGamali krüptosüsteem jt (vt ka [17], jaotis 2.2.2). Nende süsteemide turvalisus põhineb eeldusel, et diskreetse logaritmi ülesanne on vastavas rühmas raske. Käesolevas jaotises vaatlemegi seda ülesannet elliptkõvera punktide rühmal.



Joonis 1. Elliptkõver $y^2 = x^3 - 5x + 8$ üle reaalarvude



Joonis 2. Elliptkõver $y^2 = x^3 - 5x + 8$ üle korpuse $GF(37)$



Joonis 3. Punktide liitmine elliptkõveral

Kuna elliptkõverate punktide rühmad defineeritakse reeglina aditiivses notatsioonis, kõlab diskreetse logaritmi ülesanne nende puhul järgmiselt.

> On antud punktid $P, Q \in \mathcal{E}$. Leia naturaalarv n nii, et $Q = nP$.

Diskreetse logaritmi lahendamise kiirus sõltub tugevalt alloleva rühma ehitusest. Näiteks kiireim teadaolev algoritm diskreetse logaritmi arvutamiseks rühmas $(\mathbb{Z}_p^*, \cdot, 1)$ (kus p on algarv) töötab ajas

$$O\left(e^{c\sqrt[3]{(\log p)(\log\log p)^2}}\right),$$

mis kasvab mooduli p pikkuse $\log p$ suhtes kiiremini kui suvaline polünoom, kuid aeglasemalt kui eksponent. See keerukus on muuhulgas põhjuseks, miks 1024-bitised jäägiklassirühmade diskreetse logaritmi põhiseid krüptosüsteeme ei saa enam turvalisteks lugeda.

Seevastu kiireimad üldised algoritmid, mis leiavad diskreetseid logaritme üle korpuse $\mathbb{Z}_p$ defineeritud elliptkõvera punktide rühmas, töötavad ajas $O(\sqrt{p})$. On huvitav märkida, et need algoritmid ei kasuta kuidagi eriliselt elliptkõvera punktide hulga struktuuri. Teisisõnu, elliptkõvera punktide rühm tundub praeguse teadmuse põhjal käituvat nagu üldine, geneeriline rühm.

Suurus $O(\sqrt{p})$ on mooduli p pikkuse $\log p$ suhtes eksponentsiaalne. See omakorda tähendab, et võrreldes jäägiklassirühmadel defineeritud krüptosüsteemidega võimaldavad

elliptkõverate krüptosüsteemid hakkama saada palju lühemate moodulite, aga järelikult ka võtmepikkustega. Tuleb siiski meeles pidada, et niisugune olukord on võimalik ainult tänu asjaolule, et elliptkõverate punktide rühmadel pole kiiremat üldist diskreetse logaritmi leidmise algoritmi teada. Endiselt võib juhtuda, et mingi konkreetse rühma korral osutub probleem lihtsaks. Samuti võidakse edasise uurimistöö käigus leida asümptootiliselt tõhusamaid üldisi algoritme, mis sunnivad ka elliptkõverate krüptograafia puhul pikemaid võtmepikkusi kasutama.

3.3 ECDH ja ECDSA

Sagedamini kasutatavad elliptkõverate krüptograafia protokollid on Diffie-Hellmani võtmevahetus (ECDH) ning digitaalallkirjaalgoritm ECDSA (vt ka [17], jaotis 2.2.2). Oma konstruktsioonilt meenutavad nad tavalist Diffie-Hellmani võtmevahetust ning algoritmi DSA; vaid arvutusteks vajalikud algebralised operatsioonid on viidud elliptkõverate punktide rühmadesse. Täpsemalt defineeritakse need operatsioonid standardites [34] ja [1].

3.4 Elliptkõveral põhineva krüptosüsteemi ülesseadmine

Elliptkõverate põhised krüptosüsteemid on algebraliselt keerukamad kui RSA, ElGamal või teised klassikalised asümmeetrilised algoritmid. Seetõttu on ka nende ülesseadmine keerulisem protsess ning hõlmab endas pikemat valikute jada. Üldjoontes tuleb teha järgmised sammud.

1) *Baaskorpuse valik.* Reeglina on selleks kas $GF(p)$ suure algarvu p korral või $GF(2^m)$ mingi naturaalarvu m jaoks. Baaskorpuse valik tähendab siis vastavalt arvu p või m valimist. Edaspidises piirdume algarvulise järguga korpuse juhuga.

2) *Kõverat määrava võrrandi valimine.* Sisuliselt tähendab see võrrandi tüübile vastavate kordajate valimist, näiteks Weierstraßi kõvera puhul tuleb fikseerida kordajad a ja b.

3) *Alamrühma valimine.* Krüptograafilisteks rakendusteks valitakse reeglina elliptkõvera punktide rühma $\mathcal{E}$ mingi tsükliline alamrühm, mis fikseeritakse moodustava punkti $P_1 = (x_1, y_1)$ kaudu. Ka selle punkti järk (mis on ühtlasi saadava tsüklilise alamrühma järguks) ℓ valitakse algarvuline. Krüptograafiliste rakenduste jaoks on oluline, et *kofaktor* $h = \frac{|\mathcal{E}|}{\ell}$ oleks väike (st et moodustuv alamrühm $\langle P_1 \rangle$ oleks võimalikult suur).

4) *Võtmepaari moodustamine.* Selleks, et anda ECDSA digitaalsignatuuri, valib allkirjastaja salajaseks võtmeks naturaalarvu $d \in (0, \ell - 1)$ ning arvutab avaliku võtme dP.

Algarvulisel korpusel põhineva (Weierstraßi kujul) elliptkõvera parameetrite komplekt on kokkuvõttes (p, a, b, P_1, ℓ, h), salajase ja avaliku võtme paar on (d, dP).

Tüüpilises teostuses fikseeritakse kõvera parameetrid süsteemiüleselt ning käivitamise ajal saab valida ainult avaliku ja salajase võtme. Peamiseks põhjuseks on siinkohal tõhusus, sest paljud teostuse optimeerimise meetodid sõltuvad kõvera parameetrite konkreetsetest väärtustest. Sellise lähenemise puudus on vähene paindlikkus ja vajadus kõvera vahetamisel vahetada välja palju süsteemi komponente, kusjuures mõnede komponentide (nt kiipkaartide) väljavahetamine võib osutuda kulukaks.

3.5 Elliptkõverate omadused ja nende võrdlus

Kuna elliptkõverate algebraline struktuur on küllalt mitmetahuline, on ka nende kõverate võrdluskriteeriumid mittetriviaalsed. Järgnevas käsitluses tugineme Daniel Bernsteini ja Tanja Lange uurimusele turvalistest kõveratest [41].

Bernstein ja Lange on valinud täpsemaks analüüsiks 20 konkreetset kõverat, millest osa on olulised ka praeguse uuringu käsitlusalas. P-256 on üks levinumaid USA standardiorganisatsiooni NIST (*National Institute of Standards and Technology*) kõveraid, brainpoolP256t1 on tema alternatiiv Brainpool'i valikust (täpsemalt käsitleme elliptkõverate standardeid jaotises 3.6). Lisaks kuuluvad meie analüüsi huviorbiiti Bernsteini enda kõverad Curve25519 ja Ed25519. Need kõverad on biratsionaalselt ekvivalentsed ning jagavad seetõttu praktiliselt kõiki turvaomadusi, seega piirduvad Bernstein ja Lange oma uuringus vaid Curve25519 mainimisega. Nende kõverate erinevus seisneb kasutusvaldkonnas: Curve25519 sobib Diffie-Hellmani võtmevahetuse, Ed25519 aga digitaalsignatuuride jaoks.

Ülejäänud Bernsteini ja Lange uuringu kõveratest jagunevad kahte klassi – ühelt poolt standarditud kõverad (nt P-224, secp256k1, brainpoolP384t1 jt), teisalt aga teadusartiklites pakutud (nt M-221, E-222, Curve1174 jt). Ka nende jaoks kehtivad üldjoontes samad järeldused kui siinses aruandes täpsemalt käsitletud kõverate puhul: kõigil standardsetel kõveratel on algebralisi või krüptograafilisi nõrkusi, kuid krüptograafilisest seisukohast turvalised kõverad on standardites ja levinud teostustes täielikult toetamata. Neid probleeme käsitleme lähemalt jaotistes 3.6, 3.7 ning 3.9.

Bernstein ja Lange jagavad neid huvitavad kriteeriumid kolme suurde klassi.

Parameetrite omadused

Bernstein ja Lange nõuavad elliptkõvera parameetritelt, et p peab olema algarv, et defineeriv võrrand peab olema korrektne ning et baaspunkt P_1 peab asuma saadud kõveral. Kõik vähegi mõistlikud (algarvulisel korpusel põhinevad) elliptkõverad rahuldavad neid tingimusi, sealhulgas ka P-256, brainpoolP256t1 ja Curve25519.

Diskreetse logaritmi keerukus

Kõik elliptkõverate krüptosüsteemid tuginevad diskreetse logaritmi leidmise keerukusele vastavas rühmas, seega peab see probleem teadaolevate rünnete valguses tõepoolest ka raske olema.

ρ-meetodite keerukus

ρ-meetodid on üldised võtted diskreetse logaritmi arvutamiseks suvalises (tsüklilises) rühmas. Rakendatuna elliptkõverate rühmadele on nende keskmiseks oodatavaks tööajaks $\frac{\sqrt{\pi}}{2}\sqrt{\ell}$ punktiliitmist. Berstein ja Lange nõuavad, et praktilise turvalisuse saavutamiseks peab see suurus ületama 2^{100}, ning kontrollivad, et kõigi neid huvitavate kõverate (sh nii NIST P-256, Curve25519 kui brainpoolP256t1) puhul on see nõue täidetud.

ρ-meetodite jõudlust on võimalik veidi parandada, kui teatud diskriminant on piisavalt väike. Seega esitavad Bernstein ja Lange selle diskriminandi kohta eraldi suurusnõude ning kontrollivad, kas ka see nõue on NIST P-256, Curve25519 ja brainpoolP256t1 puhul täidetud.

Diskreetse logaritmi ülesande teisendused

Ühes rühmas esitatud diskreetse logaritmi ülesannet on võimalik (teatud tingimustel) teisendada ülesandeks teises rühmas, kus selle lahendamine võib osutuda lihtsamaks. Näiteks juhul kui $\ell = p$, on rühm $\langle P_1 \rangle$ isomorfne aditiivse rühmaga $\mathbb{Z}_p$, milles diskreetsete logaritmide leidmine on triviaalne. On võimalikud ka algebraliselt keerukamad teisendused ning Bernstein ja Lange nõuavad, et need teisendused ei viiks diskreetse logaritmi murdumisele. Enamiku kõverate (sh NIST P-256, Curve25519 ja brainpoolP256t1) puhul ei osutu see probleemiks.

Kõvera rangus

See omadus on seoses 2013. aastal puhkenud NSA luureskandaaliga elliptkõverate juures kõige rohkem kõneainet tekitanud. Lühidalt kokku võetuna taandub see nõudele, et kõigi parameetrite valik peab kõvera defineerimise protsessis olema võimalikult üheselt määratud ega tohi sõltuda suurtest konkreetse põhjenduseta (pealtnäha) juhuslikest arvudest.

Kui see nõue pole täidetud, on põhimõtteliselt võimalik, et kõvera valija teab mitteavalikku rünnet, mis rakendub potentsiaalselt genereeritavate kõverate hulga mingile väiksele alamhulgale, ja kasutab seda pealtnäha juhuslikku parameetrit, et kindlustada kõvera sattumine sellesse nõrka alamhulka.

Standardites mainitutest kukuvad selles kriteeriumis täielikult läbi NISTi kõverad (sh P-256), mis kasutavad ilma ühegi põhjenduseta võetud suuri arve. Brainpooli kõverate (nt brainpoolP256t1) seis on parem, kuigi ka nemad sõltuvad konstantidest, mille valik pole ilmselt ühene (nt $\sqrt{2}$, $\sqrt{3}$, e, π).

On ka terve rida kõveraid, mille valiku Bernstein ja Lange täielikult rangeks kuulutavad (sh Bernsteini enda Curve25519); täpse nimekirja leiab lugeja allikast [41].

Elliptkõvera süsteemi turvaline teostatavus

Elliptkõverate krüptograafia turvalisuseks on diskreetse logaritmi keerukus küll tarvilik, kuid kaugeltki mitte piisav. Arvesse tuleb võtta paljusid ebakorrektse teostuse põhjustatud võimalikke probleeme. Seejuures osutub, et kõik kõverad pole potentsiaalsete teostusohtude osas kaugeltki võrdsed.

Redelid

Kõige aeganõudvam operatsioon elliptkõverate krüptograafias on punkti korrutamine skalaariga. Montgomery kõverate puhul saab kasutada Montgomery redelit (*Montgomery ladder*), mis muuhulgas pakub üsna lihtsate vahenditega kindlust külgrünnete vastu. Montgomery kõverad pole ainus võimalik valik, kuid Bernsteini ja Lange hinnangul on nad üks lihtsamini turvaliselt teostatav kõverate klass.

NIST P-256 ja brainpoolP256t1 kui Weierstraßi kõverad ei võimalda kahjuks lihtsate vahenditega turvatavat teostust, Curve25519 kui Montgomery kõver aga küll.

Keeruturvalisus

On olemas klass ründeid, kus ründaja saadab ohvrile punkti, mille ta on valinud protokolliväliselt või hoopis valelt kõveralt. Kui ohver punkti õigsust eraldi ei kontrolli, võib ta ar-

vutuste tulemusena näiteks oma salajase võtme kohta teavet lekitada. Bernstein ja Lange nõuavad, et isegi kui privaatsete arvutuste tegija ei realiseeri ühtki lisakontrolli, peab lekkinud info põhjal salajase võtme leidmine endiselt keeruline olema, ning nimetavad seda omadust keeruturvalisuseks (*twist security*). Nad analüüsivad parimaid teadaolevaid ründeid ning leiavad, et näiteks NIST P-256 ja Curve25519 peavad neile rünnetele hästi vastu, aga brainpoolP256t1 mitte. See ei tähenda, et viimaiti mainitu oleks murtav, kuid tähendab, et tema teostamisel tuleb olla hoolikam ning realiseerida sisendandmete õigsuse kontrollid.

Tehete täielikkus

Nagu nägime jaotises 3.1, on elliptkõvera punktide liitmine algebraliselt küllalt keeruline operatsioon. Ülesande muudab veel keerukamaks vajadus käsitleda eraldi neutraalset elementi ning juhtu, kus elementi liidetakse iseendale. Eri kõverate eri esituste korral on liitmise realisatsioonis läbivaatamist vajavate juhtude arv erinev. Lihtsamal juhul eksisteerib üks valem, mis katab kõik juhud, keerukamal juhul tuleb liitmise defineerimiseks kasutada mitut *if-then*-lauset. Bernstein ja Lange eelistavad kõveraid, mis võimaldavad lihtsat, ilma suurema hulga eranditeta punktiliitmise definitsiooni. Nende kriteeriumi läbib küll Curve25519, kuid mitte P-256 ega brainpoolP256t1.

Jällegi ei tähenda selle kriteeriumi rahuldamata jäämine kõvera olemuslikku ebaturvalisust, vaid vajadust tema realiseerimisel tavalisest hoolikam olla.

Eristuvus juhuslikest väärtustest

Oma standardesituses on elliptkõvera punktid juhuslikest väärtustest lihtsasti eristatavad ja see võib põhjustada probleeme mõnedes krüptograafilistes protokollides kasutamisel. Standardne meetod selle probleemiga toimetulemiseks on muuta arvutuste käigus punkti esitust, aga see meetod on küllalt veaohtlik. Bernstein ja Lange pakuvad välja ka süstemaatilisema lahenduse ning tõdevad, et nende lahendus töötab näiteks Curve255519, aga mitte P-256 ega brainpoolP256t1 korral.

3.6 Kõverate toetus standardites

Bernsteini ja Lange analüüs käsitleb ainult elliptkõverate algebralisi ja krüptograafilisi omadusi. Paraku ei piisa headest krüptograafilistest omadustest nende kõverate edukaks rakendamiseks. Koostalitlusvõime tagamiseks on tarvis ka rahvusvaheliste standardite ja teostuste tuge. Käesolevas peatükis annamegi lühiülevaate tähtsamatest standarditest, mis elliptkõverate krüptograafiat käsitlevad, jaotistes 3.7 ja 3.8 aga käsitleme elliptkõverate teostusi levinumates tark- ja riistvaralistes komponentides.

Elliptkõverate standardimisega on tegelenud mitmed organisatsioonid, sealhulgas

- SECG (*Standards for Efficient Cryptography Group*) standardiga *SEC 2: Recommended Elliptic Curve Domain Parameters* [12],
- ANSI (*American National Standards Institute*) standardiga *X9.62 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)* [1],
- NIST standardiga *FIPS 186-4: Digital Signature Standard (DSS)* [3],

- IETF (*Internet Engineering Task Force*) standardiga *RFC5480: Elliptic Curve Cryptography Subject Public Key Information* [105],
- WAP (*Wireless Application Protocol*) Forum standardiga *Wireless Transport Layer Security* [10].

Nende standardite omavahelised suhted on üsna segased. Nad viitavad üksteisele ning osaliselt (aga ainult osaliselt) kattuvad defineeritavate kõverate osas. Samuti omistavad erinevad standardid vahel samadele kõveratele erinevaid objektiidentifikaatoried (OID), vt tabel 2. Kõik standardid pole ka terminoloogiliselt järjekindlad, näiteks nimetab SEC 2 mõningaid Weiestraßi kujul olevaid kõveraid eksitavalt Koblitzi kõverateks.

Saksamaa infoturbeametil BSI on olemas oma standard [15], mis kõverate valiku osas viitab edasi allikale RFC5639 [79], lisaks on välja lastud kolm seotud dokumenti RFC6932 [68], RFC69554 [84] ning RFC7027 [83]. Paraku pole ükski neist IETFi standard, vaid pelgalt infomaterjal.

Siinse aruande huviorbiidis olevatest kõveratest on vaieldamatult kõige paremini standarditud P-256, mida katavad neli ülalpool esimesena viidatud standardit. Kõver brainpoolP256t1 on spetsifitseeritud vaid informatsioonilises väljaandes ning Curve25519 ei leia kajastamist üheski standardis. Üheks probleemiks, miks Curve25519 kardetavasti niipea standarditesse ei jõua, on tema ühildumatus ANSI X9.62 nõuetega. Nimelt võimaldab ANSI X9.62 esitada kõveraid ainult Weierstraßi kujul, kuid Curve25519 on Montgomery kõver (ja Ed25519 on Edwardsi kõver). Bernstein on küll väitnud, et Curve25519 on teisendatav Weierstraßi kujule [39], kuid see ei muuda asjaolu, et olemasolevad standardid seda kõverat ennast ei toeta.

Jaotise lõpetuseks tuletame meelde, et kõvera standarditus ei välista võimalust esitada tema kohta patenditaotlusi. Elliptkõverate patentide teema leidis põhjalikku käsitlemist 2013. aasta aruandes [17]. Praeguse aruande huviorbiidis olevatest kõveratest on kõige suurem oht patendivaidluste alla langeda kõveral P-256. Brainpool'i kõverad (sh brainpoolP256t1) on valitud nii, et nende aluseks olevad algarvud ei võimaldaks potentsiaalselt patenditud optimeerimismeetodeid. Paraku kannatab tulemusena seeläbi Brainpool'i kõverate jõudlus [4]. Kõvera Curve25519 kohta on Dan Bernstein esitanud väite, et sellele ei rakendu ükski talle teada olev patent [38].

3.7 Elliptkõverate realiseeritus tarkvaralistes rakendustes

Järgnevalt vaatleme elliptkõverate toe olemasolu tuntumates teekides (OpenSSL [7], GnuTLS [9], NSS [6], Bouncy Castle [2]), keeltes/raamistikes (Java, .NET) ning operatsioonisüsteemides (Windows 7 / 8, OS X 10.10, Ubuntu 14.04 LTS, CentOS 6.6, openSUSE 12.3, Fedora 21, iOS 8.1.1, Android 4.3.1).

OpenSSL on avatud lähtekoodiga keeles C kirjutatud vabavaraline krüptoteek, mis on üks enamlevinuid krüptoteeke UNIXi-laadsetes operatsioonisüsteemides (Solaris, Linux, Mac OS X jmt).

GnuTLS on vabavaraline keeles C kirjutatud krüptoteek, mis realiseerib TLS- ja SSL-protokolli.

Network Security Services (NSS) on avatud lähtekoodiga keeles C kirjutatud vabavaraline krüptoteek, mis on enamasti kasutuses Mozilla toodetes.

Bouncy Castle on vabavaraline krüptoteek, millel on API nii Java- kui ka C#-keelele. Operatsioonisüsteemis Android kasutatakse Bouncy Castle'i piiratud versiooni, mille asemel mõned rakenduste arendajad eelistavad piiramata võimalustega ümberpakendust Spongy Castle [8].

Kõigist neist teekidest on kõige parema elliptkõverate toega OpenSSL, mille valisime ka järgneva ülevaate aluseks. Tabelis 2 on teegis OpenSSL teostatud kõverad koos nende alternatiivsete nimedega teistes teekides, lühikirjelduse ja objektiidentifikaatoriga (OID).

Nime identifikaator(id)	Lühikirjeldus	OID
brainpoolP160r1, brainpoolp160r1 [5]	Brainpool'i kõver üle 160-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.1
brainpoolP160t1, brainpoolp160t1 [5]	Brainpool'i kõver üle 160-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.2
brainpoolP192r1, brainpoolp192r1 [5]	Brainpool'i kõver üle 192-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.3
brainpoolP192t1, brainpoolp192t1 [5]	Brainpool'i kõver üle 192-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.4
brainpoolP224r1, brainpoolp224r1 [5]	Brainpool'i kõver üle 224-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.5
brainpoolP224t1, brainpoolp224t1 [5]	Brainpool'i kõver üle 224-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.6
brainpoolP256r1, brainpoolp256r1 [5]	Brainpool'i kõver üle 256-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.7
brainpoolP256t1, brainpoolp256t1 [5]	Brainpool'i kõver üle 256-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.8
brainpoolP320r1, brainpoolp320r1 [5]	Brainpool'i kõver üle 320-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.9
brainpoolP320t1, brainpoolp320t1 [5]	Brainpool'i kõver üle 320-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.10
brainpoolP384r1, brainpoolp384r1 [5]	Brainpool'i kõver üle 384-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.11
brainpoolP384t1, brainpoolp384t1 [5]	Brainpool'i kõver üle 384-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.12
brainpoolP512r1, brainpoolp512r1 [5]	Brainpool'i kõver üle 512-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.13
brainpoolP512t1, brainpoolp512t1 [5]	Brainpool'i kõver üle 512-bitise algarvulise korpuse	1.3.36.3.3.2.8.1.1.14
c2pnb163v1	X9.62 kõver üle 163-bitise binaarse korpuse	1.2.840.10045.3.0.1
c2pnb163v2	X9.62 kõver üle 163-bitise binaarse korpuse	1.2.840.10045.3.0.2
c2pnb163v3	X9.62 kõver üle 163-bitise binaarse korpuse	1.2.840.10045.3.0.3
c2pnb176v1, c2pnb176w1 [6]	X9.62 kõver üle 176-bitise binaarse korpuse	1.2.840.10045.3.0.4

[5] Kasutatakse krüptoteegis Bouncy Castle
[6] Esineb OID kirjelduses

Nime identifikaator(id)	Lühikirjeldus	OID
c2pnb208w1	X9.62 kõver üle 208-bitise binaarse korpuse	1.2.840.10045.3.0.10
c2pnb272w1, c2pnb272W1 [6]	X9.62 kõver üle 272-bitise binaarse korpuse	1.2.840.10045.3.0.16
c2pnb304w1, c2pnb304W1 [6]	X9.62 kõver üle 304-bitise binaarse korpuse	1.2.840.10045.3.0.17
c2pnb368w1	X9.62 kõver üle 368-bitise binaarse korpuse	1.2.840.10045.3.0.19
c2tnb191v1, X9.62 c2tnb191v1 [7]	X9.62 kõver üle 191-bitise binaarse korpuse	1.2.840.10045.3.0.5
c2tnb191v2, X9.62 c2tnb191v2 [7]	X9.62 kõver üle 191-bitise binaarse korpuse	1.2.840.10045.3.0.6
c2tnb191v3, X9.62 c2tnb191v3 [7]	X9.62 kõver üle 191-bitise binaarse korpuse	1.2.840.10045.3.0.7
c2tnb239v1, X9.62 c2tnb239v1 [7]	X9.62 kõver üle 239-bitise binaarse korpuse	1.2.840.10045.3.0.11
c2tnb239v2, X9.62 c2tnb239v2 [7]	X9.62 kõver üle 239-bitise binaarse korpuse	1.2.840.10045.3.0.12
c2tnb239v3, X9.62 c2tnb239v3 [7]	X9.62 kõver üle 239-bitise binaarse korpuse	1.2.840.10045.3.0.13
c2tnb359v1, X9.62 c2tnb359v1 [7]	X9.62 kõver üle 359-bitise binaarse korpuse	1.2.840.10045.3.0.18
c2tnb431r1, X9.62 c2tnb431r1 [7]	X9.62 kõver üle 431-bitise binaarse korpuse	1.2.840.10045.3.0.20
prime192v1, secp192r1 [6], ansiX9p192r1 [6], P-192 [5], nistp192 [8]	NIST/X9.62/SECG kõver üle 192-bitise algarvulise korpuse	1.2.840.10045.3.1.1
prime192v2, X9.62 prime192v2 [7]	X9.62 kõver üle 192-bitise algarvulise korpuse	1.2.840.10045.3.1.2
prime192v3, X9.62 prime192v3 [7]	X9.62 kõver üle 192-bitise algarvulise korpuse	1.2.840.10045.3.1.3
prime239v1, X9.62 prime239v1 [7]	X9.62 kõver üle 239-bitise algarvulise korpuse	1.2.840.10045.3.1.4
prime239v2, X9.62 prime239v2 [7]	X9.62 kõver üle 239-bitise algarvulise korpuse	1.2.840.10045.3.1.5
prime239v3, X9.62 prime239v3 [7]	X9.62 kõver üle 239-bitise algarvulise korpuse	1.2.840.10045.3.1.6
prime256v1, secp256r1 [6], P-256 [5] [9]	NIST/X9.62/SECG kõver üle 256-bitise algarvulise korpuse	1.2.840.10045.3.1.7
secp112r1	SECG/WTLS kõver üle 112-bitise algarvulise korpuse	1.3.132.0.6
secp112r2	SECG kõver üle 112-bitise algarvulise korpuse	1.3.132.0.7

[7] Kasututatakse Java SunEC-s
[8] Kasutatakse krüptoteegis NSS
[9] Kasutatakse .NET-raamistikus

Nime identifikaator(id)	Lühikirjeldus	OID
secp128r1	SECG kõver üle 128-bitise algarvulise korpuse	1.3.132.0.28
secp128r2	SECG kõver üle 128-bitise algarvulise korpuse	1.3.132.0.29
secp160k1, ansip160k1 [6]	SECG kõver üle 160-bitise algarvulise korpuse	1.3.132.0.9
secp160r1, ansip160r1 [6]	SECG kõver üle 160-bitise algarvulise korpuse	1.3.132.0.8
secp160r2, ansip160r2 [6]	SECG/WTLS kõver üle 160-bitise algarvulise korpuse	1.3.132.0.30
secp192k1, ansip192k1 [6]	SECG kõver üle 192-bitise algarvulise korpuse	1.3.132.0.31
secp224k1, ansip224k1 [6]	SECG kõver üle 224-bitise algarvulise korpuse	1.3.132.0.32
secp224r1, ansip224r1 [6], P-224 [5], nistp224 [8]	NIST/SECG kõver üle 224-bitise algarvulise korpuse	1.3.132.0.33
secp256k1, ansip256k1 [6]	SECG kõver üle 256-bitise algarvulise korpuse	1.3.132.0.10
secp384r1, ansip384r1 [6], P-384 [5] [9]	NIST/SECG kõver üle 384-bitise algarvulise korpuse	1.3.132.0.34
secp521r1, ansip521r1 [6], P-521 [5] [9]	NIST/SECG kõver üle 521-bitise algarvulise korpuse	1.3.132.0.35
sect113r1	SECG kõver üle 113-bitise binaarse korpuse	1.3.132.0.4
sect113r2	SECG kõver üle 113-bitise binaarse korpuse	1.3.132.0.5
sect131r1	SECG/WTLS kõver üle 131-bitise binaarse korpuse	1.3.132.0.22
sect131r2	SECG kõver üle 131-bitise binaarse korpuse	1.3.132.0.23
sect163k1, ansit163k1 [6], K-163 [5], nistk163 [8]	NIST/SECG/WTLS kõver üle 163-bitise binaarse korpuse	1.3.132.0.1
sect163r1, ansit163r1 [6]	SECG kõver üle 163-bitise binaarse korpuse	1.3.132.0.2
sect163r2, ansit163r2 [6], B-163 [5], nistb163 [8]	NIST/SECG kõver üle 163-bitise binaarse korpuse	1.3.132.0.15
sect193r1, ansit193r1 [6]	SECG kõver üle 193-bitise binaarse korpuse	1.3.132.0.24
sect193r2, ansit193r2 [6]	SECG kõver üle 193-bitise binaarse korpuse	1.3.132.0.25
sect233k1, ansit233k1 [6], K-233 [5], nistk233 [8]	NIST/SECG/WTLS kõver üle 233-bitise binaarse korpuse	1.3.132.0.26
sect233r1, ansit233r1 [6], B-233 [5], nistb233 [8]	NIST/SECG/WTLS kõver üle 233-bitise binaarse korpuse	1.3.132.0.27
sect239k1, ansit239k1 [6]	SECG kõver üle 239-bitise binaarse korpuse	1.3.132.0.3

Nime identifikaator(id)	Lühikirjeldus	OID
sect283k1, ansit283k1 [6], K-283 [5], nistk283 [8]	NIST/SECG kõver üle 283-bitise binaarse korpuse	1.3.132.0.16
sect283r1, ansit283r1 [6], B-283 [5], nistb283 [8]	NIST/SECG kõver üle 283-bitise binaarse korpuse	1.3.132.0.17
sect409k1, ansit409k1 [6], K-409 [5], nistk409 [8]	NIST/SECG kõver üle 409-bitise binaarse korpuse	1.3.132.0.36
sect409r1, ansit409r1 [6], B-409 [5], nistb409 [8]	NIST/SECG kõver üle 409-bitise binaarse korpuse	1.3.132.0.37
sect571k1, ansit571k1 [6], K-571 [5], nistk571 [8]	NIST/SECG kõver üle 571-bitise binaarse korpuse	1.3.132.0.38
sect571r1, ansit571r1 [6], B-571 [5], nistb571 [8]	NIST/SECG kõver üle 571-bitise binaarse korpuse	1.3.132.0.39
wap-wsg-idm-ecid-wtls1	WTLS kõver üle 113-bitise binaarse korpuse	2.23.43.1.4.1
wap-wsg-idm-ecid-wtls3, sect163k1, K-163	NIST/SECG/WTLS kõver üle 163-bitise binaarse korpuse	2.23.43.1.4.3
wap-wsg-idm-ecid-wtls4	SECG kõver üle 113-bitise binaarse korpuse	2.23.43.1.4.4
wap-wsg-idm-ecid-wtls5	X9.62 kõver üle 163-bitise binaarse korpuse	2.23.43.1.4.5
wap-wsg-idm-ecid-wtls6	SECG/WTLS kõver üle 112-bitise algarvulise korpuse	2.23.43.1.4.6
wap-wsg-idm-ecid-wtls7	SECG/WTLS kõver üle 160-bitise algarvulise korpuse	2.23.43.1.4.7
wap-wsg-idm-ecid-wtls8	WTLS kõver üle 112-bitise algarvulise korpuse	2.23.43.1.4.8
wap-wsg-idm-ecid-wtls9	WTLS kõver üle 160-bitise algarvulise korpuse	2.23.43.1.4.9
wap-wsg-idm-ecid-wtls10, sect233k1, K-233	NIST/SECG/WTLS kõver üle 233-bitise binaarse korpuse	2.23.43.1.4.10
wap-wsg-idm-ecid-wtls11, sect233r1, B-233	NIST/SECG/WTLS kõver üle 233-bitise binaarse korpuse	2.23.43.1.4.11
wap-wsg-idm-ecid-wtls12, secp224r1, P-224	WTLS kõver üle 224-bitise algarvulise korpuse	2.23.43.1.4.12

Tabel 2. Elliptkõverad teegis OpenSSL

Tabelis 3 on toodud eelpoolvaadeldud kõverate realiseeritus ka teistes teekides ja raamistikes.

Kõver	OpenSSL 1.0.1 / 1.0.2	GnuTLS 3.3.11	NSS 3.17.3	Java SE 7 / 8 (SunEC)	Bouncy Castle 1.51	.NET 4.5
brainpoolP160r1	-/+[10]	-	-	-	+	-
brainpoolP160t1	-/+	-	-	-	+	-
brainpoolP192r1	-/+	-	-	-	+	-
brainpoolP192t1	-/+	-	-	-	+	-
brainpoolP224r1	-/+	-	-	-	+	-
brainpoolP224t1	-/+	-	-	-	+	-
brainpoolP256r1	-/+	-	-	-	+	-
brainpoolP256t1	-/+	-	-	-	+	-
brainpoolP320r1	-/+	-	-	-	+	-
brainpoolP320t1	-/+	-	-	-	+	-
brainpoolP384r1	-/+	-	-	-	+	-
brainpoolP384t1	-/+	-	-	-	+	-
brainpoolP512r1	-/+	-	-	-	+	-
brainpoolP512t1	-/+	-	-	-	+	-
c2pnb163v1	+	-	+	-	+	-
c2pnb163v2	+	-	+	-	+	-
c2pnb163v3	+	-	+	-	+	-
c2pnb176v1	+	-	+	-	+	-
c2pnb208w1	+	-	+	-	+	-
c2pnb272w1	+	-	+	-	+	-
c2pnb304w1	+	-	+	-	+	-
c2pnb368w1	+	-	+	-	+	-
c2tnb191v1	+	-	+	+	+	-
c2tnb191v2	+	-	+	+	+	-
c2tnb191v3	+	-	+	+	+	-
c2tnb239v1	+	-	+	+	+	-
c2tnb239v2	+	-	+	+	+	-
c2tnb239v3	+	-	+	+	+	-
c2tnb359v1	+	-	+	+	+	-
c2tnb431r1	+	-	+	+	+	-
prime192v1	+	+	+	+	+	-
prime192v2	+	-	+	+	+	-
prime192v3	+	-	+	+	+	-
prime239v1	+	-	+	+	+	-
prime239v2	+	-	+	+	+	-
prime239v3	+	-	+	+	+	-
prime256v1	+	+	+[11]	+	+	+
secp112r1	+	-	+	+	+	-
secp112r2	+	-	+	+	+	-

[10] Märge -/+ tähendab, et kõver on toetatud OpenSSL teegi versioonis 1.0.2, kuid mitte versioonis 1.0.1.
[11] Vaikimisi toetatud kõverad. Teiste kõverate tugi kaasatakse kompileerimisel vastava lipu seadmisel.

Kõver	OpenSSL 1.0.1 / 1.0.2	GnuTLS 3.3.11	NSS 3.17.3	Java SE 7 / 8 (SunEC)	Bouncy Castle 1.51	.NET 4.5
secp128r1	+	-	+	+	+	-
secp128r2	+	-	+	+	+	-
secp160k1	+	-	+	+	+	-
secp160r1	+	-	+	+	+	-
secp160r2	+	-	+	+	+	-
secp192k1	+	-	+	+	+	-
secp224k1	+	-	+	+	+	-
secp224r1	+	+	+	+	+	-
secp256k1	+	-	+	+	+	-
secp384r1	+	+	+[11]	+	+	+
secp521r1	+	+	+[11]	+	+	+
sect113r1	+	-	+	+	+	-
sect113r2	+	-	+	+	+	-
sect131r1	+	-	+	+	+	-
sect131r2	+	-	+	+	+	-
sect163k1	+	-	+	+	+	-
sect163r1	+	-	+	+	+	-
sect163r2	+	-	+	+	+	-
sect193r1	+	-	+	+	+	-
sect193r2	+	-	+	+	+	-
sect233k1	+	-	+	+	+	-
sect233r1	+	-	+	+	+	-
sect239k1	+	-	+	+	+	-
sect283k1	+	-	+	+	+	-
sect283r1	+	-	+	+	+	-
sect409k1	+	-	+	+	+	-
sect409r1	+	-	+	+	+	-
sect571k1	+	-	+	+	+	-
sect571r1	+	-	+	+	+	-
wap-wsg-idm-ecid-wtls1	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls3	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls4	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls5	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls6	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls7	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls8	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls9	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls10	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls11	+	-	-	-	-	-
wap-wsg-idm-ecid-wtls12	+	-	-	-	-	-

Tabel 3. Elliptkõverate tugi erinevates teekides ja raamistikes

Windows 7 / 8 ja OS X 10.10

Nende operatsioonisüsteemide korral vaatlesime süsteemsete vahendite võimet käsitleda X.509 sertifikaate, milles genereerimisel on kasutatud elliptkõverate baasil moodustatud võtmeid. Testisime nii endasigneeritud kui ka test-CA poolt RSA võtmega signeeritud sertifikaate.

Windows'is avasime sertifikaadid detailide vaatamiseks ning kontrollisime, kas Windows suutis tuvastada sertifikaadis oleva avaliku võtme pikkuse ja parameetrid ning kas endasigneeritud sertifikaati õnnestus sertifikaadihoidlasse edukalt paigaldada.

OS X-is käitusime analoogiliselt ning proovisime paigaldada samu sertifikaate süsteemsesse võtmerõngasse (*keychain*).

Testid edukalt läbinud kõverad on tabelis 4.

Kõver	Windows 7 / 8	OS X 10.10
prime192v1	-	+
prime256v1	+	+
secp384r1	+	+
secp521r1	+	+

Tabel 4. Elliptkõverate tugi operatsioonisüsteemides Windows ja OS X

Ubuntu 14.04 LTS, CentOS 6.6, openSUSE 12.3, Fedora 21

Linuxites on üks levinumaid ning olulisemaid krüptoteeke OpenSSL. Sellest sõltuvad näiteks Apache, OpenSSH, OpenSC jt rakendused. Linuxi võime kasutada vaikeseadetes erinevaid kõveraid on seega suuresti seotud vastava versiooni jaoks pakutava süsteemse OpenSSL-iga. Tabelis 5 on kokkuvõte kõveratest, mida toetavad erinevatele distributsioonidele pakendatud OpenSSL'id.

OS	OpenSSL	Kõverad
Ubuntu 14.04 LTS	1.0.1f	Kõik vastava versiooni kõverad tabelis 3
CentOS 6.6	1.0.1e-fips	prime256v1, secp384r1, secp521r1
openSUSE 12.3	1.0.1e	Kõik vastava versiooni kõverad tabelis 3
Fedora 21	1.0.1j-fips	prime256v1, secp384r1, secp521r1

Tabel 5. Elliptkõverate tugi erinevates Linuxi versioonides vaikehäälestuses

iOS 8.1.1

Analoogiliselt OS X-ga testisime süsteemsete vahendite võimet käsitleda samu X.509 testsertifikaate. iOS-ile arusaadava sertifikaadi korral avasime profiilipaigaldaja ning võimaldasime vastava sertifikaadi süsteemi installeerida. iOS-i profiilipaigaldaja tundis ära järgmised kõverad:

- brainpoolP192r1
- brainpoolP192t1
- brainpoolP224r1
- brainpoolP224t1
- brainpoolP256r1
- brainpoolP256t1
- brainpoolP384r1
- brainpoolP384t1
- c2tnb191v1
- c2tnb191v2
- c2tnb191v3
- prime192v1
- prime192v2
- prime192v3
- prime256v1
- secp192k1
- secp224k1
- secp224r1
- secp256k1
- secp384r1
- secp521r1
- wap-wsg-idm-ecid-wtls12

Android 4.3.1

Testides elliptkõverate tuge operatsioonisüsteemil Android (versioon 4.3.1), moodustasime samuti endasigneeritud X.509 sertifikaadid kõigi tabelis 3 toodud kõverate jaoks. Selleks, et Android oleks nõus neid sertifikaadilattu (*certificate store*) lisama, tuli neile määrata laiendus `CA:true` (mis defineerib edasisertifitseerimiseks sobiva sertifikaadi). Kõigi sel moel loodud sertifikaadide lisamine Androidi sertifikaadilattu õnnestus.

Samas tuleb märkida, et Androidi näol pole tegemist ühe kindlapiirilise platvormiga, vaid operatsioonisüsteemide perega, millest iga tootja paigaldab enda poolt kohandatud versiooni. Muuhulgas tähendab see, et krüptoteegi (milleks Androidi puhul on sageli BouncyCastle) võimekus võib erinevates seadistustes märgatavalt erineda.

3.8 Elliptkõverate realiseeritus riistvaralistes rakendustes

Selles jaotises anname ülevaate elliptkõverate toest füüsilistes turvamoodulites (*hardware security module* (HSM)) ning kiipkaartides. Ülevaate koostamiseks kontakteerusid aruande autorid mitme riistvaratarnijaga ning esitasid neile küsimustiku, mis on aruande lisas A.

nShield Connect

nShield Connect HSM toetab järgmisi kõveraid.

- NIST P-192
- NIST P-224
- NIST P-256
- NIST P-384
- NIST P-521
- NIST B-163
- NIST B-233
- NIST B-283
- NIST B-409
- NIST B-571
- NIST K-163
- NIST K-233
- NIST K-283
- NIST K-409
- NIST K-571
- ANSI B163v1
- ANSI B191v1
- SECP160r1

Lisaks loetletutele on HSM-is võimalik tarkvaras teostada ka teisi kõveraid ning kasutada neid FIPS 140-2 3. tasemel sertifitseeritud turvalises täitmiskeskkonnas (*Secure Execution Engine*). Krüptomoodulite nShield Connect programmeerimiseks on olemas eraldi API, kuid tarkvaras teostatud kõverate jõudlus jääb süsteemselt pakutavatele tõenäoliselt mingil määral alla. Vajadusel on moodulite tootjafirma Thales valmis pakkuma programmeerijatele vastavat koolitust ja arendustuge.

Krüptomoodulist nShield Connect on olemas mitu jõudlusvarianti (tähistega 500+, 1500+ ja 6000+). Nende pakutav signeerimiskiirus on tabelis 6 kõverate P-192, P-256 ja P-521 korral ning võrdluseks samuti RSA2048 ja RSA4096 jaoks.

Algoritm/kõver	500+	1500+	6000+
P-192	880	1600	2300
P-256	540	1260	2400
P-521	120	330	1300
RSA2048	150	450	3000
RSA4096	80	190	500

Tabel 6. Füüsiliste krüptomoodulite nShield Connect jõudlus signeerimisoperatsioonidena sekundi kohta

SafeNet Luna

SafeNet Luna HSM toetab järgmisi kõveraid.

- NIST P-192
- NIST P-224
- NIST P-256
- NIST P-384
- NIST P-521
- NIST B-163
- NIST B-233
- NIST B-283
- NIST B-409
- NIST B-571
- NIST K-163
- NIST K-233
- NIST K-283
- NIST K-409
- NIST K-571
- BRAINPOOL P160R1
- BRAINPOOL P160T1
- BRAINPOOL P192R1
- BRAINPOOL P192T1
- BRAINPOOL P224R1
- BRAINPOOL P224T1
- BRAINPOOL P256R1
- BRAINPOOL P256T1
- BRAINPOOL P320R1
- BRAINPOOL P320T1
- BRAINPOOL P384R1
- BRAINPOOL P384T1
- BRAINPOOL P512R1
- BRAINPOOL P512T1

SafeNet Luna HSM ei võimalda kasutajal ise uusi kõveraid teostada, sest tootja väitel rikuks see FIPS ja Common Criteria sertifitseerimise tingimusi. Täiendavate kõverate toetamine tähendab seega HSM-i uue versiooni väljaarendamist ning sertifitseerimist.

SafeNet Luna HSM-i jõudlus signeerimisel erinevate kõverate korral on tabelis 7.

Kõver	Jõudlus
P-192	3100
P-256	2000
P-384	1000
P-521	570
BRAINPOOL P512R1	620

Tabel 7. Füüsilise krüptomooduli SafeNet Luna jõudlus signeerimisoperatsioonidena sekundi kohta

Kiipkaardid

Hetkel realiseerib Eesti ID-kaardi platvorm riistvaras piiratud hulka kuni 521-bitiseid kõveraid. Muude kõverate tuge ei pakuta, sest kiibi tootjafirma Infineon pole seda senini äriliselt otstarbekaks pidanud. Et ühe kaardi hind jääks mõistlikesse piiridesse, tuleb kiipe toota massiliselt, kuid praeguseks on ainult Eesti avaldanud huvi teiste elliptkõverate vastu. Kuigi potentsiaalsed miljon kasutajat on tootja seisukohalt huvitavad, tuleb täpsemate hinnangute saamiseks tootjaga pikemalt läbi rääkida.

Aruandes [58] teostatakse Curve25519 tarkvaraliselt mõnedel levinud mikrokontrollerplatvormidel ning saavutatakse teatud konfiguratsioonis Diffie-Hellmani võtmevahetuseks vajalike operatsioonide kiirus alla sekundi. Mikrokontrolleri ja JavaCardi võimekused on siiski erinevad ning täpsemate hinnangute saamiseks tuleb eksperimente jätkata.

3.9 Kokkuvõte

Ainus lähemalt vaadeldud kõver, mis turvakriitiliste nõuete osas Bernsteini ja Lange kriteeriumitele ei vastanud, on NIST P-256, mis ei osutunud rangeks. Lihtsamalt öeldes tähendab see, et P-256 sees võib olla sinna loomise ajal paigutatud tagauks, mis võimaldab sellest teadlikul osapoolel avaliku võtme järgi salajast leida (ja seega näiteks ohvri nimel digitaalallkirju anda). Samuti tuleb P-256 ja brainpoolP256t1 korral nende teostamisel võrdlemisi hoolikas olla, et mitte tahtmatuid ebakorrektsusi või turvanõrkusi sisse viia.

Kuigi krüptograafilisest vaatepunktist on kõveral Curve25519 märkimisväärseid eeliseid, jääb ta konkurentidest kaugele maha nii standardiseerituse kui ka praktilistes teostustes toetatuse poolest. Teda pole mainitud üheski standardis ega isegi standardiorganisatsiooni infomaterjalis, samuti ei realiseeri teda levinud teegid. See tähendab, et Curve25519 lülitamine krüptograafilise ökosüsteemi osaks nõuab täieliku toe pakkumist ökosüsteemi kõigis kihtides ning see võib osutuda majanduslikult õigustamatuks. Sama kehtib ka teiste Bernsteini ja Lange aruande [41] poolt turvaliseks tunnistatud kõverate kohta.

Kokkuvõttes pole lihtne valida, milliseid kõveraid Eestis kasutusele võtta. Sisuliselt tuleb otsustada, kas eelistada standardset ja laialt toetatud, aga potentsiaalsete probleemidega (sh patendiprobleemidega) kõverat, või krüptograafilises kogukonnas populaarset kõverat, mis võib endaga kaasa tuua suuri ühilduvusprobleeme. 2014. aasta detsembris publitseeritud DigiDocService spetsifikatsioon [97] on valinud esimese alternatiivi ning deklareerinud kasutamiseks NIST-i kõvera P-256. See on pragmaatilisest seisukohast prototüüpimiseks ja katsetamiseks mõistlik valik, kuid siinkohal soovitame kindlasti ehitada tulevased infosüsteemid nii, et kasutatavad krüptograafilised algoritmid oleksid vajadusel kergesti vahe-

tatavad.

Kuigi põhimõtteliselt on võimalik elliptkõveraid teostada nii, et osad parameetrid (näiteks kõverat määrava võrrandi kordajad või baaspunkti koordinaadid) saaks ette anda süsteemi käitusajal, ei toeta ükski uuringus käsitletud teostus ega standard niisugust lähenemist. Kõik praktikas kasutatavad ja teadlaste poolt välja pakutud kõverad on alati täielikult spetsifitseeritud, sest nii on võimalik tagada süsteemide koostalitlusvõimet. Seetõttu arvame, et ka tulevikus ei võeta kasutusele standardeid, mis lubaksid kõverate parameetreid käitusaegselt ette anda.

4 Räsifunktsiooni SHA-1 kasutamisest sertifikaatides

Räsifunktsioone kasutatakse paljudes olulistes turvakriitilistes rakendustes nagu digitaalallkirjad, ajatemplid, sõnumiautentimiskoodid ja autentimine. Räsifunktsioonide ründed ja nende areng võivad seetõttu oluliselt mõjutada kõigi elektrooniliste teenuste turvalisust. Räsifunktsioon SHA-1 on rakendustes tänini kasutusel, kuid kollisioonründed SHA-1 vastu on muutumas praktiliselt teostatavateks. Seetõttu on oluline teada, milline on tänaseks teadaolevate rünnete mõju elektrooniliste teenuste turvalisusele. Eelmisel aastakümnel võisime vaadelda SHA-1 eelkäija, räsifunktsiooni MD5, järkjärgulist nõrgenemist ja kollisioonrünnete mõttes täielikult ebaturvaliseks muutumist. Seetõttu kasutatakse siin sageli SHA-1 rünnete võrdlust MD5 vastavate rünnetega.

4.1 Taust: Võlts sertifitseerimiskeskus

Aastal 2009 näitasid Stevens, Sotirov, Appelbaum, Lenstra, Molnar, Osvik ja de Weger [104], et MD5 vastaseid valikprefiksiga kollisioonründeid saab kasutada serifitseerimiskeskustelt võltssertifikaatide väljapetmiseks. Ründe olemus seisneb selles, et sertifikaadi taotleja genereerib kaks erinevat X.509 sertifikaadi sisu nii, et need annaksid samasuguse MD5 räsi, mida serifitseerimiskeskus kasutab osana oma digitaalallkirjast. Üks sertifikaatidest (võltssertifikaat) on nn. sertifitseerimiskeskuse sertifikaat, millel on õigus ise uusi sertifikaate välja anda (näiteks google.com veebisaidile) ja mida suur osa brausereid jm rakendusi pimesi usaldama hakkab. Teine sertifikaat, mille andmed esitatakse serifitseerimiskeskusele, on aga tavaline kasutaja sertifikaat ilma õigusteta uusi sertifikaate välja anda. Signeerides teise sertifikaadi, signeerib serifitseerimiskeskus tegelikult ka esimese, sest nende MD5-räsid langevad kokku. Võltssertifikaadi põhjal välja antud uusi sertifikaate võib hiljem kasutada mitmesugustes e-äriga seotud petuskeemides. Ründaja saab imiteerida mis tahes turvalist internetiteenust (näiteks gmail.com või swedbank.ee) ning varastada massiliselt paroole, lugeda suvalisi e-kirju, vms.

Et viimasel ajal on olulist edu saavutatud ka SHA-1 vastastes rünnetes (näiteks Wang jt [108, 107] ning Stevens [101]) ja on teada, et paljud sertifitseerimiskeskused kasutavad veel SHA-1 räsimist, siis on loomulik uurida võltssertifikaatidega seotud rünnete võimalikkust räsifunktsiooni SHA-1 kasutavate sertifitseerimiskeskuste vastu.

Muuhulgas tuleb ära märkida, et juursertifikaatide allkirja räsi arvutamiseks kasutatud algoritmi (nt SHA-1) võimaliku murdumise mõju juursertifikaadiga väljastatud ja selle ahelasse kuuluvate sertifikaatide usaldusväärsusele puudub. Enesesigneeritud sertifikaat ei paku mingit turvalisust, ta kujutab endast ainult mugavat konteinerit avaliku võtme jaoks. Juursertifikaadis sisalduva avaliku võtme autentsus ja terviklus tuleb tagada mittekrüptograafiliste meetmetega (füüsiliselt, organisatsiooniliselt vms).

4.2 SHA-1 struktuur

SHA-1 on Merkle'i ja Damgårdi meetodil koostatud räsifunktsioon. See tähendab, et teatud pikkuseni täidistatud sisendandmed M tükeldatakse 512-bitisteks plokkideks $M_1, \ldots, M_\ell$, mida töödeldakse iteratiivselt, võttes $h_0 = \mathrm{IV}$, kus IV on (SHA-1 standardiga) fikseeritud 160-bitine algväärtus ja

$$h_i = F(h_{i-1}, M_i)$$

iga $i = 1 \ldots \ell$ korral. Andmete M räsiks $H(M)$ võetakse h_ℓ. Funktsiooni F nimetatakse räsifunktsiooni tihendusfunktsisooniks (*compression function*). Funktsiooni F sisendandmed on 160-bitine olekuräsi V ja 512-bitine sõnumiplokk M. Väljundjada on 160-bitine uus olekuräsi.

Sarnase konstruktsiooniga on ka SHA-1 eellased MD4 ja MD5 (kus olekuräsi ja väljundjada on mitte 160- vaid 128-bitine). Funktsiooni SHA-1 tihendusfunktsioon koosneb 80-st arvutustsüklist, milles tehetena kasutatakse loogika ja moodularitmeetika operatsioone.

4.3 Räsirünnete tüübid

Räsifuktsioonide rakendustes eeldatakse räsifunktsioonidelt turvaomadusi nagu ühesuunalisus, kollisioonikindlus, vms. Paljudes formaalsetes turvatõestustes modelleeritakse räsifunktsioone juhuslike funktsioonidena (juhusliku oraakli mudel, *random oracle model*). Seetõttu mõeldakse räsifunktsioonide rünnetena (nn. räsirünnetena) mis tahes ründeid, kus originaali (avateksti) ja kujutise (räsi) paaridele leitakse uusi originaale ja kujutisi viisil, mis on juhusliku oraakli mudelis arvutuslikult teostamatu. Kolm põhilist (klassikalist) räsifunktsioonidelt nõutavat omadust on järgmised.

- *Ühesuunalisus* (*one-wayness*) tähendab üldjoontes seda, et funktsiooni on võimalik arvutada vaid niiöelda õiges suunas, st sisendandmete M järgi on võimalik arvutada räsi $h = H(M)$, kuid teades ainult räsi h, ei ole võimalik leida vastavat originaali M.
- *Lisaoriginaalikindlus* (*second preimage resistance*) tähendab seda, et etteantud (juhuslikult valitud) originaalile M lisaoriginaali $M' \neq M$ leidmine (st $H(M') = H(M)$) on arvutuslikult teostamatu.
- *Kollisioonikindlus* (*collision-resistance*) tähendab seda, et kahe erineva originaali M ja M' leidmine, nii et $H(M) = H(M')$, on arvutuslikult teostamatu.

Väljendusviisi lihtsustamiseks ütleme, et räsifunktsioon on mingi ründetüübi suhtes S-turvaline, kui tõhusaim teadaolev rünne kasutab aega, mis on võrreldav S erineva originaali räsimisega. Seetõttu ei saa ükski n-bitise väljundiga räsifunktsioon olla rohkem kui

- 2^n-turvaline ühesuunaline või lisaoriginaalikindel, sest ründaja saab keskmiselt 2^{n-1} juhusliku sisendväärtuse räsimisega leida mis tahes väljundjadale (räsile) vastava originaali, ega
- $2^{\frac{n}{2}}$-turvaline kollisioonikindel, sest $2^{\frac{n}{2}}$ juhuslikult valitud originaali seas on suure tõenäosusega kaks ühesuguse räsiga.

Seega ei ole SHA-1 rohkem kui 2^{160}-turvaline ühesuunaline/lisaoriginaalikindel ja rohkem kui 2^{80}-turvaline kollisioonikindel. Räsifunktsioon MD5 aga ei saa olla enam kui 2^{128}-turvaline ühesuunaline/lisaoriginaalikindel ja rohkem kui 2^{64}-turvaline kollisioonikindel.

Käesolevas töös vaadeldakse ainult kollisioonründe, sest vaid selles vallas on teada praktiliselt teostatavaid ründe levinumate räsifunktsioonide (MD4, MD5, SHA-0, SHA-1, SHA-256, jt) või nende kärbitud (vähendatud töötsüklite arvuga) versioonide vastu.

Kollisioonründed räsifunktsioonile tervikuna

Kollisioonrünne (*collision attack*) Leia kaks erinevat sõnumit $M \neq M'$ nii et $H(M) = H(M')$.

Räsifunktsiooni SHA-1 täisversiooni vastu ei ole tehtud ühtegi edukat kollisioonrünnet. Esimese teoreetilise (2^{69} räsimist nõudva) kollisioonründe esitasid Wang, Yin ja Yu aastal 2005 [108]. Esimese eduka kollisioonründe MD5 täisversioonile teostasid Wang, Feng, Lai, ja Yu aastal 2004 [106].

Aastal 2010 esitas Grechnikov 2^{35} räsimist nõudva ründe SHA-1 kärbitud 74-tsüklilisele versioonile [60].

Valikprefiksiga kollisioonrünne (*chosen-prefix collision attack*) Etteantud sõnumite M_1 ja M_2 puhul leida sõnumid M_1' ja M_2', nii et $M_1||M_1' \neq M_2||M_2'$ ja $H(M_1||M_1') = H(M_2||M_2')$.

Räsifunktsiooni SHA-1 vastu ei ole teostatud ühtegi edukat valikprefiksiga kollisioonrünnet. Esimene teoreetilise (2^{74} räsimist nõudva) valikprefiksiga kollisioonründe avaldas Stevens aastal 2013 [101]. Esimese eduka valikprefiksiga (2^{39} räsimist nõudva) kollisioonründe MD5 vastu sooritasid Stevens, Lenstra ja de Weger aastal 2007 [102, 103].

Kui üldine kollisioonrünne ei tarvitse olla praktiliselt ohtlik, sest M ja M' ei tarvitse olla tähendusega sõnumid, siis valikprefiksiga kollisioonrünnet (räsifunktsiooni MD5 vastu) on praktikas kasutatud näiteks sertifitseerimiskeskuse sertifikaatide võltsimiseks [104].

Kollisioonründed tihendusfunktsioonile

Pseudokollisioonrünne (*pseudo-collision attack*) Leida algväärtuste ja sõnumite paarid $(V, M) \neq (V', M')$ nii et $F(V, M) = F(V', M')$. [12]

Räsifunktsiooni SHA-1 vastu ei ole tehtud ühtegi edukat pseudokollisioonrünnet. Esimese eduka pseudokollisioonründe MD5 vastu avaldasid den Boer ja Bosselaers aastal 1993 [54].

Tihendusfunktsiooni kollisioonrünne Leida $M \neq M'$ ja V nii et $F(V, M) = F(V, M')$.

Räsifunktsiooni SHA-1 tihendusfunktsiooni vastu ei ole tehtud ühtegi edukat kollisioonrünnet. Esimese eduka kollisioonründe MD5 tihendusfunktsiooni vastu sooritas Dobbertin aastal 1996 [57].

Tihendusfunktsiooni kollisioonrünne ei tähenda veel kollisioonrünnet räsifunktsioonile tervikuna, sest V ei tarvitse olla õige (standardis fikseeritud) algväärtus IV. Kui $V \neq \mathrm{IV}$, siis tihendusfunktsiooni kollisiooni laiendamiseks räsifunktsiooni kollisiooniks peab olema teada prefiks P, nii et $H(P) = V$. Sel juhul tõepoolest $H(P||M) = H(P||M')$.

[12] Tihendusfunktsiooni ründeid, kus esimesed argumendid V ja V' on vabalt valitavad, nimetatakse vahel ka *vaba algväärtusega* (*free-start collision attacks*), ja kui $V = V'$, siis *poolvaba algväärtusega* (*semi-free start*) rünneteks, et eristada neid rünnetest, kus V on enne rünnet fikseeritud.

Kui aga leitakse efektiivne rünne, mis *juhuslikult valitud* olekuräsile V võimaldab (piisavalt suure tõenäosusega) leida $M \neq M'$, nii et $F(V, M) = F(V, M')$, siis on võimalik leida ka kollisioone räsifunktsioonile tervikuna. Selleks tuleb valida juhuslik 512-bitine prefiks P, võtta $V = F(\mathrm{IV}, P)$ ja kasutada ülalmainitud rünnet leidmaks M ja M', sest siis $H(P||M) = H(P||M')$.

Lähikollisioonrünne (*near-collision attack*) Etteantud IV ja IV′ korral leida $M \neq M'$, nii et $F(\mathrm{IV}', M')$ ja $F(\mathrm{IV}, M)$ erinevad vaid vähese arvu bittide poolest või on mingil muul eelnevalt kokkulepitud viisil lähedased.

Lähikollisioonrünnetel ei ole iseseisvat praktilist tähtsust, kuid neid kasutatakse komponendina räsifiktsioonide tervikversioonide rünnetes, nagu näiteks Stevensi 2013. aastal avaldatud kollisioonründes SHA-1 vastu [101].

4.4 SHA-1 rünnete ajalugu

Aastal 2005 avaldasid Rijmen ja Oswald kollisioonründe 53-tsükliliseks kärbitud SHA-1 vastu [94], mis nõuab vähem kui 2^{80} räsimist, ning samal aastal Wang, Yin ja Yu avaldasid 2^{69} räsimist nõudva kollisioooniründe SHA-1 täisversiooni vastu [108]. Hiljem parandasid Wang, Yao ja Yao tulemust 2^{63} räsimiseni [107].

Aastal 2012 avaldas Stevens SHA-1 kollisioonründe 2^{60} räsimisega ja valitava prefiksiga kollisioonründe 2^{77} räsimisega [101]. Need ründed on tänaseni parimad teadaolevad kollisioonründed SHA-1 vastu.

4.5 SHA-1 valikprefiksiga kollisioonründe maksumus

Jaotises 2.4 hindasime SHA-1 kollisiooni leidmiseks vajalikku rahalist investeeringut umbes 67,52 miljoni dollari suuruseks. Selle eest suudab ründaja teha 2^{60} räsiarvutust (vt ka [17]).

Parim teadaolev valikprefiksiga kollisioonrünne SHA-1 vastu kasutab umbes 2^{77} räsimist [101]. Seega on ka kulu üldise kollisioonründega võrreldes 2^{17} korda suurem ja on seega umbes

$$\$67,52 \cdot 10^6 \cdot 2^{17} \approx \$8,85 \cdot 10^{12} = \$8\,850\,000\,000\,000\ .$$

Need hinnangud on mõistagi ligikaudsed ja põhinevad laiatarbearvutite ja -teenuste hindadel. Ei saa välistada, et erilahendused (näiteks massiline FPGA-de kasutus) võimaldavad hinda oluliselt alandada. Seega ei saa välistada, et valitava prefiksiga kollisioonrünne, ja seega ka SHA-1 kasutavate sertifitseerimiskeskuste sertifikaatide võltsimine on juba täna jõukohane suurematele organisatsioonidele ja jõustruktuuridele.

Kui võrrelda vastavaid ründeid MD5 ja SHA-1 vastu, siis MD5 kollisioone saab tavalist koduarvutit kasutades leida sekunditega ja valitava prefiksiga kollisioonide leidmine võtab umbes 2^{39} räsimist, st on ligikaudu $2^{36} \approx 7 \cdot 10^{10}$ (70 miljardit) korda odavam vastavast SHA-1 ründest, st on arvutuste järgi jõukohane eraisikuile (suurusjärgus $\$100$).

4.6 Räsiründed olemasolevate sertifikaatide vastu

Valikprefiksiga kollisioonrünne on ainus teadaolev praktiliselt teostatav rünne SHA-1 algoritmiga räsitud sertifikaatide vastu ja sarnast rünnet MD5 algoritmiga räsitud sertifikaatide

Tabel 8. Räsirünnete praktilised järeldused.

	Lisaoriginaalirünne	Valikprefiksiga kollisioonrünne	Kollisioonrünne
Anonüümselt teostatav	jah	ei	ei
Teesklus	jah	jah, võltssertifikaadiga	ei
Võltsallkiri	jah	jah	jah, sisuta dokumendile
Korduv võltsallkiri	jah, võltssertifikaadiga	jah, võltssertifikaadiga	ei
Võltsajatemplid	jah	jah, võltssertifikaadiga ainult RFC3161 templid	ei
Võltsarhivaal	jah, kui ei ole enne räsifunktsiooni ebaturvaliseks muutumist turvaliselt üle ajatembeldatud	ei	ei
Praktiline teostatavus MD5 vastu	Praktikas teostamatu nõuab 2^{123} räsimist	Jõukohane eraisikule nõuab 2^{39} räsimist	Lihtne tavaarvutis sekundiga
Praktiline teostatavus SHA-1 vastu	Praktikas teostamatu nõuab 2^{160} räsimist	Jõukohane suurtele organisatsioonidele	Jõukohane suurtele organisatsioonidele

vastu on ka praktikas edukalt katsetatud. See rünne on aga sooritatav vaid sertifikaadi taotlemisprotsessi ajal, mitte aga juba olemasolevate sertifikaatide vastu. Valikprefiksiga kollisioonründe abil saavad võltssertifikaate luua ainult isikud, kes on volitatud sertifikaate taotlema. Selle kindlakstegemiseks tuleb taotleja isik tuvastada, mistõttu on võltssertifikaadi avastamisel ründajat lihtne leida.

Esmapilgul võivad palju ohtlikumana tunduda olemasolevate sertifikaatide vastu suunatud ründed, mida saab sooritada ilma uue sertifikaadi taotlemise ja isiku tuvastamiseta. Näiteks võib ründaja üritada juba eksisteeriva sertifikaadi X järgi leida uue sisuga võltssertifikaati $X' \neq X$ nii, et nende SHA-1 räsid kokku langevad, st $H(X') = H(X)$. Kui X oleks näiteks Eesti ID-kaartidele sertifikaate väljastava sertifitseerimiskeskuse või kehtivuskinnitusi välja andva OCSP serveri sertifikaat, siis saaks teatud tingimustel luua mis tahes isikute näiliselt kehtivaid digitaalallkirju.

See rünne aga ei oleks enam kollisioonrünne, vaid rünne lisaoriginaalikindluse vastu, sest X on enne rünnet fikseeritud. Ühtegi praktilist lisaoriginaalirünnet SHA-1 vastu aga ei teata. Isegi räsifunktsiooni MD5 teadaolevad lisaoriginaaliründed nõuavad ligi 2^{123} räsimist, mis on 2^{46} ($\approx 10^{14}$) korda enam kui nõuavad valikprefiksiga kollisioonründed SHA-1 vastu.

Tabel 8 võtab kokku räsirünnete praktilised järeldused elektroonilistele teenustele. Lisaoriginaaliründel on enim praktilisi järeldusi, kuid see rünne ei ole tänini praktiliselt teostatav ei SHA-1 ega MD5 vastu. Üldine kollisioonrünne võimaldab küll võltsida üksikuid digitaalallkirju, kuid vaid juhusliku struktuuriga ja tähenduseta dokumentidele. Vaid valikprefiksiga kollisioonründed võimaldavad ise valida võltsitud dokumendi sisu ja võltssertifikaadi ründe kaudu teostada ka muid praktilisi ründeid. Neist sisuline praktiline tähtsus on võltssertifikaadiga seonduvatel rünnetel, mida saab teostada sertifikaadi taotleja isikut otseselt tuvastamata. Näiteks veebiserverite sertifikaadid on sobilik ründeobjekt. Seega on olulisim praktiline järeldus mitte enam kasutada SHA-1 veebiserverite sertifikaatide väljaandmisel.

5 Elektroonilise identiteedi protokollid ja laiendused

Eestis (aga ka mitmel pool mujal maailmas) on välja antud hulk kiipkaarte, mis pakuvad tänu krüptograafiliste operatsioonide võimele mugavaid vahendeid isiku tugevaks autentimiseks ning digitaalsignatuuride moodustamiseks. Sel ajal, kui niisuguste kaartide väljaandmisega algust tehti (Eestis 2000ndate aastate alguses) olid valdavateks arvutusseadmeteks laua- ja sülearvutid, millele vajalike kaardilugejate lisamine või integreerimine polnud suur probleem.

2015. aastaks on valdavateks arvutiplatvormideks muutunud mobiilseadmed. Reeglina puuduvad neil integreeritud kaardilugejad ning väliste lugejate lisamine muudaks mobiilseadme kasutaja jaoks kohmakaks.

Selle probleemi ühe võimaliku lahendusena on Eestis kasutusele võetud mobiil-ID, kuid ka see seab omad tehnilised piirangud. Näiteks eeldab mobiil-ID eraldi SIM-kaardi väljaandmist, sellal kui mobiilplatvormide tootjad on hakanud töötama välja seadmeid, millel SIM-kaardi tugi üldse puudub. On selge, et turvalise suhtluskanali loomiseks vajalikke krüptograafilisi võtmeid tuleb kusagil hoida ning selle kanali loomine ja kasutamine peavad toimuma mingi standardse protokolli alusel. Tellija defineeritud protokollide valiku kirjeldamine ongi selle jaotise üks eesmärk (vt jaotised 5.4 ja 5.5).

Selle eesmärgiga on tihedalt seotud ka tuletatud identifitseerimisvahendite temaatika. Kui primaarne identifitseerimisvahend on igapäevaseks kasutamiseks liiga ebamugav, võib osutuda otstarbekaks anda selle alusel välja mõni lihtsamini kasutatav pääsmik. Võimalikud näited soovitas Tellija. Niisugune lähenemine ei lahenda aga veel võtmehalduse probleemi (vt jaotised 5.1 ja 5.2).

Sekundaarsed elektroonilised isikutunnistused ning füüsiliste isikutunnistuste elektroonilised laiendused on juba leidnud tee igapäevakasutusse ning Tellija konkreetsetest soovitustest lähtudes teeme ülevaate ka mõnedest rakendustest (vt jaotised 5.3 ja 5.6).

5.1 NIST SP 800-157

NIST üllitas 2014. aasta detsembris soovitusliku dokumendi SP 800-157 [63], mis kirjeldab võimalusi, kuidas olemasoleva isikutuvastuskaardi (*Personal Identity Verification Card, PIV Card*) alusel tuletistõendeid (*derived credentials*) välja anda.

Saavutamaks koostalitlusvõimet olemasoleva isikutuvastustaristuga, tuginevad ka SP 800-157 alusel välja antud tuletistõendid avaliku võtme krüptograafiale. Tõendi saamiseks peab taotleja tõestama juurdepääsu esmasele isikutuvastuskaardile. Seda saab teha nii kindlustasemel LOA-3 (*Level of Assurance*), mille puhul piisab isiku digitaalsest tõendamisest, kui

ka tasemel LOA-4, mis eeldab isiku otsest füüsilist tuvastamist.

Mõlemal juhul omandab taotleja uue avaliku ja privaatvõtme paari ning vastava avaliku võtme sertifikaadi. Privaatvõtme hoidmiseks näeb SP 800-157 ette mitu võimalust.

- **Krüptopääsmikud** (*Removable Hardware Cryptographic Token*)
  - ***Krüptomooduliga SD-kaart***
  - ***Krüptomooduliga UICC-kaart*** UICC-kaart (*Universal Integrated Circuit Card*) on traditsiooniliste SIM-kaartide edasiarendus, mille puhul on lisatud erinevate turvaalade, täiendavate rakenduste jms tugi. Selle kaarditüübi on standardinud GlobalPlatform oma üllitistesarjas [67].
  - ***Krüptomooduliga USB-pääsmik*** Kuigi USB-seadmed võivad olla väga võimsad, jääb nende soovitamine SP 800-157 kontekstis segaseks, sest dokumendi algseks motivatsiooniks oli soov vabaneda välisseadmete kasutamisest.
- **Sisseehitatud krüptograafilised vahendid** (*Embedded Cryptographic Tokens*)
  - ***Riistvaralised vahendid*** on üllitises SP 800-157 kahjuks käsitletud vaid väga pinnapealselt, kuigi just nendest võib tulevikus kujuneda peamine krüptotõendite hoidmise koht. Mobiilplatvormide tootjad on lisanud oma seadmetele usaldatava täitmiskeskkonna (*Trusted Execution Environment, TEE*), mis võimaldab talletada nii rakendusi kui ka krüptovõtmeid ja pakub tõhusamat kaitset nt külgrünnete vastu. TEE standardite väljatöötamisega tegeleb samuti GlobalPlatform [5]. Ka Eesti kontekstis tasub edaspidi usaldatud täitmiskeskkondade ja nende võimaluste uurimisele rohkem tähelepanu pöörata.
  - ***Tarkvaralised vahendid*** olid pikemalt kirjeldatud SP 800-157 mustandis [64], kuid dokumendi lõppversioonist on see osa välja jäetud. Üheks võimalikuks põhjuseks on kriitika, mida mustandversioonis sätestatud tarkvaralised võtmehalduslahendused esile kutsusid (vt nt [49]). Krüptograafiliste võtmete kaitsmine pelgalt tarkvaraliste vahenditega ongi väga raske ülesanne, mida teema mahavaikimine kõnealustes soovitustes loomulikult lahendada ei aita.

Niisiis käsitleb dokument SP 800-157 küll olulist teemat, kuid jääb lahenduste pakkumisega poolele teele. Küsimusele, kuidas siis tuletatud identiditseerimisvahendi jaoks moodustatud privaatvõtmeid hoida, pole ammendavalt vastatud. Kuna kõnealuste soovituste motivatsiooniks oli vabaneda kaardilugejast, pole väliste riistvaraliste krüptopääsmike kasutamine lahendus. Mobiilseadmetesse sisse ehitatud vahendid (nt TEE) on alles arendusjärgus ning nende kasutamise tingimused ja vajalikud turvaeeldused pole veel selged. Küll aga on selge, et privaatvõtmeid ei saa hoida ainult tarkvaraliselt nõrga kaitsega mälupiirkondades.

Kokkuvõtteks võime öelda, et täna ei paku mobiilplatvormid veel piisavalt levinud vahendeid, mis võimaldaksid ID-kaardi või mobiil-ID taolistes lahendustes loobuda privaatvõtme hoidmisest kiibis, kuid nende vahendite juurutamine laiatarberakendustes on paari lähema aasta küsimus.

5.2 Pomcor

Eelmise jaotise kokkuvõtteks tõdeme, et tehnoloogia praegust arengutaset arvestades on privaatvõtmete kaitsmisel mõistlik kasutada riistvaralisi vahendeid.

Aga ka riistvaraline kaitse (nt võtmete ja rakenduste hoidmine TEE-s) ei lahenda kõiki probleeme automaatselt, sest pakutava teenuse lõppkasutaja on inimene, kelle füüsiline võime ennast seadmele võtmete volitatud kasutajana autentida on piiratud. Seega tuleb iga lahenduse väljatöötamisel arvestada võimalusega, et ründaja võib näiteks lühikese PIN-koodi ära arvata proovimise teel või et seadmesse paigaldatud kahjurvara kuulab PIN-koodi sisetamisel pealt.

Neid probleeme püüab lahendada firma Pomcor väljatöötatud lähenemine, mis kasutab TEE pakutava usaldatava kasutajaliidese võimalusi [50]. Kaitstavat privaatvõtit hoitakse TEE-s krüpteeritult, dekrüpteerimiseks vajalikku võtit aga talletatakse omakorda krüpteeritult eraldi välises serveris. Serveris hoitavale võtmele pääseb kasutaja juurde TEE usaldatava liidese kaudu, esitades oma PIN-koodi, biomeetrilise lugemi ja/või mobiilseadmes talletatud eelarvutatud väärtuse (*protocredential*). Süsteemi loojate väitel aitab selline arhitektuur võidelda seadmesse paigaldatud kahjurvara ning PIN-koodi äraarvamise vastu. Siiski jääb selgusetuks, miks on tarvis hoida krüpteeritud võtit välises serveris, kui kogu pääsufunktsiooni teostab TEE koos oma usaldatava kasutajaliidesega. Seetõttu on enne niisuguse lahenduse kasutuselvõttu kindlasti vaja täiendavat analüüsi, samuti peab Pomcor oma toodet veel rahvusvahelise kogukonna silmis tõestama.

5.3 SEB töötõend

Eestis on tuletatud identifitseerimisvahend leidnud laiemat rakendust SEB panga töötõendina [92]. See on kiibiga varustatud plastikkaart, mida saab kasutada nii visuaalseks isikutuvastuseks, uksekaardina kui ka tööjaama meldimise vahendina. Tehnoloogiliselt platvormilt on kaart identne Eesti ID-kaardiga, kasutatakse sama kiipi, operatsioonisüsteemi ja kaardil olevat rakendust. Sarnaselt ID-kaardile on kiip varustatud kahe võtmepaariga, millest ühega saab anda digitaalallkirja ning teisega autentida e-teenustesse.

Vastavad sertifikaadid kuuluvad AS Sertifitseerimiskeskuse EID-sertifikaadipuusse ja kaardi abil antud digitaalsignatuur on kehtiv digitaalallkirja seaduse mõttes. SEB töötõendi kasutatavus e-teenuste autentimisvahendina sõltub sellest, kas vastava teenuse pakkuja on otsustanud EID-harru kuuluvaid sertifikaate tunnistada (ID-kaartide sertifikaadid kuuluvad ESTEID-harru). Ainus ID-kaardi funktsioon, mida SEB töötõend põhimõtteliselt ei toeta, on Interneti-hääletamine, sest hääletamise jaoks on tarvis riiklikult väljaantud isikuttõendavat dokumenti[13], milleks SEB töötõend ei kvalifitseeru. Muus osas on see ID-kaardiga praktiliselt samaväärne, näiteks saab tema haldamiseks kasutada tavalisi ID-kaardi haldusvahendeid.

Töötõendi annab välja vastavalt SK poliitikale SEB personaliosakond, kus eelnevalt tuvastatakse inimene füüsiliselt ja kehtiva isikut tõendava dokumendi alusel. Töötõendil olevatele salajastele võtmetele vastavaid sertifikaate saab aktiveerida ainult kasutaja ise ainult riiklikult väljastatud ID-kaardiga, kasutades selleks AS Sertifitseerimiskeskuse pakutavat standardset teenust[14]. Sertifikaatide peatamise ja tühistamise õigus on ka SEB-l.

SEB töötõend on hea näide sellest, kuidas uue lahenduse sisseviimisel kasutatakse maksimaalselt ära senist taristut ning ennast praktikas tõestanud vahendeid. Kõik vajalikud komponendid on Eestis olemas, mistõttu võime soovitada sarnase lahenduse kasutamist ka teistes ettevõtetes.

[13] Vt Riigikogu valimise seadus, RT I 2002, 57, 355, §48^4 (2).

[14] `https://www.sk.ee/aktiveerimine/`

5.4 NFC-digi-ID

Ka Eesti eID-platvormi pakkuja Trüb Baltic AS on välja töötamas oma lahendust kontaktita digi-ID kasutamiseks NFC-ühenduse kaudu. Aruande koostamise ajal oli vastav arendus alles algusjärgus ning selle kohta polnud veel piisavalt täpset dokumentatsiooni. Siinkohal soovitame tulevasele NFC-digi-ID protokollile enne kasutuselevõttu põhjalikku krüptograafilist analüüsi.

5.5 OPACITY

Open Protocol for Access Control, Identification, and Ticketing with privacY (OPACITY) [76, 77] on protokollide komplekt kiipkaartide ja terminalide vaheliseks autentimiseks ja võtmevahetuseks, muuhulgas üle NFC-ühenduse. Protokollides on kaks osapoolt — kaart ja terminal — ning protokollid on lihtsa struktuuriga, koosnedes kõigest kahest sõnumist. Tegemist on autenditud Diffie-Hellmani tüüpi protokollidega, kus seansivõtme arvutamisel kombineeritakse käesoleva seansi jaoks genereeritud Diffie-Hellmani parameetrid poolte pikaealiste võtmetega, mis on sertifitseeritud. Kui mõlemad pooled arvutavad välja sama võtme, usuvad nad, et teine pool oli see, kes ta väitis end olevat ja kelle sertifikaadi ta esitas.

Tuletame meelde, et Diffie-Hellmani parameetrite isend koosneb salajasest ja avalikust väärtusest, kus salajane väärtus on võrdne avaliku väärtuse diskreetse logaritmiga kasutusel olevas rühmas. Ühe isendi genereerimine on arvutuslikult sama keerukas kui astendamine selles rühmas. OPACITY protokollis on ka pikaealisteks võtmeteks Diffie-Hellmani parameetrite isendid. Standard [76] spetsifitseerib mitu võimalikku rühma, mida kasutada erinevate turvatasemete saavutamiseks. Nad kõik on elliptkõverate rühmad.

OPACITY protokollistik koosneb kahest protokollist — OPACITY-FS (*full secrecy*) ja OPACITY-ZKM (*zero key management*). Neist esimeses on nii kaardil kui ka terminalil sertifitseeritud avalik võti, millele vastavat salajast võtit ta ainult ise teab. OPACITY-ZKM protokollis on sertifikaat ainult kaardil. Seega OPACITY-ZKM e võimalda juba ainuüksi oma disaini tõttu terminali autentimist.

OPACITY protokollides tuleb nii kaardil kui ka terminalil luua üks Diffie-Hellmani parameetrite isend ja teha kaks astendamist kasutusel olevas rühmas. Korduvate autentimiste hõlbustamiseks pakuvad mõlemad protokollid *püsisidumise* (*persistent binding*) võimalust, kus üks pool salvestab teise poole nime ja mingi seemne, millest järgmise seansi võti genereerida. Kui järgmisel seansil mõlemad pooled leiavad, et neil on olemas teise poole nimele vastav seeme, siis saab seda kasutada ning rühmaoperatsioone on tarvis tunduvalt vähem (kaart ei pea neid üldse tegema).

OPACITY üheks oluliseks turvaeesmärgiks on lisaks kokkulepitud võtme salajasusele ja osapoolte autentimisele veel tulevikuturvalisus. See tähendab, et pärast seansivõtmete ja nende võtmete loomise aluseks olnud Diffie-Hellmani parameetrite kustutamist ei tohiks need enam seansi transkriptist ja osapoolte pikaealistest salajastest võtmetest taastatavad olla. Samuti taotletakse kaartide privaatsust — pealtkuulaja ei tohiks teada saada, kas kaks eri seanssi sama terminali juures on toimunud sama kaardiga või erinevate kaartidega.

Mõlema OPACITY protokollistikku kuuluva protokolli põhjalik turvaanalüüs on läbi viidud artiklis [53]. Protokollid on piisavalt lihtsad, nii et neid on võimalik täielikult analüüsida isegi

tõestusassistente või automaatseid analüsaatoreid kasutamata. Artiklis [53] antakse osa eelpool nimetatud omaduste jaoks tõestus, et protokoll seda rahuldab. Tõestus antakse mudelis, kus on palju terminale ja palju kaarte, mille vahel ründaja võib uusi seansse algatada, ja kus osa kaarte on ründaja kontrolli all. Selle mudeli terminites formaliseeritakse tõestatav omadus ning tõestus esitatakse krüptograafiliste mängude jadana [36].

Analüüsist selgub, et OPACITY-FS tagab kokkulepitud võtme salajasuse ning autendib kaardi terminalile. Selgub ka, et protokolli disaini tõttu ei võimalda ka OPACITY-FS terminali autentimist kaardile, sest protokoll koosneb ainult kahest sõnumist, millest esimese saadab terminal. Seega pole kaardil võimalust kontrollida, kas see sõnum pole taasesitusründe osa. Võib-olla osutuks terminal autendituks pärast seda, kui neile kahele sõnumile järgneks kokkulepitud võtme kasutamine infovahetuseks ja terminal demonstreeriks, et ta seda võtit teab — analüüsis [53] seda küsimust ei uurita. Ka on OPACITY-FS protokoll tulevikuturvaline juhul, kui lekib ühe osapoole pikaealine salajane võti ja hetkeolek, mis sisaldab püsisidumiseks vajalikke andmeid.

Analüüsist selgub ka, et OPACITY-FS takistab eri protokolliseansside sidumist seal kasutatud kaartidega ainult juhul, kui püsisidumist ei kasutata. Püsisidumise kasutamisel on ründajal võimalik terminali ja kaardi püsisidumise jaoks salvestatud andmete sünkroonsust rikkuda. Seejärel on tema jaoks tuvastatav, kas järjekordsel seansil on terminali ja kaardi vastavad andmed sünkroonis või mitte. Küll aga tagab OPACITY-FS nõrgema privaatsusomaduse — kaardi identiteet pole protokolliseanssidest välja loetav.

OPACITY-ZKM on märksa nõrgemate turvaomadustega. Analüüsis [53] pannakse kõigepealt tähele, et ta ei ole lõpuni spetsifitseeritud selles osas, mis puudutab kaardi sertifikaadi kontrollimist terminalis. Analüüs kirjeldab, kuidas protokolli spetsifitseerida nii, et võtmete salajasus (seanssi pealtkuulava ründaja vastu) ja kaartide autentimine tagatud oleks. Kui OPACITY-ZKM protokolli kusagil kasutatakse, tuleb kindlasti täpsustada, kuidas sertifikaadikontroll toimub.

OPACITY-ZKM ei taga kaartide privaatsust samasugusel viisil nagu OPACITY-FS — kaardi identiteet lekib igal juhul. Samuti pole OPACITY-ZKM tulevikuturvaline.

Lisaks analüüsile [53] uuritakse OPACITY protokolli ka ühes pisut varasemas magistritöös [69]. Sertifikaatide tühistusmehhanismi puudumine näib olevat ainsaks huvitavaks analüütiliseks tähelepanekuks OPACITY puudujääkide kohta selles magistritöös.

Käesoleva aruande koostajad ei anna siinkohal ühest soovitust OPACITY protokollide kasutamiseks või mittekasutamiseks. Võib arvata, et analüüsis [53] täpsustatud protokolliomadused pole päris samad, mida spetsifikatsiooni [76] autorid silmas pidasid. Samas võivad erinevate rakenduste jaoks tarvilikud jõudlus- ja turvanõuded (tulevikuturvalisus, anonüümsus) olla sellised, et mõni OPACITY protokollidest sobib nendega. Küll aga soovitame OPACITY-FS protokolli kasutades modelleerida ka võtmevahetusele järgnevat infovahetust kokkulepitud võtmega ja analüüsida, kas selle käigus saab terminal kaardile autenditud.

5.6 Biomeetrilised reisidokumendid

Masinloetava reisidokumendi (passi) omaniku nimi ja teised isikuandmed on selles passis lisaks tavapärasele kujule ka optilise tekstituvastuse (OCR) kujul, mida masinad lugeda suudavad. Biomeetriline reisidokument (e-pass) sisaldab ka kiipi, millel on salvestatud

passiomaniku isikuandmed [11]. Tagamaks kiibil olevate andmete ja kiibi enda terviklust ja privaatsust, nõuab standard [11] järgmisi meetmeid.

Passiivne autentimine tähendab selle standardi kontekstis kiibil olevate andmete signeerimist e-passi väljaandja avaliku võtmega. E-passi lugejale peavad olema kättesaadavad selle võtme terviklust kinnitavad sertifikaadid.

Aktiivne autentimine tähendab selle standardi kontekstis pretensiooni ja vastusega (*challenge-response*) protseduuri kasutamist, mille abil pass tõestab lugejale, et ta teab mingit salajast võtit, millele vastava avaliku võtme sertifikaat on lugejale teada. Aktiivne autentimine takistab passi kloonimist.

Pääsu reguleerimine tähendab selle standardi kontekstis passi ja lugeja vahel võtme kokkuleppimist, millega krüpteeritakse edasine liiklus kiibil olevate tundlike andmete lugemisel. Enne lugemist peab lugeja passile tõestama, et ta näeb sinna kantud optiliseks tekstituvastuseks sobivaid andmeid. Selle meetme abil takistatakse kiibil olevate andmete kauglugemist, st lugemist ilma passi avamata.

Aktiivse autentimise ja pääsu reguleerimise kasutamine ei ole kohustuslik [11].

Pääsu reguleerimisel nõuab [11] ISO 11770-2 võtmevahetusprotokolli nr. 6 [95] kasutamist (standard [11] kasutab siinkohal nimetust *Basic Access Control*, BAC, vaata ka [16]). See on sümmeetrilist krüptograafiat kasutav protokoll, kus kaks poolt kasutavad eelnevalt kokkulepitud pikaealist sümmeetrilist võtit selleks, et kokku leppida (lühiealises) seansivõtmes. Standard nõuab, et passi ja lugeja vaheline pikaealine võti tuletatakse passi numbrist, kehtivuse lõpukuupäevast ja omaniku sünnipäevast. Seega ei sisalda see võti väga palju entroopiat ning on oht, et ta arvatakse ära ka passi avamata, näiteks eelnevalt pealtkuulatud pääsuseanssi analüüsides.

Lisaks BAC-ile nimetab standard [11] veel *laiendatud pääsureguleerimist* (*Extended Access Control*, EAC), jättes vastava protokolli spetsifitseerimise küll passi väljaandva riigi hooleks. Pärast EAC-i kasutamist annab pass juurdepääsu kiibil salvestatud tundlikumatele andmetele (näiteks sõrmejälje ja silmaiirise kujutised). EAC-i kasutamiseks peavad terminalil (mille küljes on kiibi lugeja) olema kiibi poolt kontrollitavad tõendid, et tal on õigus tundlikele andmetele juurde pääseda.

PACE

PACE (*Password Authenticated Connection Establishment*) on BSI pakutud [28] ja ICAO (*International Civil Aviation Organization*) kinnitatud [13] võtmevahetusprotokoll, mis on mõeldud BAC-d asendama ja selle puudusi parandama. See on *parooliga autenditud võtmevahetusprotokoll*, s.t. protokoll, kus kokkulepitava võtmematerjali autentsus (kuid mitte salajasus) tagatakse parooliga, mis ei pruugi väga palju entroopiat sisaldada. Võti ise lepitakse tüüpiliselt kokku mingi Diffie-Hellmani tüüpi konstruktsiooniga, nii et võtme entroopia võib olla märksa suurem kui parooli entroopia. Kuna parooli entroopia on väike, võib ründajal õnnestuda see ära arvata ja üht poolt teeselda. Siiski, parooli entroopia loetakse piisavalt suureks, nii et üheainsa äraarvamiskatse õnnestumise tõenäosus on aktsepteeritavalt väike. Küll aga peavad parooliga autenditud protokollid olema konstrueeritud nii, et protokolliseansse pealt kuulates või isegi neisse aktiivselt sekkudes ei saaks ründaja teada midagi, mis tal hiljem, parooli jõuga äraarvamisel, aitaks kontrollida, kas tema pakutud parool on õige.

PACE protokolli on analüüsitud [37] sarnaselt OPACITY protokollidega ja leitud, et ta on turvaline. See tähendab, et tal on olemas käsitsi koostatud, mängude jadana antud turvatõestus. Tõestatud turvaomaduseks on *autenditud võtmevahetus* (*authenticated key exchange*, AKE) [35, 29], mis väidab, et aktiivne ründaja ei suuda mingis seansis korrumpeerimata poolte vahel kokkulepitud võtit eristada juhuslikust võtmest. AKE-st ei järeldu veel, et ründaja ei suuda üht poolt teisele teeselda, kuid AKE-t rahuldavast protokollist on lihtne konstrueerida protokolli, mis tagab ka osapooltevahelise autentimise [35]. PACE-is on seda konstruktsiooni täpselt järgitud.

E-passide lugemisel kasutatakse paroolina sama passinumbrist, kehtivuse lõpukuupäevast ja omaniku sünnipäevast tuletatud bitijada, mida BAC-protokollis kasutatakse pikaealise võtmena. Seega on PACE protokollijooksu lõppedes lugeja ja kiip teineteisele tõestanud, et nad teavad, mis on kirjas passi masinloetaval osal.

ICAO, olles PACE kasutusele võtnud, lubab siiski ka endiselt BAC-d kasutada [13], kui kiip või lugeja ei peaks PACE-t toetama. Enamgi veel, nõutakse, et kui PACE on toetatud, siis peab ka BAC toetatud olema. Sõltuvalt füüsilisest ja eetrijuurdepääsust võib ründajal olla siin võimalus kasutada vana versiooni rünnet, sundides näiteks kõigepealt kiipi ja lugejat BAC-d kasutama ja seejärel kinnipüütud seansist passi pikaealist võtit jõuga ära arvates.

Laiendatud pääsukontroll

Laiendatud pääsukontroll (EAC) on mõeldud täiendama BAC-d (või PACE-t), et kontrollida juurdepääsu e-passi kiibil salvestatud tundlikele isikuandmetele nagu omaniku sõrmejälg või silmaiirise kujutis. EAC-protokollil on kaks eesmärki [11]:

1) leppida kokku turvaline, suure entroopiaga seansivõti, mille abil kaitsta tundlikke isikuandmeid nende edastamisel kiibist terminali;
2) võimaldada kiibil kontrollida, kas terminalil on õigus neid andmeid näha.

Esimese punkti kommentaariks mainime, et ka PACE-protokolli tulemusel lepitakse kokku turvaline seansivõti. PACE toetamine ei ole aga kohustuslik ja BAC abil ei saa turvalisi seansivõtmeid luua.

Euroopa Liidu passides kasutatava EAC-protokolli on spetsifitseerinud BSI [16, 28]. Protokollis kasutatakse kiibi ja terminali sertifikaate, et nende abil kokku leppida seansivõti ja veenduda, et kumbki pool teab oma sertifikaadis olevale avalikule võtmele vastavat salajast võtit. EAC-protokolli on analüüsitud ja leitud ta olevat turvaline autenditud võtmevahetusprotokoll [52].

Käesoleva aruande koostajad leiavad, et kui puuduks nõue toetada BAC-d, siis oleksid e-passides kasutatavad krüptograafilised protokollid hästidisainitud ja tagaksid selles rakenduses vajalikud turvaomadused. Leiame, et PACE protokolli võiks kasutada mujalgi, kus kaks osapoolt saavad ennast teineteisele autentimisel ainult ühisele paroolile toetuda. Soovitame ICAO-l kaaluda BAC toe nõudmise lõpetamist.

6 Kuluanalüüsi metoodika krüptograafiliste algoritmide muutmisega kaasnevate kulude hindamiseks

Krüptograafilistel algoritmidel, nagu igal teiselgi lahendusel, on oma elutsükkel. See tsükkel algab algoritmi väljatöötamisega teaduskogukonnas, jätkub standardimise, teostamise ja juurutamisega ning lõppeb algoritmi murdumise ja käibelt kõrvaldamisega. Kuna krüptograafilised algoritmid on suhtlusprotokollides kasutusel küllalt madalal tasemel, nõuab nende väljavahetamine märkimisväärset aja- ja rahakulu.

Õnneks ei murdu krüptograafilised algoritmid üleöö ning nende asendamisega saab alustada juba esimeste nõrkuse märkide ilmnemisel. Sellegipoolest tuleb asendusprotsessi pikemalt ette planeerida.

Selle jaotise eesmärk ongi luua kuluanalüüsi metoodika kirjeldus rakendustes kasutatavate krüptograafiliste algoritmide muutmisega kaasnevate kulude hindamiseks. Niisugune metoodika ei saa aga olla krüptograafiliste algoritmide spetsiifiline. Krüptograafia on vaid üks IT-süsteemide komponent ning vastav kuluanalüüsi metoodika sobib ka üldisemalt süsteemide suurte muutuste mõjude hindamiseks.

Hinnata tuleb nii rahalist kui ka ajalist kulu, arvestades näiteks riikliku finantseerimise iseärasusi, kus raha saab/tuleb kulutada kindla perioodi jooksul ja seega võib ajakulu olla otsustava tähtsusega. Kuluanalüüs võtab arvesse kulusid alates otsusest, et muudatus vajab realiseerimist. Kuluanalüüsist on välja jäetud muudatuse realiseerimise otsusele eelnevad kulud. Samuti on välja jäetud kaasnevad kulud võimalike hangete korraldamise protseduuridega seoses.

Kirjeldatav metoodika käsitleb vaid muudatuste võimaliku kuluga seotud aspekte ega sisalda üldisi muudatuste halduse protseduure, sest need on piisava täpsusega standardites kirjeldatud. Näiteks teenuste muudatuste halduse üldine protsess on kirjeldatud standardis ISO/IEC 20000-2:2012 Information technology – Service management – Part 2: Guidance on the application of service management systems.

Siin kirjeldatud metoodika annab esmase alusraamistiku, kuid ei too rahalisi ega ajalisi hinnanguid konkreetsetele kuludele. Nende väljaselgitamiseks soovitame teha koostööd vastavate uurimisrühmadega, mis tuginevad oma uuringutes konkreetsetele näidetele ning nende raha- ja ajakulu statistikale.

6.1 Kuluanalüüsi metoodika skemaatiline kirjeldus

Metoodika jaotab krüptograafiliste algoritmide muutmise protsessi etappidesse, millega kaasnevaid kulusid ükshaaval hinnatakse.

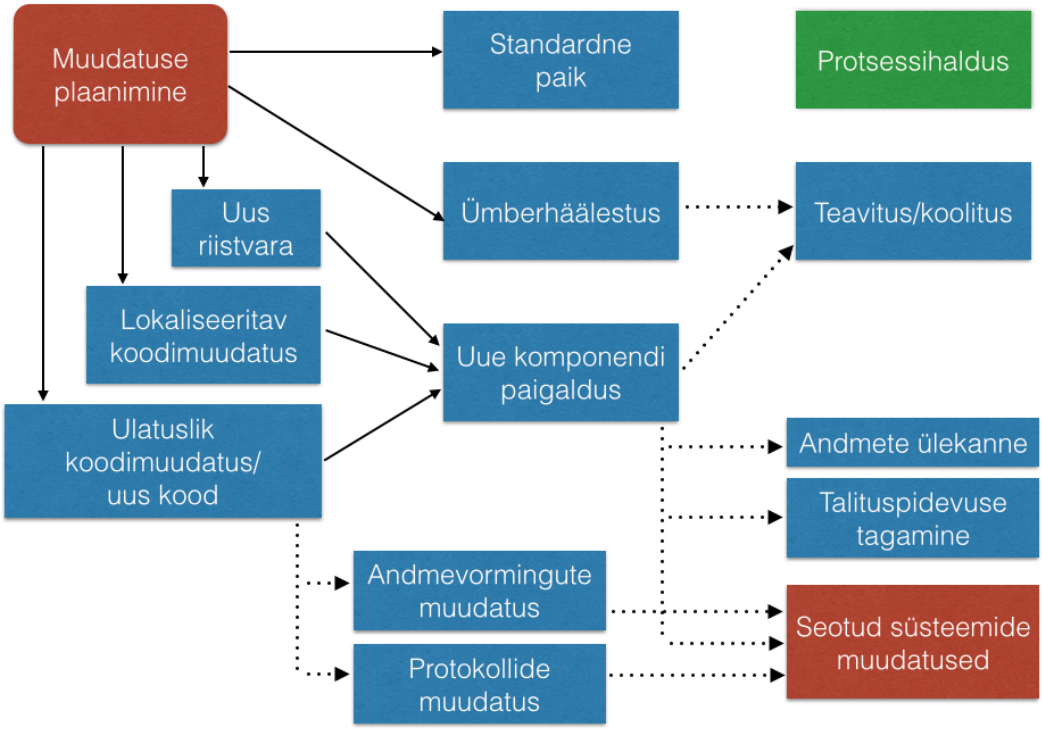
Muudatusega seotud kulude hindamist võib skemaatiliselt kujutada Joonisel 4 toodud vooskeemi läbimisena. Iga muudatus algab muudatuse klassifitseerimisest, planeerimisest ja projekteerimisest (vaskpoolne ülemine kast “Muudatus”), millega kaasnevad ka projekteerimiskulud, mis liidetakse jooksvale kogukulule (algselt 0).

Muudatuse liigist sõltuvalt siirdutakse ühte kastidest “Standardne paik”, “Ümberhäälestus”, “Uus riistvara”, “Lokaliseeritav koodimuudatus”, “Ulatuslik koodimuudatus/Uus kood”. Kui jõutakse kasti, millest edasi siiret ei ole, piirduvad kulutused seni kokkusaadud jooksvate kulutustega. Liituda võivad veel seotud süsteemide kulud. Punktiirjoonega nooled tähendavad võimalikku siiret.

Kui jõutakse parempoolsesse alumisse kasti “Seotud süsteemide muudatused”, siis see tähendab, et kogu skeem tuleb läbida uuesti, säilitades juba eelnevalt kokku liidetud jooksva kogukulu. Märgime, et alumisse parempoolsesse kasti võib jõuda kolmel erineval viisil, mis tähendab, et iga muudatus võib skeemi iga läbimisega kaasa tuua mitu erinevat seotud süsteemide muudatust, mis kõik vajavad eraldi hindamist sama skeemi järgi.

Metoodika võib realiseerida interaktiivse programmina, mis läbib vooskeemi, küsib kasutajalt vajalikke parameetreid ja peab arvet jooksva kogukulu üle. Kui kõik harud on täidetud, lisatakse summale ka protsessihalduse kulud (näiteks 10%).

Järgnevas kirjeldame neid etappe eraldi alajaotistena.



Joonis 4. Kuluanalüüsi metoodika skeem

6.2 Süsteemi analüüs

See on metoodika kõige kriitilisem etapp, millest sõltuvad kogu kuluhinnangu kvaliteet ja täpsus. Süsteemi analüüsil tuleb koostada ülevaade, milliseid komponente plaanitavad

muutused mõjutavad. Üldiselt jaotuvad need komponendid järgmistesse klassidesse:

- **tarkvara**– operatsioonisüsteemid, serveriplatvormid, rakendused jne,
- **riistvara**– võrguseadmed, füüsilised turvamoodulid, kiipkaardid jne.

Iga komponendiliigi puhul tuleb hinnata, kui palju vastavaid komponente käibel on ja milliseid toiminguid nende uuendamiseks vaja läheb.

Üldjoontes on uuemaid krüptoalgoritme võimalik kasutusele võtta kolmel viisil:

1) olemasolevaid süsteeme ümber seades,
2) süsteemidele täiendusi välja töötades ja
3) uusi süsteeme soetades.

Esimesel juhul on kulud seotud vaid juurutusega (vahel lisaks ka teavituse ja koolitusega); teisel ja kolmandal juhul tuleb läbi viia eraldi kuluanalüüs, mis arvestab ka arendustööde ja soetamise kulusid.

Arendustööd

Tarkvarasüsteemide, mille lähtekoodi üle Eestis kontroll on, täiendamiseks tuleb viia läbi vastavad arendustööd. Ka arendustööd algavad oma analüüsietapiga, mille eesmärk on kindlaks teha, kui suures ulatuses koodi muuta tuleb. Sisuliselt tuleb kood läbi vaadata ning hinnata muutmist vajavate tarkvarakomponentide arvu ja vajalike muutuste ulatust. Seejärel tuleb hinnata töö mahtu inimtundides ning selle korrutis inimtunni hinnaga ongi arendustööde kulu. Kindlasti tuleb eraldi aega plaanida ka testimisele ja koodi läbivaatuse käigus avastatud teiste vigade parandamisele. Aruande [14] järgi võib viimane kategooria võtta jämedalt kolmandiku kogu arendusmahust.

Arendustöid võib nende tüübi järgi jagada järgmistesse põhikategooriatesse.

- **Lokaliseeritavad muudatused koodis**, sh teekide täiendamine/vahetamine. Muudatusi vajab vaid hästipiiritletud osa koodist (näiteks teekide sisu). Muudatuste kulud on ligikaudu proportsionaalsed nimetatud koodiosa mahuga, st $\lambda \cdot R$, kus R on koodiridade arv ja λ on ühe koodirea keskmine maksumus. Uue teegi kasutuselevõtt võib tähendada ka vahekihi (*wrapper*) loomise kulusid, kui uue teegi liides ei ole identne vana teegi liidesega.
- **Koodi ulatuslik muutmine või uue koodi kirjutamine**. Näiteks krüptograafiliste algoritmide vahetuse kontekstis võib ulatuslik muudatus olla vajalik juhul kui krüptograafiliste atribuutidega seotud andmeväljad oluliselt muutuvad või muutub kogu koodi loomisel aluseks olnud loogika. Kogukulu on samuti $\lambda \cdot R$, kus R on koodiridade arv ja λ on ühe koodirea maksumus.

Koodi ulatuslike muudatustega võivad kaasneda järgmised arendusega seotud lisakulud.

- **Andmevormingute muutmine** andmebaasides/failides. Andmevormingute muutus võib põhjustada muudatusi ka seotud süsteemides, mille kulusid tuleb eraldi hinnata.
- **Protokollide muutmine**. Võib juhtuda, et näiteks andmeside turbeks kasutatava uue krüptograafilise algoritmi kasutuselevõtt ei ole võimalik hetkel kasutusel oleva protokolli raames. Sel juhul on vajalik uue protokolli kasutuselevõtt, mis põhjustab muudatusi ka seotud süsteemides, mille maksumust tuleb iga seotud süsteemi korral eraldi

hinnata. Protokollimuudatus ei tarvitse olla teostatav, sest protokolli kõigi osapoolte üle ei tarvitse olla kontrolli (näiteks välispartnerid).

Soetuskulud

On olemas rida komponente, mille arenduse üle puudub Eestis kontroll, näiteks operatsioonisüsteemid, võrguseadmed, üldotstarbeline serveri- ja kasutajatarkvara. Juhul kui neid pole võimalik konfigureerida turvalisemaid krüptoalgoritme kasutama, tuleb soetada uued komponendid. Nende soetuskulu on süsteemianalüüsi käigus leitud komponentide arvu korrutis ühe ühiku hinnaga. Lisaks võib suuremate koguste korral arvestada hulgiallahindlusega.

Muudatuste juurutamine

Muudatuse juurutamise kulud organisatsioonis sõltuvad orgranisatsiooni järgmistest parameetritest:

- N – kasutajate arv
- C_K – tööjõu tunnikulu

Muudatuse juurutamise kulud sõltuvad ka muudatuse liigist. Võib eristada järgmisi muudatuste liike.

- **Standardne tarkvarapaik**. On alla laetav ja teostatav standardsete vahenditega ja jõukohane pea igale süsteemi kasutajale. Eeldades, et turvapaiga allalaadimine ja paigaldus võtab T tundi, on kulud järgmised: Aeg: T tundi, Raha: $N \cdot C_K \cdot T$.
- **Uus häälestus**. Ei ole igale kasutajale jõukohane. Tavaliselt teeb uue häälestuse süsteemiülem. Paralleelne tegevus ei ole alati võimalik, kuid on ka erandeid, näiteks kogu asutuse viirusetõrjesüsteemi ümberhäälestamiseks piisab süsteemiülema ühekordsest muudatusest serveris. Eeldades, et ühe kasutajaga tegelemine võtab T tundi aega, siis süsteemiülema järjestikuse tegevuse korral on kulud järgmised:
  - aeg: $T \cdot N$,
  - raha: $N \cdot C_K \cdot T$.
- **Uue tarkvara/riistvara paigaldus**. Tarkvara/riistvara paigalduse ja häälestuse kulud sõltuvad paigaldatavate eksemplaride (st kasutajate) arvust N ja ühe eksemplari paigalduse ja häälestusega seotud ajakulust T järgmiselt:
  - aeg: $T \cdot N$,
  - raha: $N \cdot C_K \cdot T$.

Uue tarkvara/riistvara kasutuselevõtuga võivad kaasneda ka järgmised lisakulutused.

- **Andmete üleviimine** vanast süsteemist uude. Kulud sõltuvad üleviidavate andmete mahust D. Peamine kulu on ajakulu $\mu \cdot D$, kus μ on andmeühiku töötlemiseks vajalik aeg. Andmete ülekandeks võib vaja olla arendada teisendav programm, mille maksumust peab eraldi hindama.
- **Muudatused seotud süsteemides** (sh riiklikes andmebaasides), näiteks ID-kaartidele uute sertifikaatide soetamine. Kulusid tuleb eraldi hinnata.

- **Talituspidevuse tagamine** juhul kui uue lahenduse juurutamisel ei tohi esineda süsteemi seisakuid ja ei ole võimalik uut süsteemi vahetada kõikjal ja hetkeliselt, vaid see peab pika aja vana süsteemiga koos toimima. Tavaliselt nõuab talituspidevuse tagamine põhjalikku planeerimis- ja arendustööd, mille maksumust peab eraldi käsitlema.

Suuremõõtmelise süsteemiuuenduse juurutamise kulude juures võivad just lisakulutused olla kõigist kululiikidest suurimad ja samas ka kõige raskemini hinnatavad, sõltudes märkimisväärses ulatuses konkreetsest süsteemist.

Teavitus ja koolitamine

Juhul kui uuendus mõjutab lõppkasutajate vaadet süsteemile, tuleb arvestada ka kasutajate koolitamise kuludega. Tuleb leida ühe inimese koolitamise hind ning korrutada see koolitatavate inimeste arvuga. Suuremahulise teavituse korral võib osutuda otstarbekaks kuulutada välja koolitushange. Vajalikku koolituse mahtu on väga raske hinnata. Kindlasti ei saa selleks võtta fikseeritud protsenti projekti kogumaksumusest, sest tegelik protsent võib projektiti kardinaalselt erineda. Näiteks olid just teavitus- ja koolituskulud Eesti ID-kaardi juurutamisel kõige suurem kuluartikkel.

Olulise mõjuga projektide puhul võib osutuda vajalikuks ka ettevalmistav koolitus (teavituskampaania), mille käigus selgitatakse inimestele plaanitava muutuse vajalikkust.

Mõju eraettevõtetele

Krüptograafilisi meetodeid kasutatakse sidekanalite turvamiseks, sidepooled võivad aga potentsiaalselt olla väga erinevad. See tähendab, et krüptoalgoritmide uuendamise maksumus ei piirdu ainult süsteemi arendamise ja juurutamisega. Arvestada tuleb ka kaudseid kulusid, mida kannavad eraettevõtted.

Pärandsüsteemide turvamine

Kuna toimepidevuse tagamise vajaduse tõttu ei saa sageli uut süsteemi rakendada üleöö, tuleb üleminekuperioodil arvestada vana süsteemi turvamisest tulenevate lisakuludega.

Protsessihaldus

Krüptograafiliste algoritmide uuendamise protsess on keeruline ning selle haldamiseks tuleb eraldi ressurssi eraldada. Hinnanguliselt kulub protsessi juhtimisele vähemalt 10% projekti kogukuludest, mis tuleb lisada muude kulude hinnangule. Samuti kuulub protsessihalduse juurde intsidendikäsitlus, nt meediaga suhtlemine juhul kui uute lahenduste juurutamise käigus on vana lahenduse toimimine katkenud.

7 Post-kvantkrüptograafia ülevaade

7.1 Kvantarvutid

Kvantarvutid võimaldavad sooritada paralleelarvutusi, kasutades kvantmehaanikast tuntud ja mikromaailmas kehtivat superpositsiooni printsiipi, mille järgi mikromaailma protsessid arenevad paralleelselt kõikvõimalikke teid pidi kuni neid mõõdetakse ja alles siis selgub üks konkreetne tee. Näiteks võib n-bitine (sobivalt eelseadistatud) kvantregister olla korraga kõigis 2^n võimalikus olekus, nn *superpositsioonis*. Iga funktsiooni $f(x)$ saab teostada kvantskeemina nii, et kui sisendväärtus on kõikvõimalike argumentide x superpositsioon, siis tulemusena on ka väljund kõikvõimalike väärtuste $y = f(x)$ superpositsioon. Klassikalises arvutis vastaks sellele paralleelarvutus, kus korraga töötab 2^n lõime (*thread*). Et aga paraleelarvutus töötaks sama kiiresti kui üksainus lõim, tuleks kasutada 2^n korda rohkem arvutusressurssi.

Kvantarvutis saaks sellise arvutuse sooritada nö. tasuta, st selle jaoks oleks vaja niisama palju arvutusressurssi kui üheainsa lõime arvutamiseks. Probleem seisneb aga selles, et kõik lõimed ei ole korraga kättesaadavad. Kui mõõta (lugeda) väljundit, siis saame üheainsa juhuslikult (mingi tõenäosusjaotusega) valitud väärtuse $y = f(x)$. Selline paralleelarvutus ei annaks midagi olulist juurde klassikalise ühelõimelise arvutusega võrreldes, sest see oleks sama, mis arvutada $f(x)$ juhuslikult valitud argumendist x.

Klassikalises paralleelarvutuses on kõik lõimed ja nende tulemid korraga kättesaadavad ja lõimed võivad töötamise jooksul informatsiooni vahetada suvalisel viisil. Ka kvantarvutites saab korraldada lõimedevahelist infovahetust, kuid seda äärmiselt piiratult. Näiteks kui kõik lõimed arvutaksid mingit ühe-bitilist suurust (predikaati), siis kvantarvuti korral ei ole teada, kas saab kuidagi arvutada kõigi lõimede väljundbittide korrutist (mis on 1 parajasti siis kui kõik lõimed väljastavad 1). Kui see oleks võimalik, siis sellest järelduks, et kvantarvutiga saab lahendada **NP**-täielikke ülesandeid polünomiaalses ajas.

Üks ülesanne, mida kvantarvutiga hästi lahendada saab (kasutades Shori kvantalgoritmi [98]), on funktsiooni *perioodi* leidmine. Funktsiooni f perioodiks nimetatakse rangelt positiivset täisarvu λ, nii et $f(x+\lambda) = f(x)$ iga x korral. Näiteks RSA avaliku võtmega krüpteerimisfunktsiooniga $E(m) = m^e \bmod n$ seotud (tõhusalt arvutatava) funktsiooni $f(x) = a^x \bmod n$ periood on (vahemikust $(0...n)$ juhuslikult valitud a korral) suure tõenäosusega nn. *Carmichaeli funktsioon* $\lambda(n)$, mida teades saab Eukleidese laiendatud algoritmi abil arvutada RSA salajase eksponendi $d = \frac{1}{e} \bmod \lambda(n)$. Seetõttu on RSA krüpteerimisalgoritm kvantarvuti abil tõhusalt murtav. Ka näiteks elliptkõveratel põhinevad krüpteerimisalgoritmid on murtavad sarnasel põhjusel.

Mõnede krüpteerimisalgoritmide korral ei ole aga sellist seost salajase võtme ja mingi avaliku fukntsiooni perioodi vahel teada. Seetõttu ei ole ka teada, kas kvantarvutid annavad nende murdmiseks võrreldes klassikaliste algoritmidega midagi juurde.

7.2 Post-kvantkrüptograafia määratlus

Post-kvantkrüptograafia on kvantarvutitele vastupidavate krüptosüsteemide koondnimetus. Praegused laialt kasutuses olevad avaliku võtmega krüptosüsteemid põhinevad enamasti kas suure täisarvu teguriteks lahutamise või diskreetse logaritmi leidmise arvutusmahukusel, mis aga tulevaste kvantarvutite jõudluse puhul probleeme ei tekita. Juba avaliku võtmega krüptograafia algusaastaist on teada üksikuid kvantrünnete vastu turvalisi krüptograafilisi algoritme, nagu näiteks Lamporti signatuuriskeem. Alates aastast 2005 otsivad krüptoloogid aga teadlikult teistsuguseid algoritme, mis oleksid turvalised ka kvantarvutiga sooritatud rünnete vastu.

Post-kvantkrüptograafia vaatleb järgmisse nelja rühma kuuluvaid krüptograafilisi algoritme:

- võrepõhised,
- mitmemuutujalised,
- räsipõhised,
- koodipõhised.

Ka tavalised plokkšifrid (näiteks AES) jäävad tänaste teadmiste põhjal kvantarvutite ilmudes turvaliseks, kuid plokkšifrite uurimist ja konstrueerimist ei loeta enamasti post-kvantkrüptograafia valdkonda kuuluvaks.

7.3 Võrepõhine krüptograafia

Võresid uurisid esmakordselt matemaatika klassikud Joseph Louis Lagrange ja Carl Friedrich Gauss. Krüptograafias on võresid kasutatud näiteks arvkongruentsidel põhinevate pseudojuhuarvugeneraatorite krüptoanalüüsis. Miklós Ajtai näitas 1996. aastal esmakordselt, et võresid saab kasutada ka uute krüptosüsteemide loomisel [30]. Craig Gentry kasutas 2009. aastal võresid esimese täishomomorfse krüptosüsteemi loomisel [66].

Võreks[15] nimetatakse eukleidilise ruumi $\mathbb{R}^n$ vektorite (punktide) diskreetset alamhulka, mis on kinnine vektorite liitmise ja lahutamise suhtes. Öeldakse, et võre dimensioon on n kui võre ei sisaldu ruumi $\mathbb{R}^n$ üheski pärisalamruumis (st madalama dimensiooniga alamruumis). Visuaalselt kujutatuna on võre kogu ruumi ulatuses korrapäraselt paiknevate punktide kogum. Algebralise süsteemina on võre lõplikult genereeritud vaba Abeli rühm.

Võre L baasiks nimetatakse vektorite hulka B, nii et võre L iga punkt (vektor) avaldub ühesel viisil hulga B elementide täisarvulise lineaarkombinatsioonina. Kui võre dimensioon on vähemalt 2, siis on võres alati lõpmatu arv baase. Krüptograafias on aga vaja, et avatekst, krüptogramm ja võti oleksid lõplikud bitijadad. Seetõttu kasutatakse krüptograafias võresid, mis on mitte ruumis $\mathbb{R}^n$ vaid K^n, kus K on mingi lõplik korpus.

Krüptograafias kasutatavad võredega seotud rasked kombinatoorikaprobleemid on järgmised.

- *Lühima vektori probleem*: Leida baasiga B esitatud võre L lühim vektor.
- *Lähima vektori probleem*: Baasiga B esitatud võre L ja vektori $v \notin L$ korral leida võre vektor $v' \in L$, mis on lähim vektorile v.

[15] Võrel on matemaatikas ka teine tähendus, mis on seotud osaliselt järjestatud hulkadega.

Need probleemid usutakse olevat üldjuhul (enamiku baaside B korral) raskesti lahenduvad. Kui aga baasivektorid on lühikesed ja peaaegu ortogonaalsed, muutuvad mõlemad probleemid kergesti lahenduvaks. Sellise baasi koostamist nimetatakse võre baasitaanduseks (*lattice basis reduction*) ja tuntuim algoritm taanduse teostamiseks on Lenstra–Lenstra–Lovászi (LLL) algoritm [78].

Turvalisust arvestades jagunevad võrepõhised krüptosüsteemid kahte klassi.

- Turvatõestuseta, kuid väga tõhusad ja konkureerivad parimate teadaolevate algoritmidega, näiteks NTRU algoritmide perekond.
- Turvatõestusega, kuid enamasti vähetõhusad ja ebapraktilised. Näiteks vigadega õppimisel (*Learning with Errors, LWE*) põhinevad algoritmid.

Ringidel põhinev vigadega õppimine (Ring-LWE) aga lubab konstrueerida krüptosüsteeme, mis on enam-vähem tõhusalt arvutatavad ja samas ka formaalse turvatõestusega.

7.4 Mitmemuutujaline krüptograafia

On tõestatud, et mitmemuutujaliste algebraliste võrrandite lahendamine on **NP**-raske või **NP**-täielik ja teoreetikute hinnanguil ei suuda kvantarvutid **NP**-täielikke ülesandeid polünomiaalses ajas lahendada. Seega on mitmemuutujalistel võrranditel põhinevad krüptosüsteemid post-kvantkrüptograafiaks sobivad kandidaadid.

Tänaseks on teada vaid mitmemuutujalistel võrranditel põhinevaid *signeerimisskeeme* nagu *Unbalanced Oil and Vinegar (UOV)* [71], *Hidden Field Equations (HFE)*, *Hidden Field Equation Vinegar (HFEv)* [90] ja Rainbow [56]. Need signeerimisskeemid on vähese arvutusmahuga ja sobivad väikese arvutusvõimsusega seadmeisse (nt kiipkaarti), kuid nõuavad suhteliselt pikka võtit (paar tuhat baiti).

Pea kõik mitmemuutujalised (avaliku võtmega) *krüpeerimisskeemid* on aga osutunud ebaturvalisteks. Näiteks 1988. aastal esitatud Imai ja Matsumoto krüptosüsteemi murdis Patarin aastal 1995. Patarini enda poolt pakutud täienduse *Little Dragon* (“Väike Draakon”) [89] aga murdsid Coppersmith ja Patarin ise aastal 1996 [72].

Krüptograafias kasutatav mitmemuutujaliste polünoomidega seotud kombinatoorikaprobleem on algebralise võrrandisüsteemi lahendamine üle lõpliku korpuse, näiteks $\mathbb{Z}_2 = \{0, 1\}$, kus tuleb lahendada võrrandisüsteem

$$\begin{aligned} P_1(x_1, x_2, \ldots, x_{2n}) &= y_1 \\ P_2(x_1, x_2, \ldots, x_{2n}) &= y_2 \\ &\ldots \\ P_n(x_1, x_2, \ldots, x_{2n}) &= y_n \ , \end{aligned} \tag{2}$$

kus $P_1, \ldots, P_n \in \mathbb{Z}[x_1, \ldots, x_{2n}]$ on polünoomid astmega ülimalt kaks.

Sellel kombinatoorikaülesandel põhinev signatuuriskeem võib kasutada avaliku võtmena polünoomide $P_1, \ldots, P_n$ kirjeldusi. Sõnumi M signatuur on $3n$-bitine ning koosneb $2n$ bitist $x_1, \ldots, x_{2n}$ ja lisaks veel n-bitisest juhuarvust r, nii et

$$H(r, M) = P_1(x_1, \ldots, x_{2n}) || \ldots || P_n(x_1, \ldots, x_{2n}) \ ,$$

kus H on mistahes n-bitise väljundjadaga räsifunktsioon. Sõnumi M signeerimiseks tuleb esmalt arvutada räsi $H(r, M) = y_1 y_2 \ldots y_n$ kasutades juhuarvu r ja seejärel lahendada

võrrandisüsteem (2). Polünoomid P_i on valitud erilisel viisil, nii et teades nende salajast struktuuri– nn. HFE (*Hidden Field Equation*) struktuuri– on võrrandisüsteemi lihtne lahendada, kuid selle salajase struktuuri avastamiseks (kas avalikust võtmest või juba moodustatud signatuuridest) ei ole teada tõhusaid algoritme. Sellise signeerimisalgoritmi esitas Patarin aastal 1996 ja see on tänini turvaline.

7.5 Räsipõhine krüptograafia

Räsifunktsioonidel põhinevad ühed vanimaist signeerimisskeemidest – Lamporti ja Merkle'i signatuuriskeemid, mis loodi juba eelmise sajandi 70-ndate aastate lõpul. Räsipõhiseid signeerimisskeeme on uuritud juba kaua ja neid loetakse turvaliseks eeldusel, et kasutatav räsifunktsioon on turvaline. Räsipõhiste signatuuriskeemide põhipuudus on võimalike signatuuride piiratud arv.

Ühe biti b signeerimiseks Lamporti signatuuriskeemiga on vaja kahest n-bitisest (pseudo)juhuarvust k_0 ja k_1 koosnevat privaatvõtit. Avalik võti on paar $f(k_0), f(k_1)$, kus f on mingi ühesuunaline funktsioon, näiteks räsifunktsioon. Biti $b \in \{0,1\}$ signeerimiseks avalikustab signeerija poole oma privaatvõtmest, nimelt k_b, ja kustutab (unustab) teise poole k_{1-b}. Ühe võtmega saab signeerida vaid ühe biti.

Pikema sõnumi M signeerimiseks on vaja n (näiteks $n = 256$) privaatvõtit $(k_0^1, k_1^1), \ldots, (k_0^n, k_1^n)$. Sõnum räsitakse ja selle räsi $y = y_1 y_2 \ldots y_n = H(M)$ iga bitt y_i signeeritakse privaatvõtmega (k_0^i, k_1^i) eelnevalt kirjeldatud viisil. Signatuuri suurus on $n = 256$ korral seega $n^2 = 256 \cdot 256 = 65536$ bitti ehk 8 kilobaiti, avaliku võtme maht ühe sõnumi signeerimiseks aga 16 kilobaiti.

Privaatvõtme mahu vähendamiseks kasutatakse pseudojuharvude generaatorit, mis ühest n-bitisest salajasest võtmest genereerib kõik vajalikud privaatvõtmete komponendid k_j^i.

Merkle'i signatuuriskeem on Lamporti skeemi edasiarendus, kus avaliku võtme mahu vähendamiseks kasutatakse räsipuud, mille juurräsi on avalik võti. Merkle'i signatuuriskeemis saab ühe avaliku ja privaatvõtmega signeerida palju sõnumeid, ehkki nende arv on piiratud. Kuni m signeerimist võimaldava võtme korral on signatuuri suurus $n^2 + n \log_2 m$ bitti, seega mitte oluliselt suurem kui ühekordset signeerimist võimaldava skeemi korral. Näiteks $m = 1024$ ja $n = 256$ korral on signatuur $256 \cdot (256 + 10)$ bitti ehk umbes 8.5 kilobaiti. Merkle'i signatuuriskeemi turvalisus põhineb kasutatava räsifunktsiooni kollisioonikindlusel.

Lamporti ja Merkle'i signatuuriskeemi modifikatsioon, XMSS signatuuriskeem [44], mis $n = 256$ korral annab ühe sõnumi signeerimiseks vajaliku signatuuri suuruseks umbes 1.8 kilobaiti.

Lamporti ja Merkle'i signatuuriskeemi teist tüüpi modifikatsioon, mille esitasid Buldas, Laanoja ja Truu aastal 2014 [47, 45, 46], võimaldab tunduvalt vähendada signatuuri mahtu. Ühe sõnumi signeerimiseks vajalik privaatvõti koosneb nende skeemis vaid ühestainsast n-bitisest räsist, kuid signeerimine saab toimuda ainult koostöös serveriga. Sõnumi M signeerimiseks hetkel t on vaja võtit k_t, mis on mõeldud kasutuseks ainult hetkel t. Signeerija arvutab sõnumi M räsi $H(M)$ ja moodustab signeerimispäringu $P(H(M), k_t)$, mis kombineerib räsi $H(M)$ ja võtit k_t. Server väljastab vastuseks päringule ajatempli $S_T(P(H(M), k_t))$. Signatuur on kehtiv vaid siis, kui ajatempli aeg T langeb kokku võtme kasutuseks ette nähtud ajaga t. Korrektse signatuuri saamiseks on seega vaja sekundise täpsusega sünkroniseeritud kelli. Signatuuri maht, juhul kui $m \approx 2^{25}$ (sekundite arv aastas), on signatuuri

suurus ligikaudu $25{\cdot}256$ bitti, ehk 800 baiti. Räsifunktsiooni kollisioonikindlusest tõenäoliselt ei piisa sellise signatuuriskeemi turvalisuse tõestamiseks, vaid tuleb kasutada rangemaid eeldusi.

Signatuuriskeem SPHINCS [40] on esimene olekuvaba räsipõhine signatuuriskeem, kus ei ole vaja arvet pidada signeeritavate sõnumite arvu kohta (st muuta privaatvõtme olekut) ega suhelda serveriga. SPHINCS kasutab umbes 1 kilobaidist avalikku- ja 1 kilobaidist privaatvõtit, kuid signatuuri suurus on ligikaudu 42 kilobaiti. Skeemi turvalisus põhineb kollisioonikindlusel ilma lisaeeldusteta.

7.6 Koodipõhine krüptograafia

Koodipõhine krüptograafia põhineb veaparanduskoodide omadustel. On olemas nii koodipõhiseid krüpteerimisskeeme [82] kui ka signeerimisskeeme [87].

McEliece'i krüptosüsteemiga seotud kombinatoorikaprobleem on üldise lineaarse koodi dekodeerimisprobleem ehk *lähima koodsõna* probleem. Lineaarkood on lineaarkombinatsioonide suhtes kinnine vektorite hulk mingis n-mõõtmelises vektorruumis üle lõpliku korpuse $\mathbb{F}_q$, s.t. ta on ruumi $\mathbb{F}_q^n$ mingi k-mõõtmeline alamruum. Alamruumi dimensiooni k nimetatakse ka koodi järguks (*rank*). Lähima koodsõna probleemi lahendamine seisneb valitud vektorile x Hammingi kauguse mõttes lähima koodsõna (alamruumi elemendi) leidmises. On tõestatud, et kõige üldisem selle ülesande versioon on **NP**-raske. Paljudel erijuhtudel ja piiratud vigade korral on see ülesanne aga lihtsasti lahenduv, näiteks kui vektori x Hammingi kaugus alamruumist ei ületa poolt koodi *minimaalkaugusest* (minimaalne Hammingi kaugus kahe erineva koodsõna vahel).

Privaatvõtmena kasutatakse juhuslikult valitud lineaarkoodi, mille dekodeerimisalgoritm on tõhus ja teada. McEliece'i krüptosüsteemi originaalversioonis kasutatakse binaarseid Goppa koode, mis on kergesti dekodeeritavad Pattersoni algoritmi abil ja mis suudavad parandada kuni t bitiviga. Avalik võti saadakse, kui valitud kood maskeeritakse üldise lineaarkoodiga. Kui G on koodi generaatormaatriks (mille reavektorid moodustavad koodi kui alamruumi baasi), siis avalik võti G' saadakse juhuslikult valitud pööratavate maatriksite S ja P abil järgmiselt:

$$G' = S \cdot G \cdot P \ ,$$

kus $\cdot$ tähendab maatriksite korrutamist (seejuures on P permutatsioonimaatriks, mille igas reas ja veerus on täpselt üks 1) .

Sõnumi m krüpteerimiseks McEliece krüptosüsteemi abil kodeeritakse m esmalt t-bitise binaarstringina, arvutatakse vektor $c' = mG'$, genereeritakse n-bitine juhuslik vektor z, mille koordinaatidest täpselt t on võrdsed ühega, ja moodustatakse krüptogramm $c = c' + z$. Krüptogramm on seega koodsõna, kuhu on lisatud juhuslik t-bitine viga.

Krüptogrammi dekrüpteerimiseks arvutatakse maatriksi P pöördmaatriksi P^{-1} abil $c' = cP^{-1}$, dekodeeritakse c' leides m' ning leitakse avatekst $m = m'S^{-1}$. Dekrüpteerimine on korrektne, sest

$$c' = cP^{-1} = mG'P^{-1} + zP^{-1} = mSG + zP^{-1} \ ,$$

mSG on koodsõna ja vektori zP^{-1} Hammingi norm ei ületa t (P on permutatsioonimaatriks).

McEliece'i krüptosüsteem on väga tõhus nii võtme genereerimise, krüpteerimise kui ka dekrüpteerimise kiiruse mõttes, kuid tema peamine puudus on väga suur avalik võti. Näiteks 2^{128}-turvalisuse saavutamiseks peaks võti olema enam kui 100 kilobaidine.

7.7 Teostused ja standardid

Laialt levinud krüptoteegid (näiteks OpenSSL ja Bouncy Castle) ei toeta post-kvantkrüptograafia algoritme. Ehkki räsifunktsioonid ise usutakse olevat turvalised kvantrünnete suhtes, ei ole ka räsipõhised digitaalsignatuurid nendes teekides toetatud. Võrepõhistest krüpteerimisalgoritmidest on kõige paremini toetatud NTRU [16] [17]. Kõik muud post-kvantkrüptograafia algoritmid aga on teostatud üksnes eksperimentaalkorras ja nende laiem kasutuselevõtt eeldab mahukat arendustööd.

ETSI on asutanud post-kvantkrüptograafiale pühendatud töögrupi [18], kuid standardite väljatöötamine on alles algjärgus.

[16] http://tbuktu.github.io/ntru/

[17] https://github.com/NTRUOpenSourceProject/ntru-crypto

[18] ETSI launches Quantum Safe Cryptography specification group, http://www.etsi.org/news-events/news/947-2015-03-news-etsi-launches-quantum-safe-cryptography-specification-group

8 Kokkuvõtteid ja soovitusi

Alates aruande [17] valmimisest 2013. aastal pole olulisi krüptoanalüütilisi läbimurdeid toimunud. See tähendab, et aruande [17] soovitused on üldjoontes endiselt õiged. Kordame siinkohal neist olulisemaid.

- Sümmeetrilistest krüptoalgoritmidest soovitame kasutada plokkšifrit AES. Keskpikas perspektiivis (kuni 10 aastat) on turvalised kõik standardsed võtmepikkused (128, 192, 256 bitti). Ülipikas perspektiivis (30-50 aastat) turvalisuse saavutamiseks soovitame võtmepikkust 256 bitti. Samuti võib jätkuvalt turvaliseks pidada šifrit Camellia, kuid uutest rakendustes ei tohiks enam kasutada šifrit Blowfish. Loobuda tuleb šifrite RC4, DES ja 3DES kasutamisest.
- RSA ning diskreetsele logaritmile põhinevate krüptosüsteemide (nt Diffie-Hellmani võtmevahetus, DSA) korral tuleb kõikjal võimalikult kiiresti loobuda kuni 1024-bitistest võtmetest. 5 aasta perspektiivis sobivad kasutamiseks 2048-bitised, keskpikas perspektiivis vähemalt 3072-bitised ning ülipikas perspektiivis vähemalt 15360-bitised võtmed.
- Räsifunktsioonide osas tuleb üldise soovitusena loobuda funktsioonide MD5 ja SHA-1 kasutamisest. 5 aasta perspektiivis sobivad kasutamiseks kõik SHA-2 perekonna liikmed. 10 aasta perspektiivis tuleks loobuda SHA-224 kasutamisest; teised SHA-2 perekonna liikmed on sellel ajahorisondil suure tõenäosusega jätkuvalt turvalised. SHA3 standardi lõpliku kinnitamise järel on mõistlik kaaluda sellele üleminekut.
- Elliptkõverate osas on hetkel kõige rohkem toetatud NIST-i kõverad, eriti P-256. Kui rakenduse spetsiifikast lähtudes pole võimalik kasutada näiteks RSA süsteemi piisavalt pika võtmega, kujutab P-256 endast pragmaatilist alternatiivi. Samas tuleb arvestada teoreetilise võimalusega, et NIST-i kõverate loomisel on neisse kavandatud tagauksed. NIST algatas 2015. aastal uute, tagauksekahtlusest vabade elliptkõverate standardimise tsükli, kuid see nõuab enne praktiliste teostusteni jõudmist vähemalt 5 aastat.

Peale ülaltoodud soovituste tuleb TLS-i šifrikomplekti valimisel määrata veel kasutatav võtmevahetusalgoritm ning plokkšifri töörežiim.

Tulevikuturvalisuse tagamiseks (vt jaotis 5.5) soovitame kasutada efemeerset Diffie-Hellmani võtmevahetust, millele TLS-i šifrikomplektide nimes viitab lühend DHE või EDH.

Kuna klassikalised plokkšifrite töörežiimid (nt CBC) pole disainitud pakkuma sõnumi terviklust, lisatakse neile TLS-protokollistikus ühe võimalusena veel sõnumiautentimiskood (MAC-kood, vt [14]). TLS-i versioon 1.2 pakub alternatiivina ka autenditud krüpteerimise töörežiimi GCM, mis ühendab endas nii privaatsuse kui tervikluse tagamise funktsionaalsuse.

Kuna TLS-i järgmises versioonis 1.3 plaanitakse mitte-efemeerse võtmevahetuse ning mitteautenditud krüpteerimise tugi üldse kaotada (vt [93]), soovitame juba praegu šifrikomplekti valimisel eelistada efemeerset võtmevahetust ning autenditud töörežiime. Kokkuvõtteks soovitame TLS-i näidisprofiilidena järgmisi šifrikomplekte:

- `TLS_DHE_RSA_WITH_AES_128_GCM_SHA256`,
- `TLS_DHE_RSA_WITH_AES_256_GCM_SHA384`.

Nende soovituste praktikasse rakendamisel tuleb arvestada, et kõik kasutatavad programmid ja seadmed ei pruugi maksimaalset võimalikku turvalisust pakkuvaid šifrikomplekte toetada. Samuti seab šifrikomplekti valikule omad piirid vajadus tagada koostalitlusvõime erinevate osapoolte vahel.

Ka mitmete teiste TLS-profiilide turvatase on praktikas kasutamiseks piisav, kuid nende valikul tuleb arvestada selle jaotise alguses toodud soovitusi. Keelata tuleb juba teadaolevalt ebaturvalised algoritmid (nt RC4, DES).

2014. aastal leiti ja parandati mitmeid TLS-protokollistiku teostusvigu (nt OpenSSL-teegis). Seetõttu anname üldise soovituse kasutada protokollide ja teekide uusimaid versioone. Tuleb lõpetada SSLv3 protokollistiku kasutamine ning sulgeda teenused, mis uuemaid protokollistikke ei toeta. Üldise soovitusena tuleks TLS-ist kasutada võimalikult uut versiooni.

Teine üldine soovitus on arvestada uute lahenduste projekteerimisel juba eos vajadusega krüptograafilisi algoritme aegajalt uuendada. Seega peaks nende algoritmide kasutamine olema võimalikult modulaarne, näiteks konfiguratsioonifaili tasemel muudetav.

Peale uute rakenduste turvalise loomise tuleb hoolitseda ka pärandlahenduste turvalisuse tagamise eest. Nii soovitame vanade digiallkirjade pikaajalise kehtivuse tagamiseks lisada baastarkvarasse digiallkirjade üle-ajatembeldamise funktsionaalsuse.

Seoses mobiilseadmete muutumisega järjest olulisemaks arvutiplatvormiks on kasvanud ka vajadus eID lahenduste pakkumise järele mobiilplatvormidel. Esimeseks väljakutseks niisuguste lahenduste loomisel on privaatvõtmete kaitse. Kui loobuda võtmete hoidmisest kiibil, pole privaatvõtmetele praegustes mobiilseadmetes head alternatiivset säilituskohta. Hetkel tundub parimaid võimalusi pakkuvat usaldatav täitmiskeskkond (TEE), kuid see alles hakkab erinevate tootjate seadmetesse jõudma, mistõttu tema poolt pakutavat turvataset on veel vara hinnata.

Teiseks probleemiks kontaktivabade eID lahenduste puhul on nende ühendus mobiilseadmega, mis peab toimuma üle raadiokanali (nt lähiväljaside). On teada, et Eestis on väjatöötamisel NFC-digi-ID lahendus, kuid kavandatava sideprotokolli detailid polnud selle aruande autoritele kättesaadavad. Siinkohal soovitame vältida firmapäraseid, salajase spetsifikatsiooniga ning teadmata turvaomadustega protokolle ning võtta kasutusele avaliku kogukonna poolt sõltumatult analüüsitud lahendused (nt OPACITY).

2014. aasta aprillis kerkinud küsimused kettakrüptolahenduse TrueCrypt kohta said vastatud 2015. aasta alguses, mil ilmus TrueCrypti koodiauditi kokkuvõte. Audiitorid ei tuvastanud koodibaasist suuri probleeme ega tagauksi, mistõttu võib TrueCrypti kasutamist jätkuvalt turvaliseks pidada. Probleeme võib tekitada aktiivse tootetoe puudumine, mistõttu missioonikriitilistes rakendustes soovitame kasutada operatsioonisüsteemide endi poolt pakutavaid kettakrüptorakendusi.

Kirjandus

[1] ANSI X9.62:2005. Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA).

[2] Bouncy Castle cryptographic toolkit. http://www.bouncycastle.org/.

[3] Digital Signature Standard (DSS). FIPS PUB 186-4. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

[4] Elliptic Curve performance: NIST vs Brainpool. https://polarssl.org/kb/cryptography/elliptic-curve-performance-nist-vs-brainpool.

[5] GlobalPlatform Device Specifications. http://www.globalplatform.org/specificationsdevice.asp.

[6] Network Security Services. https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS.

[7] OpenSSL SSL/TLS toolkit. http://www.openssl.org.

[8] Spongy Castle – repackage of Bouncy Castle for Android. http://rtyley.github.io/spongycastle/.

[9] The GnuTLS Transport Layer Security Library. http://www.gnutls.org/.

[10] Wireless Transport Layer Security, 2001. http://technical.openmobilealliance.org/tech/affiliates/wap/wap-261-wtls-20010406-a.pdf.

[11] Doc 9303: Machine Readable Travel Documents, Part 1: Machine Readable Passports, Volume 2: Specifications for Electronically Enabled Passports with Biometric Identification Capability, 6th edition, 2006. http://www.icao.int/publications/pages/publication.aspx?docnum=9303.

[12] SEC 2: Recommended Elliptic Curve Domain Parameters, 2010. http://www.secg.org/sec2-v2.pdf.

[13] Supplemental Access Control for Machine Readable Travel Documents, Version 1.01. Technical report, ISO/IEC JTC1 SC17 WG3/TF5, 11. november 2010.

[14] Krüptograafiliste algoritmide kasutusvaldkondade ja elutsükli uuring. http://www.riso.ee/sites/default/files/elfinder/article_files/kryptoalgoritmide_elutsykli_uuring.pdf, 2011. AS Cybernetica aruanne nr A-60-1.

[15] Elliptic Curve Cryptography. https://www.bsi.bund.de/cae/servlet/contentblob/471398/publicationFile/30615/BSI-TR-03111_pdf.pdf, 2012.

[16] Technical Guideline TR-03110-1: Advanced Security Mechanisms for Machine Readable Travel Documents - Part 1 - eMRTDs with BAC/PACEv2 and EACv1, Version 2.10, 20. märts 2012.

[17] Krüptograafiliste algoritmide kasutusvaldkondade ja elutsükli uuring. https://www.ria.ee/public/PKI/kruptograafiliste_algoritmide_elutsukli_uuring_II.pdf, 2013. AS Cybernetica aruanne nr A-77-5.

[18] Chinese MITM Attack on iCloud, October 2014. http://www.netresec.com/?page=Blog&month=2014-10&post=Chinese-MITM-Attack-on-iCloud.

[19] How MS14-066 (CVE-2014-6321) is More Serious Than First Thought, November 2014. http://www.malwaretech.com/2014/11/how-ms14-066-winshock-is-worse-than.html.

[20] Insufficient SSL certificate validation. Pidgin Security Advisory, October 2014. http://pidgin.im/news/security/?id=86.

[21] libcurl not verifying certs for TLS to IP address / Darwinssl, March 2014. http://curl.haxx.se/docs/adv_20140326C.html.

[22] libcurl not verifying certs for TLS to IP address / Winssl, March 2014. http://curl.haxx.se/docs/adv_20140326D.html.

[23] OpenSSL TLS protocol downgrade attack (CVE-2014-3511). OpenSSL Security Advisory [6 Aug 2014], August 2014. https://www.openssl.org/news/secadv_20140806.txt.

[24] Pangalingi päringute tehniline spetsifikatsioon, October 2014. http://pangaliit.ee/et/arveldused/pangalingi-spetsifikatsioon.

[25] The Heartbleed Bug, 2014. http://heartbleed.com.

[26] Vulnerability in Schannel Could Allow Remote Code Execution (2992611). Microsoft Security Bulletin MS14-066, November 2014. https://technet.microsoft.com/library/security/ms14-066.

[27] Vulnerability Summary for CVE-2014-1266. National Vulnerability Database, March 2014. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-1266.

[28] Technical Guideline TR-03110-2: Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token - Part 2 - Protocols for electronic IDentification, Authentication and trust Services (eIDAS), Version 2.20, 3. veebruar 2015.

[29] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005, Proceedings*, volume 3386 of *Lecture Notes in Computer Science*, pages 65–84. Springer, 2005.

[30] Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.

[31] Steve Babbage, Dario Catalano, Carlos Cid, Benne de Weger, Orr Dunkelman, Christian Gehrmann, Louis Granboulan, Tim Güneysu, Jens Hermans, Tanja Lange, Arjen Lenstra, Chris Mitchell, Mats Näslund, Phong Nguyen, Christof Paar, Kenny Paterson, Jan Pelzl, Thomas Pornin, Bart Preneel, Christian Rechberger, Vincent Rijmen, Matt Robshaw, Andy Rupp, Martin Schläffer, Nigel Smart, Serge Vaudenay, Fré Vercauteren, and Michael Ward. ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012). Technical report, European Network of Excellence in Cryptology II, September 2012. http://www.ecrypt.eu.org/ecrypt2/documents/D.SPA.20.pdf.

[32] Alex Balducci, Sean Devlin, and Tom Ritte. Open Crypto Audit Project.TrueCrypt. Technical report, NCC Group, 2015. https://opencryptoaudit.org/reports/TrueCrypt_Phase_II_NCC_OCAP_final.pdf.

[33] Gregory V. Bard. The Vulnerability of SSL to Chosen Plaintext Attack. Cryptology ePrint Archive, Report 2004/111, 2004. http://eprint.iacr.org/.

[34] Elaine Barker, Don Johnson, and Miles Smid. Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. Technical report, NIST, 2007. NIST Special Publication 800-56A, http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf.

[35] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 139–155. Springer, 2000.

[36] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer Berlin Heidelberg, 2006.

[37] Jens Bender, Marc Fischlin, and Dennis Kügler. Security analysis of the PACE key-agreement protocol. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna, editors, *Information Security, 12th International Conference, ISC 2009, Pisa, Italy, September 7-9, 2009. Proceedings*, volume 5735 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 2009.

[38] Daniel J. Bernstein. Irrelevant patents on elliptic-curve cryptography. http://cr.yp.to/ecdh/patents.html.

[39] Daniel J. Bernstein. Curves, coordinates, and computations, 2014. https://www.ietf.org/mail-archive/web/cfrg/current/msg04816.html.

[40] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In *Eurocrypt 2015*. Springer, 2015. ilmumas.

[41] Daniel J. Bernstein and Tanja Lange. SafeCurves: choosing safe curves for elliptic-curve cryptography. http://safecurves.cr.yp.to.

[42] K. Bhargavan, A. Delignat-Lavaud, A. Pironti, A. Langley, and M. Ray. Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension. Internet draft (expires 16.05.2015), https://tools.ietf.org/html/draft-ietf-tls-session-hash-03.

[43] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Alfredo Pironti, and Pierre-Yves Strub. Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 98–113. IEEE Computer Society, 2014.

[44] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. XMSS – A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. In *Post-Quantum Cryptography*, number 7071 in Lecture Notes in Computer Science, pages 117–129. Springer Berlin Heidelberg, 2011.

[45] Ahto Buldas, Risto Laanoja, and Ahto Truu. Efficient Implementation of Keyless Signatures with Hash Sequence Authentication. Cryptology ePrint Archive, Report 2014/689, 2014. http://eprint.iacr.org/.

[46] Ahto Buldas, Risto Laanoja, and Ahto Truu. Efficient Quantum-Immune Keyless Signatures with Identity. Cryptology ePrint Archive, Report 2014/321, 2014. http://eprint.iacr.org/.

[47] Ahto Buldas, Risto Laanoja, and Ahto Truu. Security Proofs for the BLT Signature Scheme. Cryptology ePrint Archive, Report 2014/696, 2014. http://eprint.iacr.org/.

[48] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2010.

[49] Francisco Corella. NIST Omits Encryption Requirement for Derived Credentials. http://pomcor.com/2015/02/11/nist-omits-encryption-requirement-for-derived-credentials/, veebruar 2015.

[50] Francisco Corella and Karen Lewison. An Example of a Derived Credentials Architecture. Technical report, Pomcor, 2014. http://pomcor.com/techreports/DerivedCredentialsExample.pdf.

[51] Mike Czumak. Exploiting MS14-066 / CVE-2014-6321 (aka "Winshock"), November 2014. http://www.securitysift.com/exploiting-ms14-066-cve-2014-6321-aka-winshock/.

[52] Özgür Dagdelen and Marc Fischlin. Security analysis of the extended access control protocol for machine readable travel documents. In Mike Burmester, Gene Tsudik, SpyrosŠ. Magliveras, and Ivana Ilic, editors, *Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*, volume 6531 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2010.

[53] Özgür Dagdelen, Marc Fischlin, Tommaso Gagliardoni, Giorgia Azzurra Marson, Arno Mittelbach, and Cristina Onete. A Cryptographic Analysis of OPACITY. In Jason

Crampton, Sushil Jajodia, and Keith Mayes, editors, *Computer Security — ESORICS 2013*, volume 8134 of *Lecture Notes in Computer Science*, pages 345–362. Springer Berlin Heidelberg, 2013.

[54] Bert den Boer and Antoon Bosselaers. Collisions for the compression function of MD5. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 293–304. Springer Berlin Heidelberg, 1994.

[55] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol, Version 1.2, 2008. IETF RFC5246, http://tools.ietf.org/html/rfc5246.

[56] Jintai Ding and Dieter Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175. Springer Berlin Heidelberg, 2005.

[57] Hans Dobbertin. Cryptanalysis of MD5 Compress, May 1996. Rump session presentation at Eurocrypt 1996.

[58] Michael Düll, Björn Haase, Gesine Hinterwälder, Michael Hutter, Christof Paar, Ana Helena Sánchez, and Peter Schwabe. High-speed Curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. Cryptology ePrint Archive, Report 2015/343, 2015. http://eprint.iacr.org/.

[59] Thai Duong and Juliano Rizzo. Here Come The ⊕ Ninjas. http://packetstormsecurity.com/files/105499/Browser-Exploit-Against-SSL-TLS.html, May 13th 2011.

[60] E.A.Grechnikov. Collisions for 72-step and 73-step SHA-1: Improvements in the Method of Characteristics. Cryptology ePrint Archive, Report 2010/413, 2010. http://eprint.iacr.org/.

[61] Federation of Finnish Financial Service. TUPAS Identification Service for Service Providers, versioon 2.3c, jaanuar 2011. http://www.fkl.fi/en/themes/e-services/tupas/Pages/default.aspx.

[62] Federation of Finnish Financial Service. TUPAS Identification Service for Service Providers, versioon 2.4, December 2013. https://www.fkl.fi/teemasivut/sahkoinen_asiointi/tupas/Sivut/default.aspx.

[63] Hildegard Ferraiolo, David Cooper, Salvatore Francomacaro, Andrew Regenscheid, Jason Mohler, Sarbari Gupta, and William Burr. Guidelines for Derived Personal Identity Verification (PIV) Credentials, detsember 2014. NIST Special Publication 800-157, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-157.pdf.

[64] Hildegard Ferraiolo, David Cooper, Salvatore Francomacaro, Andrew Regenscheid, Jason Mohler, Sarbari Gupta, and William Burr. Guidelines for Derived Personal Identity Verification (PIV) Credentials, märts 2014. Draft NIST Special Publication 800-157, http://csrc.nist.gov/publications/drafts/800-157/sp800_157_draft.pdf.

[65] A. Freier, P. Karlton, and P. Kocher. The Secure Sockets Layer (SSL) Protocol Version 3.0. IETF RFC6101, http://tools.ietf.org/html/rfc6101.

[66] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. http://crypto.stanford.edu/craig.

[67] GlobalPlatform Card Specifications. http://www.globalplatform.org/specificationscard.asp.

[68] D. Harkins. Brainpool Elliptic Curves for the Internet Key Exchange (IKE) Group Description Registry, 2013. IETF RFC6932, https://tools.ietf.org/html/rfc6932.

[69] Koh Ho Kiat and Lee Yong Run. Analysis of OPACITY and PLAID Protocols for Contactless Smart Cards. Master's thesis, Naval Postgraduate School, September 2012. http://hdl.handle.net/10945/17385.

[70] Masashi Kikuchi. How I discovered CCS Injection Vulnerability (CVE-2014-0224), June 2014. http://ccsinjection.lepidum.co.jp/blog/2014-06-05/CCS-Injection-en/index.html.

[71] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 206—222. Springer Berlin Heidelberg, 1999.

[72] Neal Koblitz. *Algebraic Aspects of Cryptography*. Number 3 in Algorithms and Computation in Mathematics. Springer, 1998.

[73] Adam Langley. Apple's SSL/TLS bug, February 2014. https://www.imperialviolet.org/2014/02/22/applebug.html.

[74] Adam Langley. Early ChangeCipherSpec Attack, June 2014. https://www.imperialviolet.org/2014/06/05/earlyccs.html.

[75] Adam Langley. The POODLE bites again, December 2014. https://www.imperialviolet.org/2014/12/08/poodleagain.html.

[76] Eric Le Saint, Dom Fedronic, and Steven Liu. Open Protocol for Access Control Identification and Ticketing with privacY: Specifications, 15. juuli 2011. Versioon 3.7, http://www.smartcardalliance.org/resources/pdf/OPACITY_Protocol_3.7.pdf.

[77] Eric F. Le Saint and Dominique Louis Joseph Fedronic. Open protocol for authentication and key establishment with privacy, 9. juuli 2010. USA patenditaotlus nr. 20120144193, http://www.freepatentsonline.com/y2012/0144193.html.

[78] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[79] M. Lochter and J. Merkle. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639, http://www.ietf.org/rfc/rfc5639.txt.

[80] Hans Lõugas. Läti tudeng leidis Eesti pangalinkidest turvaaugu. Eesti Päevaleht, 27. september 2012. http://www.epl.ee/news/eesti/lati-tudeng-leidis-eesti-pangalinkidest-turvaaugu.d?id=65024204.

[81] Nikos Mavrogiannopoulos. gnutls 3.2.12 / GNUTLS-SA-2014-2. E-mail to the gnutls-devel@lists.gnutls.org mailing list, March 2014. http://article.gmane.org/gmane.comp.encryption.gpg.gnutls.devel/7341.

[82] R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. The Deep Space Network Progress Report, DSN PR 42-44, pages 114–116, January and February 1978.

[83] J. Merkle and M. Lochter. Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS), 2013. IETF RFC7027, https://tools.ietf.org/html/rfc7027.

[84] J. Merkle and M. Lochter. Using the Elliptic Curve Cryptography (ECC) Brainpool Curves for the Internet Key Exchange Protocol Version 2 (IKEv2), 2013. IETF RFC6954, http://tools.ietf.org/html/rfc6954.

[85] Bodo Möller, Thai Duong, and Krzysztof Kotowicz. This POODLE Bites: Exploiting the SSL 3.0 Fallback, September 2014. https://www.openssl.org/~bodo/ssl-poodle.pdf.

[86] Randall Munroe. Heartbleed Explanation, 2014. http://xkcd.com/1354.

[87] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, (15):159–166, 1986.

[88] Arnis Paršovs. Security Analysis of Internet Bank Authentication Protocols and their Implementations. Master's thesis, Tallinna Tehnikaülikool, 2012.

[89] Jacques Patarin. Asymmetric Cryptography with a Hidden Monomial. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 45–60. Springer Berlin Heidelberg, 1996.

[90] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *Eurocrypt '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 1996.

[91] Peeter Puusemp. *Üldalgebra alused*. TTÜ kirjastus, 2012.

[92] Kaido Raiend. SEB töötõndist, 2013. Ettekanne SK aastakonverentsil, https://sk.ee/upload/files/AK2013_Kaido%20Raiend_SEB%20tootoendist.pdf.

[93] Eric Rescorla. TLS 1.3, 2015. http://www.realworldcrypto.com/rwc2015/program-2/RWC-2015-Rescorla-TLS.pdf.

[94] Vincent Rijmen and Elisabeth Oswald. Update on SHA-1. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 58–71. Springer Berlin Heidelberg, 2005.

[95] Lara Schmid. Improving the ISO/IEC 11770 standard, 4. september 2013. Bakalaureusetöö, ETH Zürich.

[96] R. Seggelmann, M. Tuexen, and M. Williams. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension. IETF RFC6520, http://tools.ietf.org/html/rfc6520.

[97] AS Sertifitseerimiskeskus. DigiDocService spetsifikatsioon, detsember 2014. https://sk.ee/upload/files/DigiDocService_spec_est.pdf.

[98] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.

[99] Joseph H. Silverman. An Introduction to the Theory of Elliptic Curves. http://www.math.brown.edu/~jhs/Presentations/WyomingEllipticCurve.pdf, juuli 2006.

[100] Nigel P. Smart, Vincent Rijmen, Benedikt Gierlichs, Kenneth G. Paterson, Martijn Stam, Bogdan Warinschi, and Gaven Watson. Algorithms, key size and parameters report – 2014. Technical report, ENISA, 2014. http://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-size-and-parameters-report-2014/.

[101] Marc Stevens. New collision attacks on SHA-1 based on optimal joint local-collision analysis. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 245–261. Springer, 2013.

[102] Marc Stevens, Arjen Lenstra, and Benne de Weger. Chosen-Prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 1–22. Springer Berlin Heidelberg, 2007.

[103] Marc Stevens, Arjen K. Lenstra, and Benne De Weger. Chosen-prefix Collisions for MD5 and Applications. *Int. J. Appl. Cryptol.*, 2(4):322–359, July 2012.

[104] Marc Stevens, Alexander Sotirov, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Osvik, and Benne de Weger. Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 55–69. Springer Berlin / Heidelberg, 2009.

[105] S. Turner, D. Brown, K. Yiu, R. Housley, and T. Polk. Elliptic Curve Cryptography Subject Public Key Information. IETF RFC5480, https://tools.ietf.org/html/rfc5480.

[106] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the Hash Functions MD4 and RIPEMD. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *LNCS*, pages 1–18. Springer, 2005.

[107] Xiaoyun Wang, Andrew C Yao, and Frances Yao. Cryptanalysis on SHA-1, 2005. NIST cryptographic hash workshop, http://csrc.nist.gov/groups/ST/hash/documents/Wang_SHA1-New-Result.pdf.

[108] Xiaoyun Wnag, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer Berlin Heidelberg, 2005.

[109] Meggie Woodfield. Patch tuesday: Winshock vulnerability, November 2014. https://blog.digicert.com/winshock-vulnerability/.

A Elliptkõverate riistvaralise toe küsimustik

Kuna osa riistvaratarnijatest edastas küsimustiku tootjatele, vormistati küsimustik inglise keeles.

1. Which elliptic curve primitives are supported on your platform? Please list specific curves - NIST P-* curves, which ones? Brainpool curves, which ones? Bernstein curves (Ed25519, Curve25519)? Any other specific curves?
2. Is it possible to get some performance indicators (e.g. signatures per second) for these curves? Is there some documentation available?
3. It is possible that your platform supports not just some specific curves, but a wider class of them (e.g. all curves in Weierstraß short form). Is this the case? If so, then how much flexibility is there in specifying a particular curve in the class, i.e. which parameters can be given freely to the application? What is the flexibility/performance trade-off?
4. How complicated is it to implement a new curve on your platform? How big would be the performance penalty? E.g. in principle it is possible to implement Curve25519 in Java, but it would probably be prohibitively slow and it would be very difficult to protect the key. However, designing new hardware also has a high price. How high exactly?