TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Pavel Kargin 164212IAPB

EESTI ELEKTROONILISE DOKUMENDI TEHNILISTELE NÕUETELE VASTAVUSE TESTIMINE

Bakalaureuse lõputöö

Juhendaja: Juhan-Peep Ernits, PhD

Andrei Kargin
MSc

Tallinn 2019

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Pavel Kargin

21. mai 2019. a.

Annotatsioon

Käesolevas bakalaureusetöös kirjeldatakse Java-kaartide tööpõhimõtteid ja nende kasutamist tänapäevaste elektrooniliste ID-kaartidena. Esitatakse ülevaade kiipkaartide standardite kohta ja nende kasutamisest Eesti ID-kaartides. Selgitatakse, kuidas tuleb lugeda informatsiooni kaardilt ja selle vastavuse testimine spetsifikatsioonis toodud andmete põhjal.

Töö aluseks on võetud GlobalPlatformi spetsifikatsioon ja Eesti ID-kaardi tehniline kirjeldus arendajatele. Töös demonstreeritakse kui vajalik on kaartidega suhtluse testimine ja kuidas aitab JUnit testidel põhinev testimiskeskkond lihtsustada tulemuste kontrolli. Antud töö lõpptulemuseks on valminud keskkond, mis loeb maha kaardilt konkreetsed andmed ja võrdleb neid malliga. Lisaks on ära toodud töö käigus ilmunud probleemide ja lahenduste nimekiri.

Loodud tarkvara loeb maha kaardilt personaalsed andmed, autentimis- ja signeerimisandmed, CPLC ning ATR-i. ATR-i kontrollitakse kohe alguses ja selle põhjal ka otsustatakse, mis kaardiga on tegu, kui tegemist on tundmatu ATR-iga, siis kaart peetakse vigaseks. CPLC puhul kontrollitakse kogu sisu, nii võtmesõnade olemasolu kui ka nende väärtusi. Ülejäänud andmete puhul vaadatakse ainult võtmesõnade olemasolu, kuna andmed võivad igal kaardil erineda.

Töö tulemusel olid leitud eripärad, mis võivad olla kasulikud tarkvaraarendajatele. Nende hulgaks sertifikaatide lugemisel tuleva vastuse pikkus. See ei ole maksimaalne lahtri suurus, vaid tagastab andmeid väiksemas koguses. Lisaks tingimus, et CPLC andmed peab lugema enne, kui saadetakse kaardile mingi GET DATA päring. Arvestada tuleb ka PIN-ide saatmise loogika muutmisega, nimelt andmete lahter APDU käsus on fikseeritud suurusega ja kui PIN on väiksem, siis tühjades lahtrites peab olema väärtus 0xFF.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 6 peatükki, 12 joonist, 3 tabelit.

Abstract

Testing the Compliance of the Estonian Electronic Document to the Technical Specification

Bachelor thesis describes the principles of Java card work and their use in today's electronic ID cards. An overview of chip card standards and their use in Estonian ID cards is provided. Explains how to read information from the card and test its compliance with the data in the specification. The work is based on the GlobalPlatform specification and the technical description of the Estonian ID card for developers. The work concludes with the need for software that interacts with cards and how the JUnit test-based environment helps to simplify the control of results. The end result of this work is a completed environment that reads out the specific data on the map and compares them with the template. In addition, a list of problems and solutions that appeared during the work.

The software reads personal data, authentication and signature data from the card, CPLC and ATR. The ATR is checked at the outset and in the case of an unknown ATR the card is considered defective. CPLC checks all content. For the rest of the data, only the presence of keywords is reviewed, as the data may differ on each card. During testing were found useful information for software developers. Length of the response from reading the certificates returns less data than expected. In addition, the condition that CPLC data must be read before sending a GET DATA command to the card. Also sending PINs logic is different, namely the data cell in the APDU command is fixed in size and if the PIN is smaller, the empty cells must have a value of 0xFF.

The work consists of six sections. The first describes the technical capabilities of Java cards and their use in Estonian ID cards. In addition, a brief overview of Estonia's two ID card versions. The second and third focus on the goals of the given work and how they are planned to be achieved. The fourth and fifth logic of descriptive work and test creation. The latter talks about software plans for the future.

The thesis is in Estonian and contains 27 pages of text, 6 chapters, 12 figures, 3 tables.

Lühendite ja mõistete sõnastik

APDU	Application protocol data unit ehk aplikatsiooni protokolli andme ühik
ATR	Answer To Reset ehk vastus täiskäivitamisele
PKI	Public Key Infrastructure
eID	Elektrooniline indentiteet
CRL URL	Certificate revocation list Uniform Resource Locator
PIN	Personal identification number ehk isiklik tuvastamisnumber
PUK	Personal unblocking code ehk personaalne avamis kood
NFC	Near Field Communication ehk lähiväljaside
KMA	Kodakondsus- ja Migratsiooniamet
PPA	Politsei- ja Piirivalveamet
CPLC	Card Production Life Cycle ehk kaardi tootmise elutsükkel
PD	Personal Data ehk personaalsed andmed
IEC	International Electrotechnical Commission

Sisukord

Autorideklaratsioon 2
Annotatsioon 3
Abstract Testing the Compliance of the Estonian Electronic Document to the Technical Specification 4
Lühendite ja mõistete sõnastik 5
Sisukord 6
Jooniste loetelu 8
Tabelite loetelu 9
1 Sissejuhatus 10
1.1 Kiipkaart Eestis 10
1.2 Eesti eID 11
1.2.1 Elektrooniline indentiteet 11
1.2.2 Gemalto Eesti ID-kaart 12
1.2.3 IDEMIA Eesti ID-kaart 13
1.3 Töö eesmärk 14
1.3.1 Ülesandepüstitus 14
2 Töö taust 16
2.1 Java-kaart 16
2.2 GlobalPlatform 17
3 Kaardi sisu lugemine 19
3.1 Kasutatud vahendid 19
3.2 Kasutatud tehnoloogiad 19
3.2.1 GlobalPlatfomPro 19
3.2.2 OpenSC 20
3.3 Töö käik 22
3.3.1 Tehnoloogiate uurimine 22
3.3.2 Implementeerimine 24

3.4 Protsessi läbipaistvaks muutmine 25
3.4.1 Kaardi vastuste käsitlemine 25
3.4.2 Logifailide loomine 26
3.5 Ettetulnud probleemid 27
4 Kaardi sisu testimine vastu ID kaardi profiili 30
4.1 Loetud andmete struktuur 30
4.2 Testide loomine 31
4.2.1 Ühiktestid 32
4.3 Vastavuse hindamine 33
5 Plaanid kasutamisel tulevikus 35
5.1 Testitav funktsionaalsus ja võimalikud edasiarendused 35
5.2 Lisanduv funktsionaalsus 36
6 Kokkuvõte 37
Kasutatud kirjandus 39
Lisa 42

Jooniste loetelu

Joonis 1 Uue ja vana ID kaardi kujundus 14
Joonis 2 GlobalPlatformi kaardi arhitektuur. [10] 18
Joonis 3 OpenSC vastuse näide 21
Joonis 4 APDU käskude erinevad elementide kombinatsioonid [5] 23
Joonis 5 Kaardi vastuse haldamine 25
Joonis 6 Kaardi vea käsitlemine, kui pole teada ATR 26
Joonis 7 Logifailis näidatud saadetud käsu kuju. 26
Joonis 8 Logifailis näidatud saadud vastuse kuju. 27
Joonis 9 Vana ja uue ID-kaardi struktuur 30
Joonis 10 Edukate testide tulemus 32
Joonis 11 Vale tulemuse puhul vea kuvamine 33
Joonis 12 APDU vastus katsel lugeda PersonalData NFC abil 36

Tabelite loetelu

Tabel 1 Saadetud kiibile APDU struktuur. 23
Tabel 2 Personaalsete andmete struktuur 31
Tabel 3 Näite andmed ja testkaardil testitavate andmete sisu 42

1 Sissejuhatus

Eestis on elektroonilise identiteedi kandjana kasutuses ID kaart, mis on tehniliselt Java-kaart. Käesolevas juhatame sisse Java-kaardi temaatika ja sellega seotud tehnoloogiad sisse, kirjeldame kaardi kiipide spetsifikatsiooni GlobalPlatformis ja anname nii uute kui ka vanade Eesti eID kaartide lühikese ülevaate koos elektroonilise indentiteedi kirjeldusega.

1.1 Kiipkaart Eestis

Esimese ID-kaardi loomisel 2000ndate alguses, oli kiipkaartide maailm erinev praegusest ning kiipkaarti saab ka nimetada aegunud lähenemiseks. Kogu protsess keerles ümber failisüsteemi abstraktsiooni, mis on kirjeldatud ISO/IEC 7816-4 standardis [15]. Kogu ISO/IEC 7816 on rahvusvaheline standard, mis on seotud elektroonilise identiteedi kandmist toetavate kaartidega. Selle standardi ISO/IEC 7816-4 osa on nimetatud Part 4: *Organization, security and commands for interchange*. Erinevad tootjad pakkusid enda kaarte, millel oli tootja loodud „operatsioonisüsteem“, millest mingi osa oli avalik ning standardikohane, eelnevalt mainitud standardi ja failisüsteemiga, kuid ka esines enda arendatud ebastandardseid osi.

Samal ajal standardiga 7816-4 ilmus ka PKCS#15 standard [16], RSA poolt loodud standard, mis pani paika, mis nimega failidesse ja kuidas kirjutada metainformatsiooni võtmete ja sertifikaatide kohta, et nende otsimine ja kasutamine oleks võimalikult universaalne. Praeguseks on PKCS#15 ISO perekonna liige, numbriga 7816-15 [15].

Algselt otsustati valida ID-kaardiks Micardo, mille peale kirjutati andmed PKCS#15 standardiga mitteühilduvalt. Selle kaardi operatsioonisüsteemi võimekus, käsud ja ülesehitus olid kirjutatud kohalikuks standardiks EVS 827:2004 [23]. EstEID kiibirakendus määras ära kiibi operatsioonisüsteemi käsustiku ning ID-kaardi vajaduste jaoks üles sätitud failisüsteemi paigaldust koos failide sisuga. Sellisel moel ehitatud kiipkaardi eripärad võimaldasid kiipide kaughaldust. See isetehtud osa ja võtmestki kiibil,

sai vaikimisi standardiks, mida PPA, KMA ja pangakontorid pidid arutlema, et väljastada PIN ümbrikke või genereerida võtmeid.

2010.-nda aasta paiku toimus kiipkaartide maailmas muudatus, mis tõi uue kaartide generatsiooni. Uuteks standarditeks muutusid Java-kaart [1] ja GlobalPlatform [2], mille ümber koondusid kõik olulisemad kaarditootjad. Nüüd ei räägita pigem enam failisüsteemist, vaid funktsionaalsetest rakendustest, millel on standardiseeritud liides nagu näiteks: PIV (Personal Identity Verification), passirakendus, PGP(Pretty Good Privacy), EMV(Europay, MasterCard and VISA) rakendus jne.

JavaCard paneb paika, kuidas neid rakendusi ehitatakse. Java alamhulk koos liidestega kiibil ja nende võimalustega nagu võtmete genereerimine, algoritmid jne. GlobalPlatform paneb paika, kuidas neid rakendusi turvaliselt kaardil käsitleda, näiteks kuidas laadida rakendusi kaardile või kustutada neid.

1.2 Eesti eID

ID-kaart (EstEID) on kohustuslik isikut tõendav dokument, mida saab kasutada elektroonselt digiallkirja andmise, isikutuvastamise ning andmete krüpteerimise funktsioonides. ID-kaardile on salvestatud kaks X.509-vormingus sertifikaati: 1) sertifikaat digitaalseks isikutuvastuseks ning andmete allkirjastamiseks ja krüpteerimiseks; 2) sertifikaat digitaalseks allkirjastamiseks, mille abil saab kaardiomanik anda kehtiva digitaalallkirja.

Enne 1. jaanuari 2007 välja antud ID-kaart kehtib 10 aastat ning sellel olevad sertifikaadid 3 aastat. Kehtivuse lõppemisel saab sertifikaate tasuta uuendada. Alates 1. jaanuarist 2007 välja antud ID-kaartidel kehtivad sertifikaadid sama kaua kui kaart ise ehk siis 5 aastat ning sertifikaate uuendama ei pea.[3]

1.2.1 Elektrooniline indentiteet

Elektrooniline identiteet on andmete kogum, mis seob elektroonilises keskkonnas isiku tema füüsilise identiteediga. Elektroonilise identiteedi toimimise alus on avaliku võtme infrastruktuur ehk PKI.

Eesti mõistes on eID elektrooniliseks kasutamiseks mõeldud isikut tõendav dokument, nende hulgas ka ID-kaart, mida väljastab Politsei- ja Piirivalveamet. Eesti eID kasutab nii

nimetatud turvalist andmekandjat, mis koosneb kiibist, selles talletatud sertifikaatidest ja võtmetest (PKI), mis tagavad toimingute turvalisuse. Sertifikaat on isikuga seotud elektrooniline dokument, mis sisaldab isikuandmeid ja tema tuvastamiseks vajalikku võtit.

PKI mudel põhineb kahel võtmel – salajane võti ja avalik võti. Nii nagu võtmete nimed juba ütlevad, siis salajane võti peab olema kaitstud ning seda peab saama kasutada vaid inimene, kellele see on väljastatud. Avalik võti jällegi on kättesaadav kõigile ning nende kahe võtme vahel on kindel seos. [4]

1.2.2 Gemalto Eesti ID-kaart

Eesti ID-kaart on esmane isikut tõendav dokument ehk isikutunnistus Eesti Vabariigis. Eestis hakati välja andma eID-kaarti alates 2002 aastast, mille tootjaks oli Gemalto. Selle abil omanik saab teostada isikutuvastust nii elektroonselt, kui ka kasutada tavalise dokumendina ning kasutada seda ka digitaalallkirjastamiseks. Iga Eestis elav isik, kes on vähemalt 15 aastat vana, peab omama ID-kaarti. See dokument on kehtiv kogu Euroopa Liidus ja ka Šveitsis. Hädaolukorra seaduse kohaselt on ID-kaardiga autentimine ja digiallkirjastamine elutähtis teenus.

Üks tähtsatest teenustest, mis vajab ID kaardi kasutamist on näiteks e-hääletamine. Selleks, et hääletada elektroonselt, kasutaja peab tuvastama ennast kasutades ID-kaarti või mobiil-ID-d. Esimene kord e-hääletamist kasutati 2005 aastal kohalikel omavlitsuse valimistel. See näide näitab, kui tähtsal kohal on e-identiteet Eesti Vabariigis ja kui tähtis on selle tuvastamise võimalus ning turvalisus.

Kaardi peal hoitakse personaalsed andmed ning ka sertifikaadid, mille abil saab kindlaks teha isiku identiteeti. Osa informatsioonist on peidetud PIN1 ja PIN2 taha. Nende abil saab maha lugeda kaardilt näiteks autentimise ja signeerimise privaat võtmed ning ka see on vajalik operatsioonide kinnitamiseks interneti keskkonnas nagu näiteks pangas.

2017. aasta septembris avastati kaartide kiipides olevas tarkvaras turvaviga, mis põhjustas vajaduse uuendada paljude inimeste sertifikaadid kaardil. See tekitas palju probleeme, kui selle kriisiga saadi hakkama.

1.2.3 IDEMIA Eesti ID-kaart

Alates 2018. aasta detsembrist tulid kasutusele uued ID-kaartid. Uuendamise põhjuseks on senise tootja Gemalto asendamine uue partneriga IDEMIA-ga Politsei- ja Piirivalveamet poolt. Põhjusteks võib tuua välja mitu erinevat asjaolu. Leping Gemaltoga sai läbi ning korraldati uus konkurss, et leida tootja järgmisele kaardi partiile. Konkursi käigus otsustati, et Gemalto ei saa pakkuda teenust samadel tingimustel nagu IDEMIA. IDEAMIA poolt pakutud toodangu arve ja halduse hind, koos pakutud teenusega oli parematel tingimustel, kui ülejäänud konkursil osalejatel. [6] Lisaks ka asjaolu, et Gemalto kaotas oma usaldusväärtuse pärast intsidenti, mis tuli teatavaks 2018. aasta oktoobris. Nimelt Tartu Ülikooli teadlased ja AS Cybernetica eksperdid avastasid, et Gemalto ei täitnud ühte tähtsatest tingimustest kaardi toodangul sõlmitud lepingu perioodil. Toodangu tingimuste järgi peavad kõik privaatsed võtmed olema toodetud kaardi kiibi peal mitte sellest väljas, kuid Gemalto ei täitnud seda tingimust aastatel 2011 kuni 2014. [7]

Uue kaardi tulekuga tuleb meeles pidada, et sellele lisandusid ka muudatused, mis peamiselt puudutavad isikuid või ettevõtteid, kes kasutavad aktiivselt kaarte isikutuvastamiseks või digitaalsel allkirjastamisel. Näiteks on muutunud PersonalData faili struktuur ja nende poole pöördumise käsud, mille pärast ei saa uusi kaarte kasutada näiteks osades kliendikaardisüsteemides samal viisil nagu varem. Muutmata jäi ainult üks ATR. Sertifikaatidest kaob ära CRL URL, mis võib mõjutada süsteeme, milles autentimisel kasutatakse sertifikaadi kehtivuse kontrolliks tühistusnimekirju. Uuel kaardil on lisandunud ka kontaktvaba liides, mis tulevikus võimaldab suurendada kaardi kasutamise võimalusi. Uued (Idemia) kaardid on hetkel paralleelselt kasutusel juba olemasolevate (Gemalto) kaartidega ning ülemineku periood uutele kaartidele on 5 aastat, mille jooksul jääbki ID-kaardi ökosüsteemi kasutusse kaks erinevat kaardi varianti.

Uue ID-kaardi visuaalne pool erineb eelmisest samuti. Uuele kaardile lisati mõned uued turvaelemendid, muudeti informatsiooni paigutust ning pilt muudeti värviliseks. Lisaks on muudetud ka peidetud turvaelemendid, mis on võimalik näha ainult UV-valgusega. Mõned kaartide erinevused välimuses on näha joonisel Joonis 2.

Joonis 1 Uue ja vana ID kaardi kujundus

1.3 Töö eesmärk

Uute ID kaartide arenduse protsessis loodi mitu versiooni kaartidest, mida oli vaja testida, et kontrollida, kas need vastavad nõuetele ning, et informatsioon nende peal oleks vastavuses spetsifikatsiooniga. Kuna eelnevalt toimus sisu kontrollimise protsess käsitsi ning kõikide APDU käskude sisestamine toimus käsitsi ja vajas manuaalset tulemuste kontrolli, siis suure kaartide koguse tõttu tekkis vajadus teha protsess rohkem automaatseks. Automatiseerimise all on siinkohal mõeldud seda, et programm loeb kaardilt maha kaardi informatsiooni ning testide abil võrdleb olemasoleva profiiliga.

Käesoleva töö eesmärk on luua keskkond, mille abil saab senisest lihtsamalt kontrollida nii vana kui ka uue ID kaardi vastavust profiilile. Testiotsuse tegemine peab olema automatiseeritud. Tulemuseks vormistatud dokument loetud info vastavuse kirjeldusega.

Probleemiks on uue ja vana ID kaardi testimiskeskkondade lahusus. Lahenduseks on tarkvara, mis võimaldab kontrollida mõlema kaardi vastavust nõuetes spetsifitseeritud profiilile. Loodav tulemus kergendab kaartide spetsifikatsiooni vastavuse kontrolli.

1.3.1 Ülesandepüstitus

Käesolevas töös otsime vastust peamisele küsimusele: kuidas luua tarkvara, mis võimaldaks mugavalt koguda testandmeid Java-kaadi tehnoloogial põhinevalt Eesti eID kaardilt?

Selle küsimusele vastamiseks tuleb leida vastus järgmistele alamküsimistele:

- Kuidas töötab Java-kaart ja millisel moel on võimalik sellega suhelda?

- Kuidas paikneb informatsioon kaardil ja mis käskudega on võimalik selleni jõuda?

- Kuidas hinnata saadud tulemuste põhjal kaardi vastavust spetsifikatsioonile?

Soovitud tulemust on plaanis saavutada, uurides kaartide erinevust erinevates arengu faasides. Lisaks ka testides vastu vigast kaarti, mis alati tagastab valet informatsiooni või infot, mis ei ole kooskõlas olemasoleva eID kaardi spetsifikatsiooniga.

Eesmärkideni jõudmiseks viidi läbi uuring vana ning uue kaardi mittetäielikest dokumentatsioonidest ning testiti erinevaid keskkondi, mis on mõeldud kaardiga suhtlemiseks.

2 Töö taust

Eelnevalt olid mainitud tehnoloogiad nagu Java-kaart ja GlobalPlatform-i standart. Nende spetsifikatsioonide tundmine ja mõistmine on vajalik, et saada aru, kuidas peab lähenema kiipkaartide testimisele.

2.1 Java-kaart

Java-kaart on tarkvaratehnoloogia, mis võimaldab käivitada Java-põhiseid rakendusi kiipkaartidel. Java-kaart on väikseim Java-platvorm, mis on mõeldud kasutamiseks sisseehitatud seadmetele. Java-kaart annab kasutajale võimaluse programmeerida seadmeid ja muuta need rakendusespetsiifiliseks. Seda tehnoloogiat kasutatakse laialdaselt SIM-kaartides ja pangakaartides. Esimest Java-kaarti näitas rahvale 1996. aastal kaardi tootja Schlumberger, mis hiljem liideti Gemploga, et luua Gemalto. Java-kaardi tooted põhinevad Java-kaardi platvormi spetsifikatsioonidel, mille on välja töötanud Sun Microsystems, mis hiljem liideti Oracliga. Paljud Java-kaardi tehnoloogiaga tooted tuginevad GlobalPlatformi spetsifikatsioonidel, et muuta rakenduste kasutamist kaardil turvalisemaks. GlobalPlatformi spetsifikatsioon sätestab sellised operatsioonid nagu näiteks: rakenduse allalaadimine, installimine, personaliseerimine, kustutamine. Java-kaardi tehnoloogia peamisteks eesmärkideks on, et tehnoloogia oleks kergelt kaasaskantav ja turvaline. [1]

Java-kaardi tehnoloogia võimaldab kiipkaartidel ja teistel väga piiratud mäluga seadmetel käivitada Java-tehnoloogiat kasutavaid väikeseid rakendusi, mida nimetatakse appletideks. See annab kiipkaartide tootjatele turvalise ja mugava rakendusplatvormi, mis võimaldab salvestada ja uuendada mitut rakendust ühes seadmes. Java-kaardi tehnoloogia on kooskõlas olemasolevate kiipkaartide standarditega.

Tehnoloogia võimaldab arendajatel ehitada, testida ja kasutada rakendusi ja teenuseid kiiresti ja turvaliselt. See kiirendatud protsess vähendab arenduskulusid, suurendab toodete mitmekesisust. Lisaks muudab Java-kaardi tehnoloogia kergemaks turvatähiste integreerimise suurematesse Java tehnoloogial baseeruvatesse tarkvaralahendustesse.

Java-kaardi baitkood, mida käivitab Java-kaardi virtuaalmasin on funktsionaalne alamhulk Java 2 virtuaalmasinast baitkoodi tasemel, kuid erinevad kodeerimisega, mis optimeerib andmete mahtu. Java-kaardi applet kasutab tavaliselt vähem bait koodi, kui Java lähtekoodist kompileeritud Java applet. See aitab säästa ruumi väiksema mäluga objektidel nagu kiipkaardid. Selle tehnoloogia piiranguks on mõnede Java tehnoloogiate puudumine ja suuruse piirangud. [1]

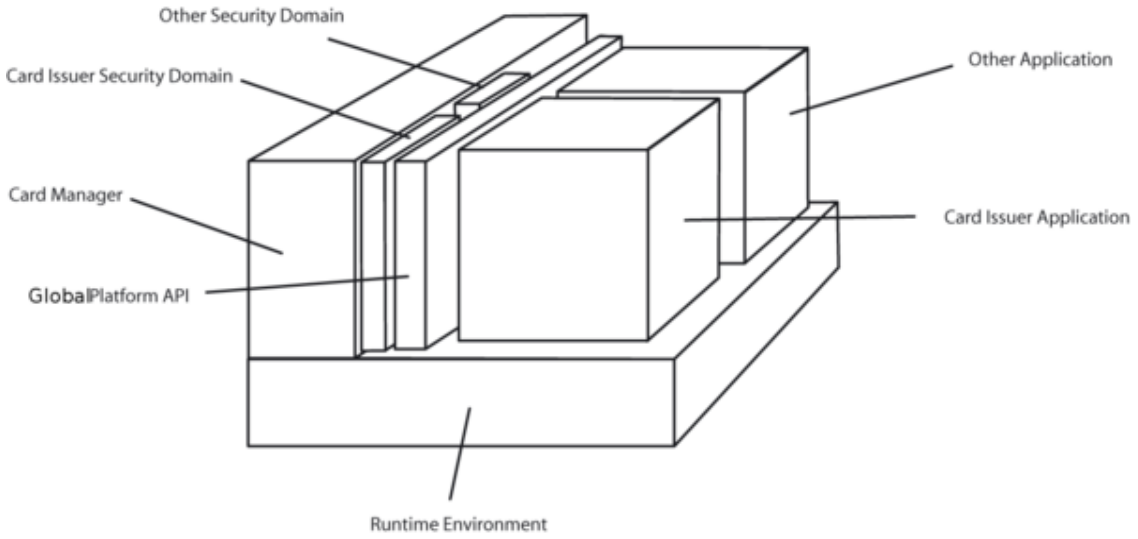
2.2 GlobalPlatform

GlobalPlatform on mittetulundusühing, mis loob ja avaldab turvalise kiiptehnoloogia spetsifikatsioonid.

GlobalPlatformi tehnoloogia muudab teenuseosutajatele ja seadme tootjatele teenuste kasutuselevõtu sujuvaks ja turvaliseks, sõltumata turust või seadme tüübist. Tänu sellele koostööle on võimalik muuta massilise turvalise digitaalteenuste turustamise võimalikuks. See spetsifikatsiooni pakub turvalise arhitektuuri koos paindlikkusega, et toetada erinevaid turunõudeid sama tüüpi turvaliste komponentide jaoks.

Tehnilised dokumendid pärinevad koostöö käigus loodud ühingute järgmistest kommiteedest: Turvaline Element (Secure Element (SE)), Usaldatud Käivitamis Keskkond (Trusted Execution Environment (TEE)) ja Usaldatud Platvormi Teenus (Trusted Platform Services (TPS)). Enne käivitamist läbivad dokumendid põhjaliku läbivaatamisprotsessi, mis sisaldab komisjoni, liikmete ja avalikku ülevaadet.[2]

GlobalPlatfom paneb paika spetsifikatsioonid kiipkaartidele, mille sisuks on kaardi kasutamine ja arhitektuur. (Joonis 1) näitab GlobalPlatformi kaardi arhitektuuri.



Joonis 2 GlobalPlatformi kaardi arhitektuur. [10]

Runtime enviroment ehk käitusaeg keskkond on mõeldud käivitumiseks iga turvalise mitme rakenduse kaardi käitusaeg keskkonnal. Eesti ID-kaardi puhul tegemist on Java Card Runtime Enviroment-iga. Käitusaeg keskkond vastutab selle eest, et riistavara neutraalne API aplikatsioonidel, turvaline andmete hoidla ning käivitamis ruum aplikatsioonidel oleks turvaline. See kindlustab iga aplikatsiooni koodile ja andmetele nende lahusust ja turvalisust teistest aplikatsioonidest kaardil. Lisaks see osa vastutab selle eest, et saaks suhelda kaardiga ja kaardi väliste vahenditega.

Security Domain ehk turvalisuse domeen võimaldab kontrollida kaardil kindlat informatsiooni turva riskideta. Nende hulgaks on ka kaardil olevad võtmed. Seda osa võib ka nimetada aplikatsiooniks, millel on erinevad õiguste komplektid.

GlobalPlatform API annab aplikatsioonidel kaardil ligipääsu teatud manageerimis funktsioonidele kaardil ja võimalus autentida kommunikatsiooni partnerit, kasutades turvalist kanalit koos sellega hetkel valitud turvalises domeenis.

Card Manager ehk kaardi haldur on GlobalPlatform kaardi keskne komponent. Iga teenus on käivitatud läbi selle ning lisaks see annab võimalust kasutada liidest, mida saab kasutada sisemiselt läbi GlobalPlatfom API või väliselt läbi APDU käskude. Lisaks selles on kaardi tootja turva domeen.

3 Kaardi sisu lugemine

3.1 Kasutatud vahendid

Kaardiga suhtlemiseks on vaja kasutada spetsiaalsed kaardilugejad, mis võimaldavad arvutil suhelda kaardil paikneva kiibiga. Kuna uue eID tulekuga lisandus sellele ka võimalus lugeda informatsiooni NFC tehnoloogia abil, siis antud töö raames on kasutusel kaks kaardilugejat. Esimeseks on OMNIKEY Smart Card Reader kontaktliidese kaudu suhtlemiseks ja ACS ACR122U PICC kontaktita liidese kaudu suhtlemiseks.

3.2 Kasutatud tehnoloogiad

Kuna Java-kaardi puhul on tegemist Java-l põhineva tehnoloogiaga, siis tarkvara arendamiseks on kasutatud Java 1.8 versiooni, kuna arendusprotsessi alustamisel see oli kõige stabiilsem versioon. Kasutatud IDE on Eclipse Photon, kus kaardiga suhtlemiseks on kasutusel javax.smartcardio[18] teek. See teek võimaldab Java API-l suhelda kiipkaartidega, kasutades standardit ISO / IEC [22] 7816-4 APDU-sid. See võimaldab Java rakendusel suhelda rakendusega kiibil, salvestada ning saada andmeid kaardilt.

3.2.1 GlobalPlatfomPro

Üheks mugavaks vahendiks tehtud tarkvara üle kontrollimiseks on GlobalPlatfomPro [8]. Selle lahendusega on võimalik suhelda kaartidega ning lugeda nendelt informatsiooni saates APDU käske vastavad Eesti eID ja GlobalPlatformi spetsifikatsioonidele. Antud lahendus on mugav ja kiire võimalus käsu toimimise kontrollimiseks, kuid ei võimalda kohe automaatselt lugeda kogu vajalikku infot.

GlobalPlatformPro on avatud lähtekoodiga töövahend , mis võimaldab toetatud kaartidele laadida ja kaartidel hallata aplette. Käsud saab täita käsurealt Java projektis. Siin on kasutatud DWIM2 lähenemine [8].

GlobalPlatformPro tarkvara kujutab endast kergelt kasutatavat ja kõrge taseme liidest, mis enamuse ajast lihtsalt töötab, on paindlik ja avatud lähtekoodiga vaba tarkvara, mis

on kättesaadav GitHub-is. Selle lahenduse töötas välja Martin Paljak, kes on Eestis väga tunnustatud Java-kaartide arenduse spetsialist.

Antud lahendus sobib erinevat tüüpi kaartide testimiseks. Nende hulgas järgmiste tootjate kaardid: Athena, Gemalto, Giesecke & Devrient, Infineon, NXP (JCOP), Morpho, Oberthur. Antud töö raames uue eID puhul on tegemist IDEMIA poolt tehtud kaardiga, kuid vana kaardi tootjaks on Gemalto. Üldiselt see liides sobib kõikidele uutele Java-kaartidele, mis suhtlevad GlobalPlatform standardi 2.1.1 või uuema kohaselt.

Liidese kasutamiseks on vaja teada võtmeid, mis on seotud selle kaardiga. Käsud, mis on sisestatud käsurealt, eeldavad vaikimisi testvõtmeid. Tavaliselt kaartidel on kindlad võtmed ning nende teadmiseks on vaja kontakteeruda kaardi väljastajaga. Võtmete muutmiseks tuleb kasutada käsku ***-key***. Mõnede kaartidel puhul tuleb ka täpsustada mitmekesisust käskudega ***-emv*** või ***-visa2***.

Üheks selle liidese võimaluseks on lugeda kaardiga seotud infot. Nende hulgas on kaardi tootja informatsioon, versioonid, võtmed, paigaldatud paketid ning info, millised on ka installitud. Selles töös kasutatud kaardi jaoks loetakse informatsioon kaardilt välja käsuga:

```
gp -dvli -key 2X…XC -kdf3 --sdaid A0000001510000
```

Käsu peale tagastatakse informatsioon kaardi osadest nagu näiteks: ATR, CPLC, kaardi andmed, võtmete üldine informatsioon ja laetud ning installitud apletid.

3.2.2 OpenSC

OpenSC annab ligipääsu mitmele teegile ja vahendile, mis on vajalikud tööks Java-kaartidega. Selle peamine rõhk on kaartidel, mis toetavad krüptograafilisi operatsioone ja nende kasutus turvalistes aplikatsioonides nagu näiteks autentimine, meilide krüpteerimine ja digitaalne allkiri. OpenSC implementeerib standardseid API-si JavaCard-idele nagu näiteks: PKCS#11 API, Windows' Smart Card Minidriver ja macOS Tokend. [9]

Samamoodi nagu ka GlobalPlatformPro, on OpenSC on täielikult vaba tarkvara, mida on võimalik kasutada kohe, kui selle kood on GitHubist alla laetud ja seadistatud.

Käesoleva töö raames on OpenSC-d kasutatud peamiselt uue eID kaardi andmete lugemiseks. Andmete lugemiseks selles keskkonnas tuleb teada kaardi struktuuri, kuna kogu suhtlemine toimub APDU5 käskude tasemel.

Peamiseks eeliseks OpenSC-l ma nimetaks, et on võimalik näha kogu suhtlemis protsessi kaartide vahel. CommandAPDU saatmisel näitab ka saadud responseAPDU-d, mis aitab mõista, mis kujul saadakse vastused kaardilt ja kergendab oma tarkvara loomist selle põhjal.

OpenSCd on võimalik kasutada uue eID personaalsete andmete ja autentimise ning signeerimis sertifikaatide lugemiseks. Personaalsete andmete lugemiseks tuleb kasutada järgmist käsku:

```
opensc-tool                    -v                    -s
00A4040010A000000077010800070000FE00000100 -s 00A4000C00  -
s 00A4010C02D003 -s 00B0000000   -s 00A4010C025000     -s
00A4010C025001 -s 00B0000000       -s 00A4010C025002 -s
00B0000000      -s 00A4010C025003 -s 00B0000000      -s
00A4010C025004 -s 00B0000000       -s 00A4010C025005 -s
00B0000000      -s 00A4010C025006 -s 00B0000000      -s
00A4010C025007 -s 00B0000000       -s 00A4010C025008 -s
00B0000000      -s 00A4010C025009 -s 00B0000000      -s
00A4010C02500A -s 00B0000000       -s 00A4010C02500B -s
00B0000000      -s 00A4010C02500C -s 00B0000000      -s
00A4010C02500D -s 00B0000000       -s 00A4010C02500E -s
00B0000000    -s 00A4010C02500F -s 00B0000000
```

Antud APDU käsud küsivad kaardil informatsiooni seotud andmetega, mis asuvad personaalsete andmete failis. Vastuseks tulevad read koos APDU käskudega ja saadud vastustega ja kui vastusega tulid andmed, siis ka nende tõlge ASCII kujule nagu joonisel (Joonis 3).

```
Received (SW1=0x90, SW2=0x00)
Sending: 00 B0 00 00 00
Received (SW1=0x90, SW2=0x00):
4A 41 41 4B 2D 4B 52 49 53 54 4A 41 4E 20 JAAK-KRISTJAN
```

Joonis 3 OpenSC vastuse näide

3.3 Töö käik

3.3.1 Tehnoloogiate uurimine

Kogu protsessi alustamiseks oli vaja algul tutvuda ISO 7816 [15] standardiga, et saada arusaam APDU käskude tüüpidest. Kuna kaartidega suhtlemisel on vaja saada arusaam, mis infot peab saatma kaardile ja mida sa saad vastu.

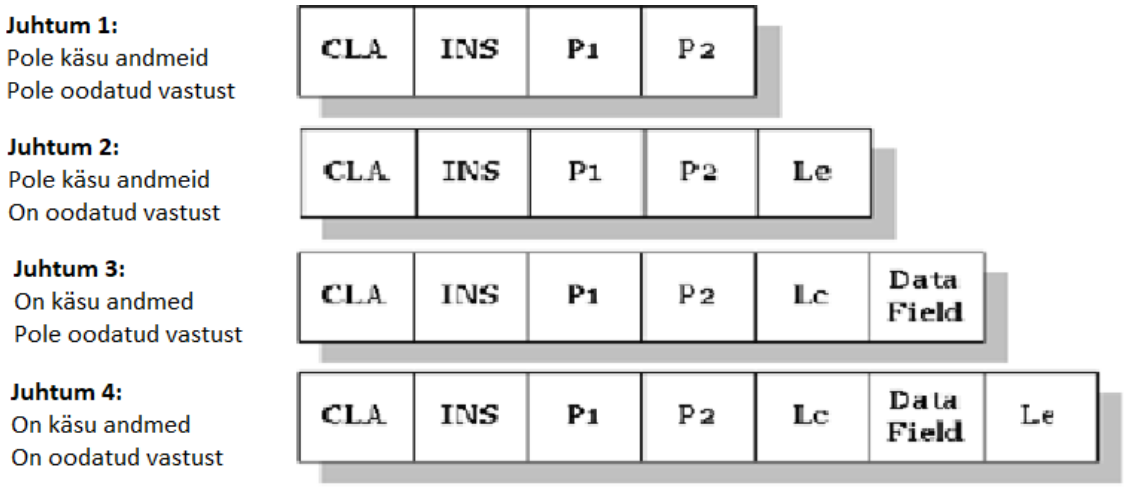
APDUsid on kahte tüüpi: käsk ja vastus. Iga käsk koosneb järgmistest osadest:

- *CLA* (klassi käsk),
- *INS* (käsu kood),
- *P1* (käsu parameeter 1),
- *P2* (käsu parameeter 2),
- L_c (baitide kogus andmete väljas, see väli jääb tühjaks kui pole andmete välja),
- Data Field (andmete väli, see väli pole kohustuslik) L_e (oodatava vastuse pikkus).

Standardi järgi peab kindlate käskude jaoks olema kindel *CLA, INS, P1 ja P2* väärtused. Peamised APDU käsud, mis kasutatakse selles töös on SELECET FILE, GET DATA, READ BINARY, VERIFY. Joonisel (Joonis 4) on näidatud erinevad variandid, mis kujul on võimalik saata käske kaardile. Selleks on neli võimalust.

Tabel 1 Saadetud kiibile APDU struktuur.

Tüüp	Nimi	Pikkus	Detailid
CLA	Klass	1 Bait	Käsu instruktsioon, kas käsk kasutab turvalist suhtlust või mitte.
INS	Instruktsioon	1 Bait	Käsu instruktsioon
P1	Parameeter 1	1 Bait	Instruktsiooni esimene parameeter
P2	Parameeter 2	1 Bait	Instruktsiooni teine parameeter
L_c	Pikkuse käks	0-3 Baiti	Käsu andmete pikkus
Data	Andmed	Lc Baidid	Käsu andmed (APDU request)
L_e	Eeldatud pikkus	0-3 Baiti	Eeldatud vastuse pikkus (APDU response)



Joonis 4 APDU käskude erinevad elementide kombinatsioonid

Vastuse APDU koosneb kolmest osast: Response (väli vastuse sisulise poolega), *SW1* ja *SW2*. Tavaliselt kui kõik on korras, siis *SW1* on 90 ja *SW2* 00 ehk vastus on 90 00. Kui käsk APDU ei eelda vastust ehk Le on tühi või puuduvad andmed tagastamiseks, siis vastuseks tuleb ainult vastuse kood, mis koosneb *SW1* ja *SW2* väärtustest.

Üheks tähtsaks kohaks töö alustamisel oli ka selgitada erinevust protokollide T=0 ja T=1 vahel ning nende kasutamise erinevus. Standardis ISO/IEC 7816-3 [22] on defineeritud

kaks andmevahetusstandardit kommunikatsiooniks kiipkaartide ja terminali vahel. Baidi põhine poolduplex andmeedastusprotokoll T=0 ja blokk-orienteeritud poolduplex protokoll T=1. Lühidalt saab öelda, et T=1 on turvalisem meetod andmete edestamiseks, kuna T=0 on spetsiifiliselt orienteeritud protokoll ja võimaldab korraga saata väga väikest andmehulka: umbes 300 baiti. See eest T=1 protokoll töötab koos vigade leidmisega (error detection EDC) bloki lõpus ning selle abil korraga edastatav andmemaht on umbes 1100 baiti.

Antud töö raames on vaja ka tutvuda ATR-iga. Gemalto kaardi variandil oli Eesti ID-kaardil kaks ATR-i esimene on külm, mis tuleb pärast kaardi esimest käivitamist, ja teine soe, esineb peale kaardi ühendamist. Uuel kaardil ei ole külma ja sooja, vaid on üks kindel ATR peale kaardi ühendamist lugejaga, kuid see on erinev kontakt ja kontaktvaba lugejaga ühendamisel.

3.3.2 Implementeerimine

Kuna tegemist on Java-kaartidega, siis ka tarkvara on loodud Java keeles. Tegevuse alustamisel oli kasutusel JDK 8 ja JRE 8, ja teek javax.smartcardio. Töö käigus jäi põhiplatvormiks Java 8, kuna üleminek versioonile 11 põhjustas probleemi javax.smartcardio teegiga.

Tarkvara eesmärgiks on lugeda informatsiooni nii vanalt kui ka uuelt ID-kaardilt. Kuna vanade kaartide muutmist enam kavas ei ole ja lähiima 5 aasta jooksul nende kasutamine lõpetatakse, siis töös on keskendatud rohkem uuemale kaardi versioonile.

Rakenduse arhitektuur tugineb *Singelton Class* põhimõttele. Kuna kaartide ATR erineb, siis just ATR põhjal otsustatakse, mis kaardiga on tegemist, kas uue Eesti ID-kaardiga, vana Eesti ID-kaardiga või mingi muu kiipkaardiga. Mingi muu kaardi puhul tagastatakse objektina tühi Optional, mis kasutamisel tagastab vea, et tegemist on tundmatu kaardiga.

Failide väljakutse kaardil toimub kõik eelnevalt loodud klassis. Failide valimiseks kasutatakse APDU käsku SELECT FILE, mille *INS* väärtuseks peab olema 0xA4. Seda käsku kasutatakse näiteks Master Faili, AWP või QSCD faili valimisel kiipkaardil. Lisaks sellele on vajalik käsk READ DATA, mille INS väärtuseks on 0xCA. See võimaldab lugeda selliseid andmeid nagu näiteks CPLC.

Vajalikud andmed lugemiseks oli otsustatud panna kõik erinevatesse objektidesse, et lihtsustada nende järgnev kasutamine, sealhulgas ka nende sisu kontrollimine. Igas andmetüübis on kasutusel peamisel APDU käsk Read Binary Transparent, mis on kujul 00B0000000, 00B0010000 … 00B0FE0000, 00B00FF000. See käsk eeldab, et saadetakse kõik andmed, mis on võimalik saada valitud lahtrist. Sertifikaatidel on lugemiseks mitu lahtrit ja nende puhul tuleb alati kontrollida, kas tuleb lugeda ka järgmist lahtrit.

3.4 Protsessi läbipaistvaks muutmine

Kuna kaardi lugemiseks edestatavad käsud ise ei tagasta, mis vastus tuli kaardilt, siis protsessi arusaadavamaks muutmiseks oli vaja kirjutada tugi Command- ja ResponseAPDU käskude näitamiseks konsoolis ja nende salvestamiseks eraldi faili. Vea ilmumisel või tulemuste võrdlemiseks on vajalik kaardi suhtlemise protsessi detailset ajalugu.

3.4.1 Kaardi vastuste käsitlemine

Iga kaardi vastuse saabumisel on vaja teha kindlaks, kas kaardilt laekus vajalik info. Selleks kergeim meetod oli luua funktsioon, mis kontrollib algul saadud vastuse *SW1* ja *SW2* koode. Vastuste näide lugemisel tulevate vastuste näide tehtud töös on joonisel (Joonis 5).

```
public String handleResponse() {
    switch (responseApduSW) {
    case 0x9000:
        logger.info(JCConverter.convertByteToHex(responseApdu.getData()));
        logger.info(JCConverter.convertByteToAscii(responseApdu.getData()));
        return "[Info]: Response OK. Code: 9000\n";
    case 0x6A81:
        return "[Error]: Function not supported. Code: 6A81\n";
    case 0x6A82:
        return "[Error]: File not found. Code: 6A82\n";
    case 0x6A88:
        return "[Error]: Referenced data not found. Code: 6A88\n";
    case 0x6282:
        return "[Warning]: End of file/record reached before reading Le bytes. Code: 6282\n";
    case 0x6B00:
        return "[Error]: Wrong parameter(s) P1-P2. Code: 6B00\n";
    case 0x6983:
        return "[Error]: Authentication method blocked. Code: 6983\n";
    case 0x63C1:
        return "[Warning]: Verify fail, 1 retrie left. Code: 63C1\n";
    case 0x63C2:
        return "[Warning]: Verify fail, 2 retries left. Code: 63C2\n";
    case 0x63C3:
        return "[Warning]: Verify fail, 3 retries left. Code: 63C3\n";
    case 0x6700:
        return "[Error]: Wrong length. Code: 6700\n";
    case 0x6982:
        return "[Error]: Security Conditions. Not Satisfied Code: 6982\n";
    default:
        String response = JCConverter.convertByteToHex(this.responseApdu.getBytes());
        String responseToWrite = response.length() > 4
                ? response.substring(response.length() - 4, response.length())
                : response;
        return "[Info]: Response NOT OK. Code: " + responseToWrite + "\n";
```

Joonis 5 Kaardi vastuse haldamine

Vastuse koodide saamisel mõnede puhul tagastatakse nende tähendus. Lisaks sellele, kui kood on 0x9000, siis tagastatakse ka andmed kuueteistkümnendsüsteemis ja tõlgitud ASCII kujul. Juhul kui vastus on koodiga, mida pole selles tabelis, siis tagastatakse ainult vea kood ilma lisaandmeteta.

Tarkvara on keskendunud just Eesti ID-kaartidele, nii et juhul kui klassi loomisel ei leita teadaolevatest ATR-idest õiget, siis tagastatava klassina kasutatakse ülemklassi. Iga meetodi puhul, siis tagastatakse veateade nagu on näidatud joonisel (Joonis 6).

```
/* (non-Javadoc)
 * @see main.java.org.javacard.JavaCardInfoInterface#openPersonalDataFile()
 */
@Override
public void openPersonalDataFile() throws CardException {
    getCard().disconnect(false);
    throw new CardException("Card must be specified");
}
```

Joonis 6 Kaardi vea käsitlemine, kui pole teada ATR

Lisa vigade juurde kuuluvad vead nagu kaardi puudumine või viga terminaliga. Nende ilmumisel tagastatakse ka vastav veateade konsooli ja salvestatakse logifaili.

3.4.2 Logifailide loomine

Töö tegemise käigus kaardiga suhtlemisel kerkisid esile erinevad vead. Nende hulgas oli ka juhuseid, kus saadeti valesid käske. Selleks, et muuta arusaadavaks, mis käsk oli saadetud ning mis vastus selle peale tuli, oli vajalik luua logimissüsteem logide faili salvestamiseks. Kaardi käskude saatmine kaardile toimub tarkvaras kõik läbi ühe meetodi, mis kergendab käskude saatmise voogu. Iga käsu saatmisel salvestatakse teksti faili selle sisu heksadetsimaalsüsteemi kujul nagu joonisel (Joonis 7).

```
May 10, 2019 4:20:12 PM main.java.org.javacard.JavaCardInfo sendCommandToCard
INFO: Command APDU in Hex: 00A4010C025000
```

Joonis 7 Logifailis näidatud saadetud käsu kuju.

Pärast vastuse saamist ja nende koodide kontrolli salvestatakse ka need faili, vastavalt sellele, kas vastuses olid mingid andmed või mitte. Juhul, kui andmeid pole, tagastatakse

ainult vastuse kood, kas 0x9000 või mingi muu koos kommentaariga selle koodi kohta. Vastaval korral alguses näidatakse vastuse koodi ning seejärel andmeid nagu joonisel (Joonis 8).

```
May 10, 2019 4:20:13 PM main.java.org.javacard.JavaCardInfo sendCommandToCard
INFO: Command APDU in Hex: 00B000E719
May 10, 2019 4:20:13 PM main.java.org.javacard.JavaCardInfo sendCommandToCard
INFO: [Info]: Response OK. Code: 9000

May 10, 2019 4:20:13 PM main.java.org.javacard.JavaCardInfo addErrorToLog
INFO: Certificate Response Part 2: 383110300E06035504040C074AC395454F5247311630140603
```

Joonis 8 Logifailis näidatud saadud vastuse kuju.

Iga logirea juurde salvestatakse ka kellaaeg ja kuupäev, millal see käsk saadeti ning ka klass ja funktsioon, milles see käsk välja kutsuti. Selline lähenemine kergendas tulemustest kiire ülevaate saamist ja lihtsustas andmete kopeerimist juhendisse arendajatele näitena. Lisaks iga logifaili nimi on kujul 2019-05-10_16-20-11.log, mis näitab faili loomise täpset aega. Kõik failid on vaikimisi kuvatud kronoloogilises järjekorras, mis oluliselt lihtsustab vigade leidmist või vigaste andmete ilmumispõhjuse otsimist.

3.5 Ettetulnud probleemid

Töö käigus ilmusid ka mõned probleemid. Need polnud enamasti seotud vigadega kaartidel, vaid muudatustega kaartidel ja asjaoluga, et dokumentatsioonis polnud kõik detailid hästi selgitatud.

Esimese probleemina tuli välja PIN1, PIN2 ja PUK APDU käskude saatmine. Eelmise ID-kaardi puhul parooli saatmiseks piisas sellest, et APDU andmete lahtris oli vaja kirjutada parooli numbrid kuueteistkümnendsüsteemis kujul 0x30 0x31 … 0x38 0x39, kus viimane arv on parooli number ja sellele eelnev on kolm ning andmete lahtri suurus on sama, mis on parooli sümbolite arv. Uuemal kaardil andmete lahter peab olema alati kindla suurusega, milleks on 12 baiti. Parooli kirjutatakse samal kujul nagu vanal kaardil, kuid tühjad lahtrid peavad olema täidetud väärtustega 0xFF. Juhul kui andmete pikkus on lühem või esinevad muud sümbolid, siis APDU vastuseks tuleb üks kolmest vigadest:

- Jäi kaks katset sisestada parool kood **0x63C2,**
- Jäi üks katse sisestada parool kood **0x63C1,**
- Autentimise meetod blokeeritud ehk katseid enam pole **0x6983.**

Antud lähenemine uuel kaardil on iga parooli tüübile ehk PIN1, PI2 kui ka PUK-ile.

Teiseks suuremas arusaamatuseks muutus sertifikaatide lugemise vastus. Kuna maksimaalne baitide arv, mis peaks vastusega kaasa tulema on 0xFF baiti (256 baiti), siis eeldusel, et kaardi lahter sertifikaatide baitidega koosneb mitmest täidetud osast, siis vastuse pikkus peaks olema iga osa jaoks (v.a. viimase jaoks) 256 baiti. Praktikas tuli vastuseks ainult 0xE7 baiti (231 baiti). See põhjustas sertifikaatides vigu pärast teisendamist ASCII kujule. Antud probleemi lahendamiseks töötas lahendus, et kontrollida tuleva vastuse pikkust ning juhul, kui eeldades pikkust 0xFF tuleb vastus lühem, siis proovida lugeda andmed nihkega ehk antud vea puhul alates 0xE7 kuni 0xFF ning selle puhul oodatav vastuse pikkus ehk L_e ainult 0x19 baiti (25 baiti). Kui saata APDU käsuga L_e pikkuseks suurem väärtus, siis koos tagasi tulnud andmetega tuleb mitte kood 0x9000, et kõik on korras, vaid hoiatuskood 0x6282, et faili/salvestuse lõpp tuli enne, kui jõuti lugeda L_e baiti. Vaikimisi L_e 0x00 puhul kaart tagastab kõik andmed, mis on tagastada antud failis ning ei tagasta koodi 0x6282, kui kõik andmed pole loetud. Sama probleemi ei esine Gemalto ID-kaardil.

Üheks juhuslikult esile kerkinud probleemiks oli ka CPLC andmete lugemine. Nende andmete lugemisel tuleb arvestada, et juhul, kui proovitakse neid lugeda pärast GET DATA tüüpi APDU käskude kasutamist, siis tuleb veateade 0x6E00, mis tähendab, et klass pole toetatud. Selle probleemiga toime tulekuks arendasin kaks lahendust. Esimeseks proovisin lugeda CPLC andmeid kohe pärast kaardiga ühendamist ilma, et kasutaja oleks sisestanud soovi need saada. Teiseks tuli funktsioonis, mis luges maha vajalikud andmed, panna CPLC lugemine esimeseks käsuks.

Neid juhtumeid võib pidada dokumenteerimata nõueteks või andmete lugemiseks kasutatud teegi eripäraks. Probleemide hulka ka lisandus *Singelton Class* lähenemine. Peamine probleem ilmnes selles, et teadaolevast erineva ATR-i puhul tuleb kohe alguses veateade, et tegemist on tundmatu kaardiga. See ei võimaldanud kontrollida kaarte, milles oli vigane ATR, kuid esinesid ka teised andmed kontrollimiseks. See probleem ei vaja

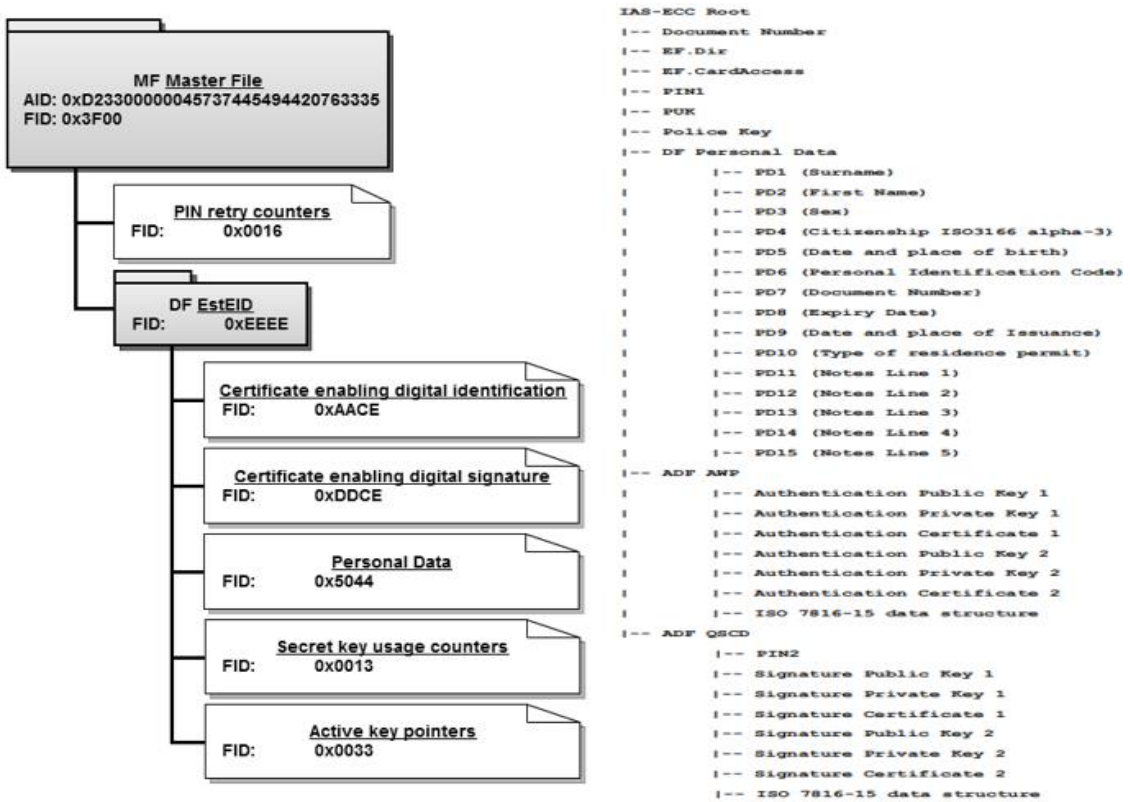
otsest lahendust, kuna vigase ATR puhul võib pidada kaarti vigaseks igal juhul. Võimaliku lahendusena sarnastelt kaartidelt info hoolimata vigasest ATRist lugemiseks võib tulevikus programmile lisada konkreetse kaardi tüübi parameetrina ette andmise.

4 Kaardi sisu testimine vastu ID kaardi profiili

4.1 Loetud andmete struktuur

Algselt tuli teha kindlaks, kuidas erineb andmete struktuur kahel kaardil. Kuna andmete paigaldus erineb ilma struktuuri ülevaateta tegutseda ei saa. Joonisel (Joonis 9) on näha, kuidas paiknesid andmed kaardil enne kaardi 2018.-nda aasta versiooni ilmumist.

Nagu eelnevalt peatükis 1.5.3 mainitud toimusid muudatused struktuuris. Nimelt on näha, kui võrrelda struktuure joonisel (Joonis 9), kuidas andmed uuel kaardil on rohkem struktureeritud. Mõnede andmete puhul on erinevus sõnastuses ja lahtrite arvus. Uuel kaardil on andmed jagatud peamisesse kolme ossa. Nendeks on personaalsed andmed, AWP ja QSCD. AWP-s asuvad andmed autentimis sertifikaatide kohta ja QSCD osas signeerimis sertifikaadid. Vanemal kaardil kõik asus Eesti poolt arendatud standardi põhjal EstEID-is ning pole siukest korrastatud struktuuri, kui uuel kaardil. Lisaks erinevad PIN-ide asukohad, kui vanal kaardil kõik asusid peafailis, siis uuel kaardil see asub QSCD osas.



Joonis 9 Vana ja uue ID-kaardi struktuur

Näitena võib tuua personaalsete andmete tabeli (Tabel 2). Muutus nii järjekord kui ka mõnede väljade sisu. Nagu näiteks sündimiskuupäev on nüüd koos sünnikohaga samas lahtris või nimede jaoks lahtrite arv on ühe võrra suurem.

Tabel 2 Personaalsete andmete struktuur

Vana Eesti ID-kaardi personaalsed andmed	Uue Eesti ID-kaardi personaalsed andmed
PD1 (Surname) PD2 (First name line 1) PD3 (First name line 2) PD4 (Sex) PD5 (Nationality) PD6 (Birth date (dd.mm.yyyy)) PD7 (Personal identification code) PD8 (Document number) PD9 (Expiry date (dd.mm.yyyy)) PD10 (Place of birth) PD11 (Date of issuance (dd.mm.yyyy)) PD12 (Type of residence permit) PD13 (Notes line 1) PD14 (Notes line 2) PD15 (Notes line 3) PD16 (Notes line 4)	PD1 (Surname) PD2 (First Name) PD3 (Sex) PD4 (Citizenship) PD5 (Date and place of birth) PD6 (Personal Identification Code) PD7 (Document Number) PD8 (Expiry Date) PD9 (Date and place of Issuance) PD10 (Type of residence permit) PD11 (Notes Line 1) PD12 (Notes Line 2) PD13 (Notes Line 3) PD14 (Notes Line 4) PD15 (Notes Line 5)

4.2 Testide loomine

Testide loomiseks kasutati JUnit 5 teeki ja testiklassi sees imporditi paketti org.junit.Assert.assertEquals. Kogu protsessi alustamiseks tehakse klassis, millel on annotatsioon BeforeClass, JSON failide import ning loetakse välja kõik võtmed. Võtmed salvestatakse HashMap-i koos JSON objekti väärtusega. Juhul, kui JSON fail puudub, siis seda proovitakse Väärtused, mis on just objektide sees ei oma tähtsust, kuna need erinevad vastavalt kaardi omanikule. Peamiseks eesmärgiks on kontrollida, kas kõik samad võtmed on ka just testitud kaardil.

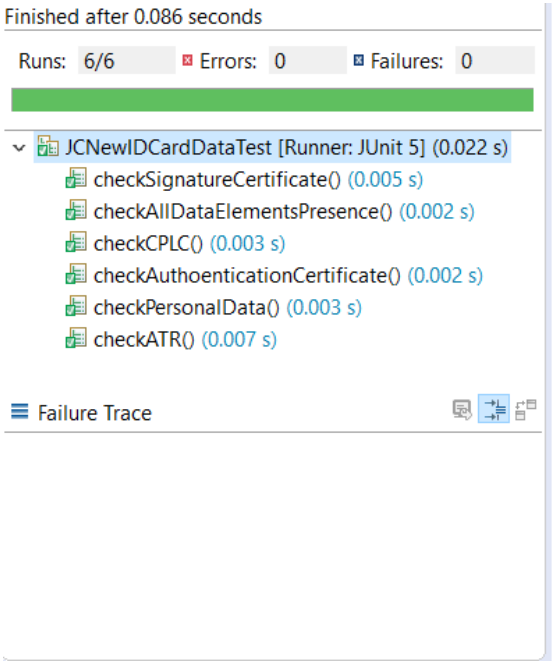
Kuna kaardi struktuur peab olema samal igal kaardil, siis juhul kui kaardi lugemisel tuleb viga nagu näiteks kood 0x6A82, mis ütleb, et fail pole leitud või mingi muu viga, mis vihjab, et fail puudub või ei saa olla loetud, siis kaart ei vasta tingimustele. See selgub

juba kaardi lugemise ajal. Vead salvestatakse JSON faili sisse vastava andme nimetuse alla. Lisaks salvestub see logifaili.

Pärast kaardi lugemist loetakse maha kaardi ATR ja tagastatakse vastav objekt, kas uue või vana Eesti ID-kaardi jaoks. Juhul kui ATR ei vasta sellele, mis on toodud spetsifikatsioonis, siis tagastatakse veateade ja andmete lugemisega edasi ei minda, kuna puudub õige käskude nimekiri ja lugemismeetod. Antud tulemus tagastab informatsiooni ainult tundmatu ATR-i kohta, aga sellest piisab, et pidada kaarti vigaseks.

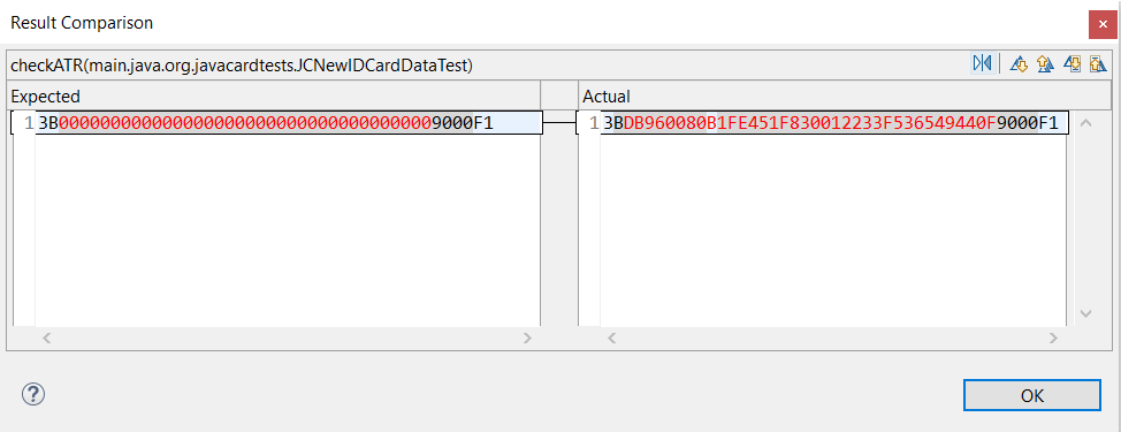
4.2.1 Ühiktestid

Ühik testid kontrollivad standartsete assert meetoditega andmete õigsust. Hetkel on testid struktureeritud nii, et iga andmetüübi jaoks on eraldi test. Tulemuse õigsuse puhul kuvatakse see nagu joonisel (Joonis 10).



Joonis 10 Edukate testide tulemus

Vastasel juhul testid näitavad viga ja kuvavad tulemuste erinevust. Erinev osa on märgitud punasega ja võimaldab tuvastada viga kiiremini. Nagu näidatu joonisel (Joonis 11), siis ATR test kuvas kaardil vea.



Joonis 11 Vale tulemuse puhul vea kuvamine

Peamiseid teste on kuus, mis jagavad enda vahel ülesannet testida õigete võtmete olemasolu kaardil. Alguses vaadatakse läbi, kas peamised andmerühmad on kaardil olemas. Andmerühmad on järgmised: Personal Data, CPLC, Authentication Certificate, Signature Certificate, ATR.

- Personaalsete andmete puhul kontrollitakse, kas võtmete hulk on õige ja seejärel kontrollitakse nende võtmete õigsust, aga ei kontrollita sisu, kuna see ei saa olla igal kardil sama.

- CPLC puhul kontrollitakse, kas võtmed vastavad dokumendis toodud võtmete nimekirjale ja kas nende sisu vastab oodatule.

- Sertifikaatide puhul kontrollitakse võtmeid ja mõnede võtmete sisu nagu näiteks Authoentication Subject, Authoentication Issuer, Signature Subject ja Autoentication Issuer.

- ATR kontrollitakse ka, kuid antud osa vajadus tuleb, kui antakse ette juba maha loetud andmeid. Selles testis võrreldakse iga sümbolit, kuna ATR on kindlalt paika pandud.

4.3 Vastavuse hindamine

Antud testide töö tulemusel saab vaadata ja hinnata, mis osa ei tagasta oodatud tulemust ning teha vastavad järeldused. Peamine järeldus ühe testi läbikukkumisel on see, et kaart ei vasta nõuetele ja seda ei saa lasta toodangusse ning tuleb teha mingid muudatused.

Kaarti saab pidada sobivaks ainult juhul, kui kõik testid on läbitud, kuid seejärel saab soovi korral kontrollida ka tekstifaili, milles asuvad kontrollitud andmed teksti kujul.

5 Plaanid kasutamisel tulevikus

Kuna kaartide tootmine jätkub, siis on vajalik ka jätkuv kontroll uutele kaardi partiidele. Antud tarkvara võimaldab kontrollida kaardi kindla osa vastavust ning tagastab testide tulemust mugavalt loetaval kujul. Mugavuse all mõeldakse siin peamiselt tulemuste kuvamist ja vigaste tulemuste esiletoomist ehk lahendab probleemi, mis võimaldab enam mitte vaadata kogu suhtlemis voogu konsoolis, vaid näha kohe tulemust failis kirjutatuna.

Juhul, kui kaardile tehakse muudatused, siis antud tarkvara suudab samal moel lugeda maha andmed kaardilt, kuna struktuur on pandud paika standardiga. Tulevikus on plaanitud lisada uued apletid, kuid need ei pea mõjutama kaartide andmeid, kuna asuvad erinevates turvadomeenides ning paigaldatavad apletid asuvad lisadomeenis.

5.1 Testitav funktsionaalsus ja võimalikud edasiarendused

Kuna kogu kaardi lugemiseks on vaja mõista lõplikult kõiki spetsifikatsiooni osi, siis antud töös on keskendatud ainult infole, mille kontroll on igal juhul vajalik. Hetkel on võimalik kaardilt lugeda informatsiooni nagu personaalsed andmed, autentimis- ja signeerimissertifikaadid, sertifikaadi ISO andmed AWP osast ja QSCD osast, ATR ning CPLC osa.

Personaalsete andmete lugemine võimaldab lugeda maha kõik salvestatud andmed ja kontrollida, kas kõik elemendid eksisteerivad vastavalt struktuurile, mis on joonisel Joonis 9. Andmete sisu poolest lisandub ka asjaolu, kas andmed, mis on trükitud kaardile ja mis on kiibi peal, on samad. Nende erinevuse puhul võib pidada kaarti vigaseks. Testkaartide puhul ei ole võimalik kontrollida nende välimuse õigsust.

Tähtis osa kiipkaartidel on sertifikaadid ja nende olemasolu ning õigsus kaardil. Hetkeseisuga saab kindlaks teha nii sertifikaatide olemasolu kui ka lugeda maha, mis neis on kirjas. Sertifikaatidelt on võimalik välja lugeda mõned samad andmed, mis on olemas ka personaalsetes andmetes nagu nimi või riik, mis on dokumendi väljastanud.

Kaardid testimiseks tulevad suurtes partiides, mis peaks kõik olema loodud sama malli järgi ning nende testimisel piisab mõnede kaartide kontrollimisest, kuna tulemus peab

olema kõigil sama. Kuigi loodud tarkvara võimaldab kiiremini näha tulemust, jääb kaartide sisestamine terminali ikka käsitsi tehtavaks protsessiks.

5.2 Lisanduv funktsionaalsus

Hetkel puudub võimalus kuvada automaatselt kontrollitud informatsiooni, mis on peidetud kaardil PIN-ide taha, nende hulgas ka sertifikaatide privaatseid võtmeid. Uue kaardi lisadomeeni (Supplementary Domain) ja kontaktvaba liidese lisandumisega, kaardile ilmusid uued võimalused ja osad, mida saab tulevikus loodavates rakendustes kasutada ja testida.

Tarkvara võimaldab sisestada PIN-e ja kontrollida nende õigsust, kuid ei tagasta informatsiooni kujul, mida oleks võimalik kontrollida. Lisaks puudub võimalus korralikult sisestada parooli selle kontrollimiseks. Antud funktsionaalsus on olemas, kuid antud töös sellest ei räägita põhjalikult, kuna puudub automatiseeritud tulemuste kuvamine antud osast.

Lisadomeeni peamiseks eesmärgiks on võimaldada tulevikus paigaldada kaardile kolmandate osapoolte toodetud applete. Võrreldes kaardi vanema versiooniga, millele sellist võimalust polnud, saab uut kaarti põhimõtteliselt tulevikus kasutada ka näiteks ühistranspordis pileti valideerimiseks. Kuna rakenduste paigaldamisprotsess võib põhjustada kaardil teadmata vigu, siis vajab see osa ka testidega katmist. Lisaks ka fakt, et kaardi mälu pole piiramatu ning selle kontrolli oleks tarvis teostada ka arusaadaval ja mugaval kujul kaardi kontrollijatele.

Kontaktvaba liides võimaldab teoorias lugeda informatsiooni kaardilt maha kasutades standardset NFC kaardi lugejaid. Hetkel oleks võimalik seda teostada ka mõnede Androidi telefonidega, mis toetavad NFC funktsionaalsust. Kuigi uute kaartidega on võimalik suhelda NFC abil, siis hetkel on selle liidese kaudu kaardilt informatsiooni saada võimatu, kuna APDU vastus katsele lugeda on 0x6982, mis vastab veale [Error]: *Security Conditions. Not Satisfied* (Turva tingimus pole täidetud). Näide on toodud joonisel (Joonis 12).

```
Command APDU in Hex: 00A4010C025000
[Error]: Security Conditions. Not Satisfied Code: 6982
```

Joonis 12 APDU vastus katsel lugeda PersonalData NFC abil

6 Kokkuvõte

Java-kaartide tehnoloogia leiab laia kasutust igal pool maailmas tänu tehnoloogia mugavusele ning funktsionaalsusele. Tänu nende laiadele kasutamisvõimalustele nagu näiteks panga- ja transpordikaardid, kasutatakse kõigis arenenud riikides. Nende hulka kuuluvad ka tänapäevased elektroonilised ID-kaardid, mis on kasutusel ka Eestis. Eesti ID-kaart on vahend, mis kannab elektroonilist identiteeti. Seda kasutatakse Eestis juba aastast 2004, millal ilmus esimene Eesti ID-kaart, mis oli toodetud Gemalto poolt. Alates 2018. aasta detsembrist väljastatakse uusi variante kaardist, mille tootja on IDEMIA ja millega suhtlemiseks oli vaja teha muudatusi ka rakendustarkvaras, et suuta leida andmeid uuelt kaardi struktuurilt.

Käesolevas töös kirjeldasime testimislahendust ID-kaartidelt andmete lugemiseks ja nende nõuetele vastavuse kontrollimiseks. Testimisprotsessi läbimiseks peavad kaardid funktsioneerima korralikult ja tagastama informatsiooni konkreetsel kujul. Selleks on varasemalt loodud erinevaid tarkvaralahendusi, nagu näiteks OpenSC ja GlobalPlatformPro, mis aitavad saata vastavaid APDU käske kaartidele. Kuigi nende hulgas on väga funktsionaalseid vahendeid, need keskenduvad kaartidega suhtlemiseks üldiselt. Kiipkaartide kontrollimise protsess võib olla väga ajakulukas ja vahel ebamugav. Seda probleemi aitab leevendada käesolevas töös kirjeldatud projekti raames välja töötatud tarkvara. See keskendub Eesti ID-kaartidele ning võimaldab lugeda maha peamist informatsiooni kaardilt, mis võib olla rikutud ja mis võib tekitada tehnilisi tõrkeid isikutuvastamisel.

Kaardi informatsioon paigaldus uuel kaardil muutus. See on rohkem struktureeritud nimelt AWP-s asuvad andmed autentimis sertifikaatide kohta ja QSCD osas signeerimis sertifikaadid. Eraldi failis asuvad kõik personaalsed andmed. Nende andmete lugemiseks saab kasutada APDU käsud. Käskude loogika mõlemal kaardil on sama, kuid erineb failide viitamiseks mõeldud andmete sisu *Data* lahtris APDU käsus.

Selle projekti raames tehtud töö tulemusel loodud tarkvaraga saab suhelda Eesti ID-kaartidega läbi APDU käskude ja luua maha loetud andmetest andmefailid ja võrrelda

neid olemas oleva malliga. Järeldust saab näha, kas JUnit testide tulemusel või pärast nende testide käivitamist eraldi faili analüüsides. Töö raames avastasin, et mõnede andmete lugemisel võivad tekkida probleemid, mida vanade ID-kaartidega ei esinenud. Näiteks sertifikaatide lugemisel ei tagastata uue kaardi puhul oodatud baitide kogust, vaid eeldatakse lugemiskäsu korduvat kasutamist. Lisaks muutus paroolide saatmisel andmete osa pikkus fikseeritud suuruseks, mis põhjustab ebamugavust, kui proovitakse aru saada, milles on viga. Lisaks tulid töö käigus välja mõned uue kaardi täiendavad eripärad nagu CPLC andmete lugemisel, et neid ei saa lugeda, kui enne seda prooviti saada informatsiooni GET DATA abil.

Avastatud eripärasid pole spetsifikatsioonis mainitud, mis tegi nendega toimetulemise pärast avastamist raskemaks. Selliseid eripärasid võib pidada dokumenteerimata nõuetena või lugemiseks kasutatud teegi eripärana. Kirjeldatud probleemide dokumenteerimine käesoleva bakalaureusetöö raames võiks aidata tarkvaraarendajatel, kes arendavad uue ID kaardiga suhtlemise tarkvara, nende eripäradega seotud probleeme kiiremini lahendada. Testimisosa tarkvarast kontrollib andmeid, mis asuvad personaalsete andmete failis, autentimis- ja signeerimissertifikaatide failis ning koos nendega ka ATR ja CPLC. Viimase kahe puhul peab kontrollima täpset informatsiooni kaardil, kuna spetsifikatsiooni järgi mõlema väärtused on fikseeritud. Ülejäänute puhul peab arvestama, et neil oleks olemas kõik võtme elemendid. Antud projekti raames ehitatud Java testid kontrollivad selle informatsiooni õigsust.

Kogu protsess on läbipaistev, mis aitab aru saada vigade ilmumisel, millega on tegu, ja võimaliku vea põhjust. Logi failid salvestatakse kõik eraldi kausta, kus neid on võimalik hiljem vaadata. Kuna tehtud tarkvara täidab soovitud tulemust, siis seda on võimalik kasutada uute kaardi partiide testimiseks. Kaardi peale plaanitakse tuua muudatused ning selle tarkvaraga on võimalik testide uute partiide eksemplare, et kontrollida nende andmete õigust ja vastavust spetsifikatsioonile. Hetke seisusuga kaarti peetakse õigeks, kui peamised andmed nagu sertifikaadid ja personaalsed andmed on kõik olemas kaardil ning nende võtmelahtrid on õiged ehk vastavad spetsifikatsioonile.

Kasutatud kirjandus

[1] “ Java Card Technology,” [Online]. Available:
https://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html.
[Accessed 28 April 2019].

[2] “ Protecting digital services through standardization,” [Online]. Available:
https://globalplatform.org/specifications/technical-overview/.
[Accessed 28 April 2019].

[3]“ eID lühitutvustus,” [Online]. Available:
https://eid.eesti.ee/index.php/EID_l%C3%BChitutvustus.
[Accessed 28 April 2019].

[4]“ Elektrooniline identiteet eID,” [Online]. Available: https://www.ria.ee/et/riigi-infosusteem/elektrooniline-identiteet-eid.html.
[Accessed 28 April 2019].

[5]“ EstEID,” [Online]. Available:
https://esteid.org/avatud-id-kaart/ .
[Accessed 10 April 2019].

[6]“ PPA võitis ID-kaardi tootmise riigihankega seotud kohtuvaidluses Gemaltot,” [Online]. Available: https://www.delfi.ee/news/paevauudised/eesti/ppa-voitis-id-kaardi-tootmise-riigihankega-seotud-kohtuvaidluses-gemaltot?id=85981123.
[Accessed 12 April 2019].

[7]“ ID-kaartide tootja Gemalto: PPA 152 miljoni nõue on ebaproportsionaalselt suur,” [Online]. Available: https://www.delfi.ee/news/paevauudised/eesti/id-kaartide-tootja-gemalto-ppa-152-miljoni-noue-on-ebaproportsionaalselt-suur?id=83833885.
[Accessed 12 April 2019].

[8]“ GlobalPlatformPro,” [Online]. Available:
https://github.com/martinpaljak/GlobalPlatformPro.
[Accessed 20 April 2019].

[9]“ OpenSC documentation,” [Online]. Available:
https://github.com/OpenSC/OpenSC.
[Accessed 10 May 2019].

[10]“ Implementation of GlobalPlatform smart card specification,” [Online]. Available:
https://sourceforge.net/p/globalplatform/wiki/GlobalPlatform%20Card%20Specificatio/.
[Accessed 10 May 2019].

[11]“ Jan Philipps, Alexander Pretschner, Oscar Slotosch, Ernst Aiglstorfer, Stefan Kriebel, Kai Scholl: Model-Based Test Case Generation for Smart Cards. Electr. Notes Theor. Comput. Sci. 80: 170-184 (2003)

[12]“ Estonian Electronic ID card application specification,“, [Online]. Available: https://www.id.ee/public/RIA-EstEID-Chip-App-v3.5.8_fix_form.pdf.
[Accessed 12 May 2019].

[13]“ Estonia ID1 Chip/App 2018,“, [Online]. Available: https://installer.id.ee/media/id2019/TD-ID1-Chip-App.pdf.
[Accessed 12 May 2019].

[14]“ Java Card 3 Platform Development Kit User Guide, Classic Edition,“, [Online]. Available: https://docs.oracle.com/javacard/3.0.5/guide/index.html.
[Accessed 12 May 2019].

[15]“ ISO 7816 Smart Card Standard,“[Online]. Available: https://cardwerk.com/iso-7816-smart-card-standard/.
[Accessed 12 May 2019].

[16] Magnus Nyström: PKCS#15 - A Cryptographic Token Information Format Standard. Smartcard 1999

[17]“ Complete list of APDU responses,“[Online]. Available: https://www.eftlab.com/index.php/site-map/knowledge-base/118-apdu-response-list.
[Accessed 12 May 2019].

[18]“ Package javax.smartcardio,“[Online]. Available: https://docs.oracle.com/javase/7/docs/jre/api/security/smartcardio/spec/javax/smartcardio/package-summary.html.
[Accessed 12 May 2019].

[19]“ Smart Card Reader T0 T1 communication on APDU level,“[Online]. Available: https://stackoverflow.com/questions/29993498/smart-card-reader-t0-t1-communication-on-apdu-level.
[Accessed 12 May 2019].

[20]“ Smart card ATR parsing,“[Online]. Available: https://smartcard-atr.appspot.com/.
[Accessed 13 May 2019].

[21]“ Understanding Java Card 2.0,“[Online]. Available: https://www.javaworld.com/article/2076617/understanding-java-card-2-0.html.
[Accessed 13 May 2019].

[22]“ Developing International Standards,“[Online]. Available: https://www.iec.ch/about/activities/standards.htm.

[Accessed 13 May 2019].

[23]“ EVS 827:2004,“[Online]. Available: https://www.evs.ee/tooted/evs-827-2004.

[Accessed 13 May 2019].

Lisa

Tabel 3 Näite andmed ja testkaardil testitavate andmete sisu

ATR	3BDB960080B1FE451F830012233F536549440F9000F1
CPLC	IC Fabricator: 9F7F IC Type: 2A47 Operating System: 906B Operating System Release Date: 4882 Operating System Release Level: 3171 IC Fabrication Date: 8100 IC Serial Number: 9A805701 IC Batch Identifier: 0004 IC ModuleFabricator: C125 IC ModulePackaging Date: 0011 ICC Manufacturer: 4282 IC Embedding Date: 6311 Pre-personalizer Identifier: 4382 Pre-personalization Date: 6311 Pre-personalization Equipment: 44826301 Personalizer Identifier: 1B00 Personalization Date: 0011 Personalization Equipment: 45829102
Personal Data	Surname : JÕEORG First name : JAAK-KRISTJAN Sex : M Citizenship : EST Date and place of birth : 08 01 1980 EST Personal identification code : 38001085718 Document number : AS9991019 Expiry date : 18 10 2023 Date and place of issuance : 18 10 2018 Type of residence permit : Empty

	Notes line 1 : Empty Notes line 2 : Empty Notes line 3 : Empty Notes line 4 : Empty Notes line 5 : Empty
Authoentication subject	SERIALNUMBER=PNOEE-38001085718 GIVENNAME=JAAK-KRISTJAN SURNAME=JÕEORG CN="JÕEORG,JAAK-KRISTJAN,38001085718" C=EE
Authoentication Issuer	CN=TEST of ESTEID2018 OID.2.5.4.97=NTREE-10747013 O=SK ID Solutions AS C=EE
Authoentication Serial Number	120276599402251628960986069547601100912
Signature Subject	SERIALNUMBER=PNOEE-38001085718 GIVENNAME=JAAK-KRISTJAN SURNAME=JÕEORG CN="JÕEORG,JAAK-KRISTJAN,38001085718" C=EE
Signature Issuer	CN=TEST of ESTEID2018 OID.2.5.4.97=NTREE-10747013 O=SK ID Solutions AS C=EE
Signature Serial Number	134568582442789304330132665910710207034