TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatika instituut
Infosüsteemide õppetool

Rakendus asutuse digitaalse templi operatsioonideks

bakalaureusetöö

Üliõpilane: Fred Martmaa
Üliõpilaskood: 123583 IAPB
Juhendaja: Dr Gunnar Piho

Tallinn
2015

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.

(*kuupäev*) (*allkiri*)

Annotatsioon

Antud bakalaureusetöö eesmärgiks on tutvustada digitaalset templit andva rakenduse võimalusi ning luua digitaalset templit võimaldava rakenduse prototüüp.

Antud bakalaureusetöö keskendub asutuse digitaalse templi andmise võimalustele ning olemasolevale asutuse digitaalset templit võimaldava käsureautiliidile kasutajaliidese ning lisafunktsionaalsuse loomisele.

Konkreetse bakalaureusetöö tulemusena valmis esialgne rakenduse prototüüp, mis võimaldab teostada digitaalse templiga seotud operatsioone.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 38 leheküljel, 5 peatükki, 16 joonist, 2 tabelit.

Abstract

The aim of this thesis is an attempt to introduce the possibilities with digital stamp of the institution and to create a new application using possibilities with the digital stamp of the institution.

Main problem of this thesis is to introduce existing possibilities with digital stamp of the institution and to create new user interface with advanced functionality using existing command line program.

Main result of this specific bachelor thesis is a prototype, which allows to use operations with digital stamp of the institution.

The thesis is in Estonian and contains 38 pages of text, 5 chapters, 16 figures, 2 tables.

Lühendite ja mõistete sõnastik

PKI	***Public Key Infrastructure*** Avaliku võtme infrastruktuur, mis koondab endas digiallkirja andmiseks ja kontrollimiseks vajaminevaid teenuseid ning võimaldab ebaturvalistes keskkondades riskivabalt suhelda.
USB	***Universal Serial Bus*** Universaalne järjestikusiin on arvuti välissiini standard, mis lubab välisseadmeid külge ja lahti ühendada ilma, et arvuti oleks vaja välja lülitada.
XML	***Extensible Markup Language*** Laiendatav märgistuskeel, mille eesmärgiks on struktureeritud info jagamine.
PKCS	***Public-Key Cryptography Standard*** Avaliku võtme krüptograafia standard defineerib RSA formaadi avaliku ja privaatvõtme jaoks.
OCSP	***Online Certificate Status Protocol*** Kehtivuskinnitus, mis kujutab endast klient-server süsteemi, millega saab infot sertifikaadi kehtivuse/mittekehtivuse kohta.
Library	***Teek*** Kollektsioon funktsioone, makrosid või klasse, mis on mõeldud korduvkasutuseks programmides.
RFC	***Request for Comments*** Interneti standard.
AES	***Advanced Encryption Standard*** Andmete krüpteerimise spetsifikatsioon.

CA	***Certificate Authorities*** Asutus, mis annab välja digitaalseid sertifikaate
URL	***Uniform Resource Locator*** Internetiaadress ehk universaalne ressursilokaator, mida kasutatakse internetis.

Jooniste nimekiri

Tabelite nimekiri

Sisukord

1. Sissejuhatus

AS Citadele banka Eesti filiaali andmetel digiallkirjastatakse iga kuu ligi 6000 dokumenti ning sellest mõnikümmend on digiallkirajstatud asutuse digitempliga. Iga kuu kasvab vajadus anda dokumentidele ettevõtte digitempel. Sertifitseerimiskeskus tagab asutuse digitempli andmiseks Tempelplus tarkvara, mida Sertifitseerimiskeskus väljastab vaid krüptopulgal.

Bakalaureusetöö eesmärgiks on projekteerida ja välja töötada digitembeldamise keskkond. Tulenevalt ettevõtte AS Citadele banka Eesti filiaali soovist krüpteerida, dekrüpteerida ning anda dokumentidele ettevõtte digitempel, arhiveerida dokumente ning pidada logifaile tehtud operatsioonide kohta, projekteerisin prototüübi keskkonnast, mis võimaldab anda ligipääasu ettevõtte digitempli sertifikaatidele kõikidele selleks operatsiooniks volitatud töötajatele. Töö käigus selgitati digitempli põhimõtteid ning tehnoloogiat, millega on andmete digitembeldamine, krüpteerimine ja dekrüpteerimine võimalik, koostati keskkonna tarkvara dokumentatsioon ning loodi ka süsteemi peegeldav prototüüp.

Seoses projekti suurusega, selgitatakse rakenduse tööd ühe allsüsteemi (dekrüpteerimise) näitel.

1.1 Probleem

Asutusel AS Citadele Banka Eesti filiaal on üks digitempli krüptopulk, millel on asutuse sertifikaadid. Asutuse digitempli andmiseks ning dokumentide krüpteerimise/dekrüpteerimise eelduseks on digitempli tarkvara ligipääs krüptopulgal olevatele sertifikaatidele, mida on võimalik saavutada ainult krüptopulga ühendamisel arvutiga. Hetkel on krüptopulk ainult ühel isikul ning seega on ka ainult ühel inimesel filiaalis võimalik krüpteerida, dekrüpteerida ning digiallkirjastada dokumente. Seoses ettevõtte kiire arenguga ja töötajate arvu kasvuga ei ole enam ühel inimesel võimalik teostada kõiki digitempliga seotud operatsioone. Jagades ühte krüptopulka mitme inimese vahel ei ole enam hiljem võimalik kontrollida, milline töötaja millise dokumendi digiallkirjastas, mis on pangale liiga suur turvarisk.

1.2 Ülesande püstitus

Antud keskkonna eesmärk on igale keskkonnale ligipääsu omavale töötajale anda võimalus teostada asutuse digitempliga vastavalt töötajale määratud õigustega operatsioone, milleks on andmete krüpteerimine/dekrüpteerimine ning allkirjastatud andmete töötlemine. Teine eesmärk on piisavalt optimaalne digiallkirjastatud dokumentide arhiveerimine ettevõtte serveris asuvas andmebaasis, mis annaks võimaluse kontrollida dokumendi sisu, mida allkirjastati. Kolmas eesmärk on piisavalt informatiivse logifaili pidamine, kust on võimalik selgelt välja lugeda, milline töötaja millise dokumendi allkirjastas ning millal antud tegevus toimus. Neljas eesmärk on muuta keskkond lihtsasti integreeritavaks, mis jätaks piisava isoleerituse programmi funktsionaalsust teostava osa ning kasutajaliidese vahel.

2. Metoodika

2.1 Digitaalne tempel

Digitaalne tempel on tõend, et elektroonilisel viisil saadetud dokumendid pärinev info pärineb just sellest asutusest. See on digitaalsel viisil antud kinnitus, et vastav asutus on seotud konkreetse digitaalse dokumendiga ning dokumenti ei ole vahepeal muudetud. Digitaalset templit saavad kasutada nii ettevõtted, riigi –ja kohaliku omavalitsuse asutused, füüsilisest isikust ettevõtjad kui ka avalik-õigusliku ameti kandjad. Digitaalset templit saab kasutada nii koos digiallkirjaga kui ka ilma. Kasutades digitaalse templi sertifikaati koos isiku digitaalallkirjaga, võib olla kindel, et dokumendi allkirjastanud isik on volitatud vastava ettevõtte nimel dokumente allkirjastama. Digitaalse templi kasutamine on sarnane ID-kaardi kasutamisega allkirjastamisel, st dokumendile lisatakse templi andmise aeg ja kehtivuse info. Digitemplit on võimalik kasutada DigiDoc tarkvaraga ja Sertifitseerimiskeskuse poolt pakutava Tempelplus tarkvaraga.Digitemplit saab kasutada ID-kaardi baastarkvaraga paigaldatavat DigiDoc3 klient programmi, massiallkirjastamiseks mõeldud TempelPlus tarkvara või kirjutada teistsugune digitembeldamise rakendus kasutades DigiDoc teeke. [11]

2.1.1 TempelPlus

TempelPlus on digitembeldamise tarkvara, mis on arendatud kasutades JDigiDoc teeki. TempelPlus tarkvara on mõeldud suurema koguse failide allkirjastamiseks asutuse digitempliga. TempelPlus tarkvara on abiks juhtudel, kui allkirjastatavaid faile on palju ja iga allkirjastatava faili kohta peab tekkima eraldi allkirjastatud konteiner.Lisaks massallkirjastamisele on TempelPlus tarkvaral ka hulk lisavõimalusi allkirjastatud failide töötlemiseks – allkirjade kontroll, allkirjastatud konteinerist andmefailide väljavõtmine jne. Lisaks saab andmefaile hulgi krüpteerida/dekrüpteerida. TempelPlus tarkvara on võimalik kasutada vaid asutuse digitempliga. [2]

2.2 eToken PRO

eToken PRO on kaasaskantav USB autentimise krüptopulk, mis sisaldab kiipkaardi tehnoloogiat. eToken PRO kasutab sertifikaadi põhist tehnoloogiat(PKI), et genereerida ja hoida privaatvõtmeid, paroole ja digitaalseid sertifikaate. eTokenil olevaid andmeid hoitakse kiipkaardi kiibi turvatud keskkonnas. Autentimiseks on oluline omada nii krüptopulka kui ka krüptopulga kasutamiseks vajalikku parooli. Antud meetod tagab kaheastmelise kinnitamine tagabturvalisema keskkonna kui tavalise parooli sisestus. eToken PRO tagab vahendid, et seda lihtsalt integreerida kolmanda osapoolte süsteemidesse. Asutuse kaubamärgi toetamiseks tagab eToken PRO võimalust konfigureerida asutuse logo trükkimist. [7]

Joonis 1 - eToken Pro krüptopulk

2.3 JDigiDoc

JDigiDoc on programmeerimiskeeles Java loodud teek. Antud teek pakub funktsionaalsust DIGIDOC-XML 1.3 ja BDOC 2.1 formaadis digitaalselt allkirjastatud failide loomiseks, lugemiseks, allkirjastamiseks, kehtivuskinnituse hankimiseks ja allkirjade ning kehtivuskinnituste kontrolliks. Lisaks digiallkirjastamisele pakub JDigiDoc ka krüpteerimist ja dekrüpteerimist vastavalt XML-ENC(XML Encryption Syntax And Processing) standardile. Antud standard käsitleb XML dokumentide või nende osade krüpteerimist aga lubab ka krüpteerida ka suvalisi binaarifaile kodeerides need eelnevalt Base64 kodeeringus. JDigiDoc suhtleb PKCS#11 mooduliga, mis töötleb krüpteeritud kiipe kiipkaartidel. [5]

2.4 DigiDoc Failivormingud

2.4.1 DDOC

Digitaalallkirjastatud failide formaat baseerub ETSI TS 101 903 standardil, mida kutsutakse *XML Advanced Electronic Signatures*(XAdES). Antud standard kirjeldab digitaalallkirjastatud dokumentide struktuuri erinevatel täiendava kehtivuskinnituse info sisaldavuse tasemetel.

DigiDoc vastav XAdES profiilile „XAdES-X-L". Antud profiil võimaldab allkirjaga siduda järgnevad allkirjastatavad atribuudid:

- Allkirjastamiseks kasutatav sertifikaat
- Allkirjastamise aeg
- Allkirjastamise asukoht
- Allkirjastaja roll või resolutsioon
- Allkirjas sisaldub allkirjastaja sertifikaadi kehtivuse info
- OCSP vastus
- OCSP serveri sertifikaat

Antud mudeli tulemusena on võimalik XAdES-C-L profiilile vastavat allkirja kontrollida ilma täiendava infota-allkirja kontrollija peab usaldama allkirjastaja sertifikaadi väljaandjat ja OCSP kehtivuskinnituse serveri sertifikaati. Näiteks DigiDoc kliendi puhul tähendab see, et antud sertifikaadid peavad olema Windowsi sertifikaadihoidlas. DigiDoc süsteem kasutab ülaltoodud mudelile vastavate failide puhul **.ddoc** laiendit. [3]

2.4.2 BDOC

BDOC on uus allkirja vorming, mis asendas 2015. Aastal Eesti-spetsiifilise DDOC digitaalallkirja vormingu. Oluline muudatus võrreldes DDOC-iga on see, et .bdoc faili näol on tegu ZIP-konteineriga, mis sisaldab allkirjastatud faile, allkirju ja juhtinfot ning mida saab põhimõtteliselt avada suvalise ZIP-formaati tundva programmiga. Tänu sellele on seal sees olevad failid lihtsalt loetavad.

Kuna ZIP on pakitud vorming, võivad BDOC-vormingus failid olla DDOC-ist oluliselt väiksemad. Lisaks on BDOC failide edastamine meili teel paremini toetatud, kuna DDOC-

vormingus dokumentide edastamisel on tekkinud probleeme (osad meiliserverid filtreerisid DDOC vormingus olevad failid manusest välja).

BDOC digiallkirjaformaadist on olemas kaks alamformaati, mis on tehniliselt erinevad.

- .bdoc ehk BDOC-TM ehk ASiC-E LT-TM allkiri on BDOC allkiri ajamärgendiga – allkirja pikaajaline tõestusväärtus on tagatud kasutades RFC2560 standardil põhinevat ajamärgendit. See on Eestis vaikimisi kasutusel olev allkirjaformaat alates 2015. aastast.

- .asice ehk BDOC-TS ehk ASiC-E LT allkiri on BDOC allkiri ajatempliga, mille allkirja pikaajaline tõestusväärtus on tagatud RFC 3161 standardil põhineva ajatempliga. [3]

2.4.3 CDOC

CDOC-vormingu puhul järgitakse *XML Encryption Syntax and Processing* (XML-ENC) standardis väljatoodud struktuuri.[3] Konteineriks on XML-dokument juurelemendiga *EncryptedData* – selle sees on:

- Krüpteerimisalgoritmi AES-128-CBS identifikaator
- Sümmeetriline võti krüpteeritud kujul (krüpteerimisalgoritmi RSA 1.5 identifikaator, vastuvõtja nimi, vastuvõtja sertifikaat, võti krüpteeritud kujul)
- Sisalduvate andmete metainfo (algfaili nimi, faili suurus, faili MIME tüüp)
- Algandmed krüpteeritud kujul

2.5 Avaliku võtme infrastruktuur (PKI)

2.5.1 Ülevaade avaliku võtme krüptograafiast

Avaliku võtme krüptograafias kasutatakse kahte võtit – avalikku ja salajast võtit. Avalikku võtit kasutatakse vastavalt krüpteerimiseks ja vastava salajase võtmega antud signatuuride verifitseerimiseks. Avalik võti ja salajane võti genereeritakse korraga ja nad on omavahel seotud eelnevalt kirjeldatud omaduste abil, kuid avalik võti ei paljasta informatsiooni salajase võtme kohta ning samuti pole võimalik avalikku võtit kurjasti ära kasutada [9]. Avaliku võtme krüptograafia jaguneb põhiliselt kaheks – sümmeetriline ja asümeetriline krüptograafia.

2.5.2 Asümmeetriline krüptograafia

1976. aastal avalikustasid Whitfield Diffie ja Martin Hellman asümmeetrilise krüptograafia meetodi ning taolist võtmete vahetuse meetodit tuntakse kui Diffie-Hellmani algoritmina.

Asümmeetrilise võtmega algoritmid kasutavad kahte võtit andmete krüpteerimiseks ja dekrüpteerimiseks. Kuna asümmeetrilise krüpteerimise algoritm on ressursi ja ajamahukas, siis reaalses rakenduses kasutatakse andmete krüpteerimisel sümmeetrilist algoritmi.

Algoritmi esmärk on võimaldada kahel isikul vahetada oma võtmeid turvaliselt, nii et neid võtmeid oleks võimalik kasutada sümmeetrilistes algoritmides. Algoritmide aluseks on raskused, mis tekivad diskreetsete logaritmide alusel.

Primitiivse juure mõiste – A on algarvu Q primitiivne juur kui on võimalik moodustada sellest arvust lähtuvad jadad. A $mod\ Q$,$A^2 mod\ Q$,…,$A^{q-1} mod\ Q$, kus Q väärtuseks on kõik võimalikud täisarvud vahemikus 1…Q-1. Sellisel juhul leidub iga Y < Q ning primitiivse juure A ning algarvu Q jaoks ainult üks selline x, et

$$Y = A^x mod\ Q, kus\ 0 \leq x \leq (Q-1)$$

Arvu x nimetatakse diskreetseks logaritmiks.

Näide: Olgu mõlemale isikule teada g ja p. Need arvud ei ole salastatud ja võivad olla ka teistele teada. Selleks, et saada salastatud võtmed, tuleb mõlemal isikul (nt Alice ja Bob) genereerida suured arvud: näiteks A ja B. Alice arvutab $A = g^a mod\ p$ ning saadab Bobile. Bob arvutab $B = g^b mod\ p$ ning saadab Alicele. Alice teab arvu B ja saab arvutada $K = B^a mod\ p = g^{ab} mod\ p$. Samuti teab Bob arvu A ning saab arvutada $K = A^b mod\ p = g^{ab} mod\ p$. [12]

2.5.3 RSA

RSA on asümmeetriline krüpteerimise algoritm, mida kasutatakse sõnumite krüpteerimiseks ja dekrüpteerimiseks. Esimest korda tutvustasid RSA algoritmi Ron Rivest, Adi Shamir ja Leonard Adleman 1978. aastal.

Võtmete loomine:

- Alustuseks valitakse kaks algarvu p ja q
- Algarvud korrutatakse omavahel n=pq
- Arvutatakse $\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$
- Valitakse eksponent e nii, et suurim ühistegur $\big(e, \varphi(n)\big) = 1$. Teisisõnu peavad e ja $\varphi(n)$ olema kaasalgarvud.
- Arvutatakse e pöördarv moodulis $\varphi(n)$; $d \equiv e^{-1}(mod\varphi(n))$. Selleks saab kasutada laiendatud Eukleidese algoritmi.

Täisarvu paar (e,n) on avalik võti ja (d,n) on privaatne võti. [10]

2.6 Digitaalne Sertifikaat

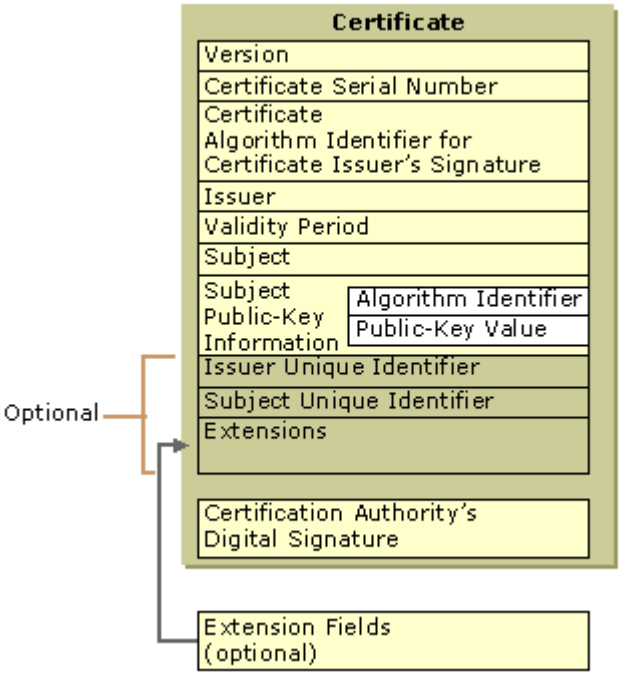
Krüptograafias tähendab avaliku võtme sertifikaat elektroonilist dokumenti, millega tõestatakse, et avaliku võtme omanik on just see isik, kes väidab ennast seda olevat. Sertifikaat sisaldab informatsiooni võtme kohta, omaniku identiteedi kohta ning digitaalallkirja kolmandalt osapoolelt, mis tõestab, et antud sertifikaat kehtib. Usaldust avaliku võtme omaniku kohta kinnitab usaldus kolmanda osapoole vastu.

Tavalises PKI struktuuris on allkirjastajaks CA(*Certificate Authority*), kelleks on tavaliselt omakorda sertifikaate väljastav asutus. *Web of Trust* skeemis on allkirjastajaks kas avaliku võtme omanik või usaldusväärne sertifikaatide väljastaja.

Usaldatud sertifikaate kasutatakse selleks, et luua turvaline ühendus serveriga kasutades internetiühendust. Sertifikaat on oluline selleks, et vältida pahatathlikku kolmandat osapoolt, kes ühenduse ajal väidab ennast olevat sihtpunkt, mille pihta ühendus loodi. Antud stsenaariumit nimetatakse *„man in the middle attack“* rünnakuks. [6]

2.6.1 X.509 Sertifikaat

X509 sertifikaat on ITU-T avaliku võtme infrastruktuuri(PKI) ja priviligeeritud haldamise infrastruktuuri(PMI) standard. X509 kirjeldab ära standardse formaadi avalikule võtmele, tühistatud sertifikaatidele, sertifikaadi atribuutidele ja sertifikaadi valideerinud algoritmi. [4]



Joonis 2 - x509 sertifikaat

3. Rakendus

3.1 Arhitektuur

Rakendus on loodud programmeerimiskeeles Java. Rakenduses järgib Modal-View Controller (MVC) ning objekt-relatsioonilise kaardistamise(Object Relational Mapping - ORM) mustrit. Rakendus kasutab Spring raamistikku ning andmebaasiga suhtlemiseks Hibernate teeki. Rakenduse andmebaasisüsteemi haldamiseks kasutatakse MySQL-i. Rakendus hakkab töötama Apache Tomcat Servletil.

Rakendus on loodud keskkonnas Eclipse ning testitud kasutades Wamp serverit, mis sisaldab endas Apache veebiserverit ning MySQL andmebaasi.

Rakenduse kasutajaliides on loodud kasutades JavaServer Pages tehnoloogiat. Kasutajaliidese kujundus on loodud kasutades Twitter Bootstrap teeki.

Rakenduse turvalisuse tagab Spring Security Cross Site Request Forgery (CSRF) protection ja Spring Security BCryptPasswordEncoder, mis genereerib oluliste väljade räsi.

Operatsioonide logimiseks kasutatakse Apache log4j utiliiti.

Rakenduse digitaalse templi operatsioonid teostab TempelPlus käsurea utiliit, mis on antud rakenduse jaoks sobivaks seadistatud

Keskkonna testimiseks kasutatakse Aladdin eToken Pro krüptopulka, kuhu peale on pandud asutuse Citadele Banka Eesti filiaali testsertifikaadid ning mis tuleb ühendada siis kas serveri või mõne muu seadme USB porti.

3.2 TempelPlus seadistamine rakenduse jaoks

Selleks, et kasutada TempelPlus tarkvara, tuleb see eelnevalt seadistada vastavalt enda programmi nõuetele. TempelPlus eeldab Java JDK/JRE alatest versioonist 6 ning Aladdin eToken-i tarkvara (Safenet Authentication Client), mida on võimalik saada Sertifitseerimiskeskuselt digitempli tellimisel.

Aladdin eToken-i tarkvara installeerimisel seadistatakse ka eTPKCS moodul, mille kaudu on ka teistel tarkvaradel võimalik suhelda krüptopulgaga, sealhulgas SK TempelPlusil. Aladdini eToken-i tarkvara käivitamisel saan kontrollida krüptopulga toimimist ja näha sellel olevaid sertifikaate. Samuti saan sertifikaate exportida, mida hiljem ka krüpteerimise näites kasutan.

Joonis 3 - Safenet Authentication Client

Kuna JdigiDoc on TempelPlus baasteek, siis tuleb konfigureerida ka JdigiDoc. Antud konfiguratsioonifailis jdigidoc.cfg kirjeldatakse logifailide asukoht ning kirjeldatakse aktsepteeritavad CA sertifikaadid, sh ka testsertifikaadid.

TempelPlus käsurea utiliidi kasutamiseks tuleb määrata tempelplus.conf failis jdigidoci kodukataloog. Samuti on vaja kirjeldada moodul (antud rakenduse puhul eTPKCS), mille läbi tempelplus suhtleb krüptopulgaga ning OCSP kehtivuskinnituse URL aadress (http://demo.sk.ee/ocsp), mis on mõeldud oma rakenduse testimiseks.

TempelPlus tarkvara kasutamise jaoks testsertifikaatidega, tuleb sertifikaadid eelnevalt eraldi paigaldada jdigidoc/lib kausta. [8]

3.3 TempelPlus testimine käsurealt

Kui seadistamine on tehtud, saame testida käsurealt digitempliga tehtavaid operatsioone. Antud rakenduse puhul testin dokumendi allkirjastamist, krüpteerimist ja dekrüpteerimist.

Dokumendi allkirjastamine asutuse digitempliga kasutame käsku *sign*, mis võtab parameetriteks dokumendi või kausta asukoha, mida soovime digitembeldada ning väljund, kuhu allkirjastatud dokumendid lähevad. Lisaparameetritena võib lisada veel –pin, mis on antud eTokeni parool, kuid mille antud rakenduse puhul olen seadistanud tempelplus.conf failis.

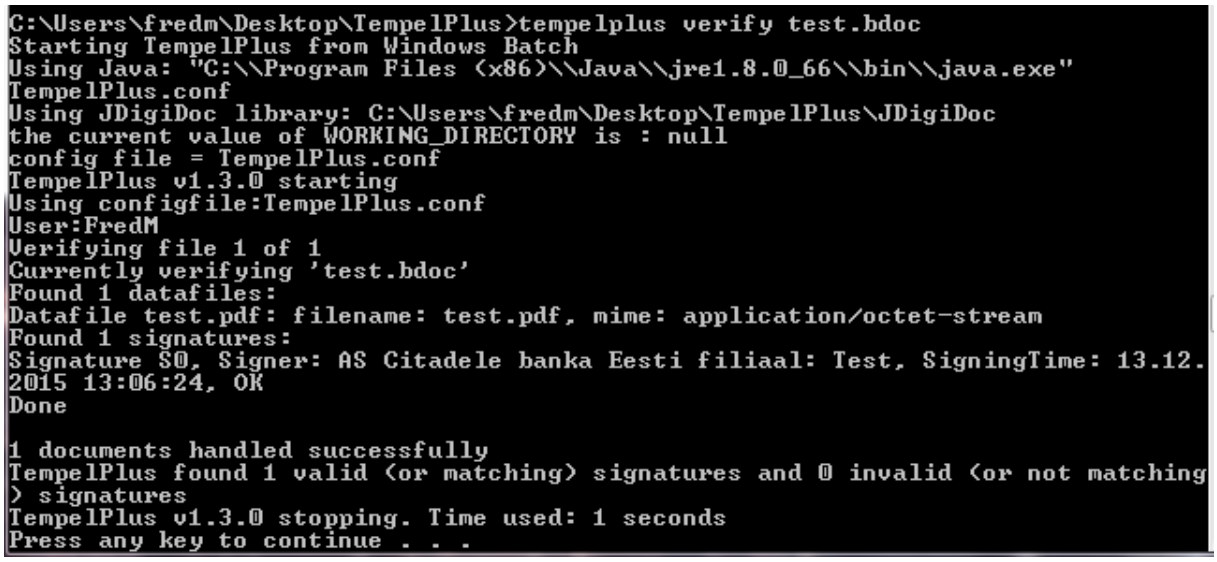
Allkirjastan faili test.pdf käsuga „**tempelplus sign test.pdf**“

```
C:\Users\fredm\Desktop\TempelPlus>tempelplus sign test.pdf
Starting TempelPlus from Windows Batch
Using Java: "C:\\Program Files (x86)\\Java\\jre1.8.0_66\\bin\\java.exe"
TempelPlus.conf
Using JDigiDoc library: C:\Users\fredm\Desktop\TempelPlus\JDigiDoc
the current value of WORKING_DIRECTORY is : null
config file = TempelPlus.conf
TempelPlus v1.3.0 starting
Using configfile:TempelPlus.conf
User:FredM
test.pdf

Reading pin from TempelPlus configuration file
Pin OK!
Executing operation with Corporate certificate
Signing file 1 of 1. Currently signing 'test.pdf'
File is not DigiDoc, converting..
Get conf: S0
Creating new container: C:\Users\fredm\Desktop\TempelPlus\test.bdoc
cleaning cache
Done
1 documents signed successfully
TempelPlus v1.3.0 stopping. Time used: 1 seconds
Press any key to continue . . .
```

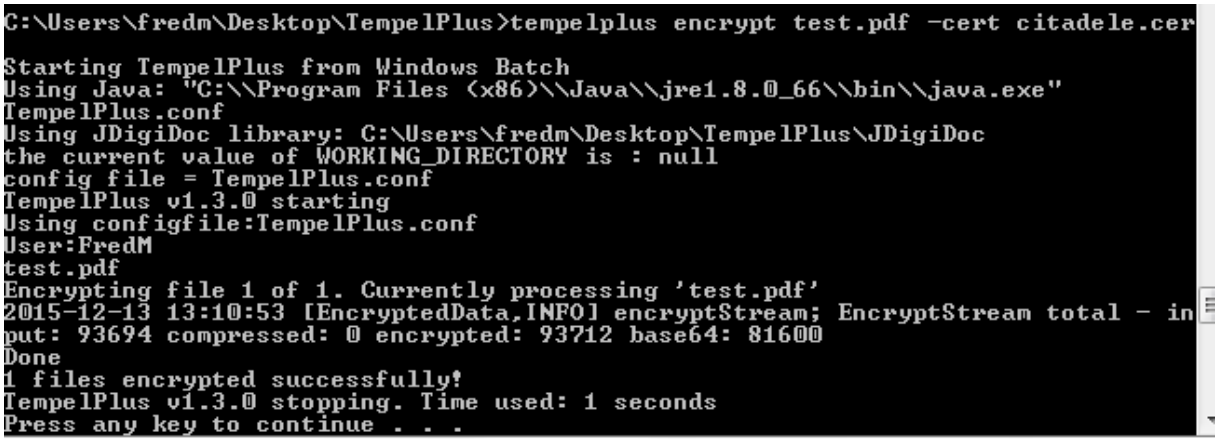
Joonis 4 - TempelPlus käsurida: allkirjastamine

Kasutades argumenti „verify“, võime näha allkirja valiidsust. Käsuga „**tempelplus verify test.bdoc**“



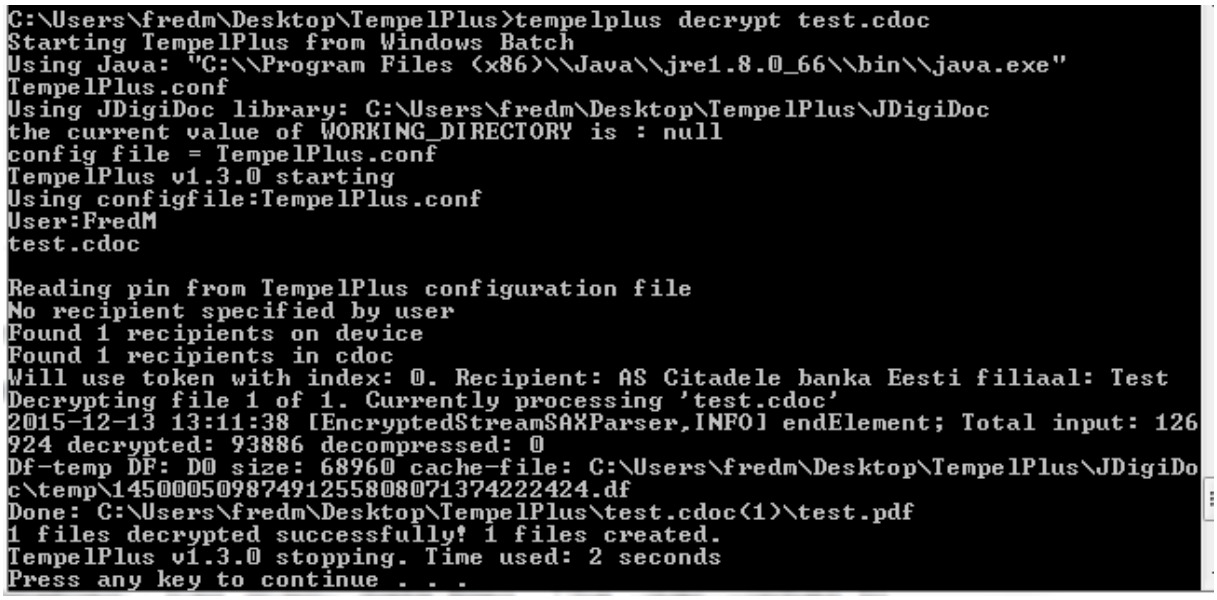
Joonis 5 - TempelPlus käsurida: allkirja valiidsuse kontrollimine

Nüüd krüpteerin antud dokumendi. Testimise eesmärgil krüpteerin selle iseendale. Selleks ekspordin krüptopulgal oleva sertifikaadi ning salvestan selle nimega 'citadele.cer'. Nüüd kasutades parameetrict –cert saan krüpteerida dokumendi käsuga „**tempelplus encrypt test.bdoc –cert citadele.cer**".



Joonis 6 - TempelPlus käsurida: krüpteerimine

Nüüd saan dekrüpteerida antud dokumendi käsuga „**tempelplus decrypt test.cdoc**".



Joonis 7 - TempelPlus käsurida: dekrüpteerimine

3.4 Kasutusjuhud

Kasutaja tuvastamine

Dokumendifailide üleslaadimine

Faili allalaadimine

Kasutaja seisundi muutmine

Uue kasutaja loomine

Ligipääsu lubamine/keelamine kasutajale

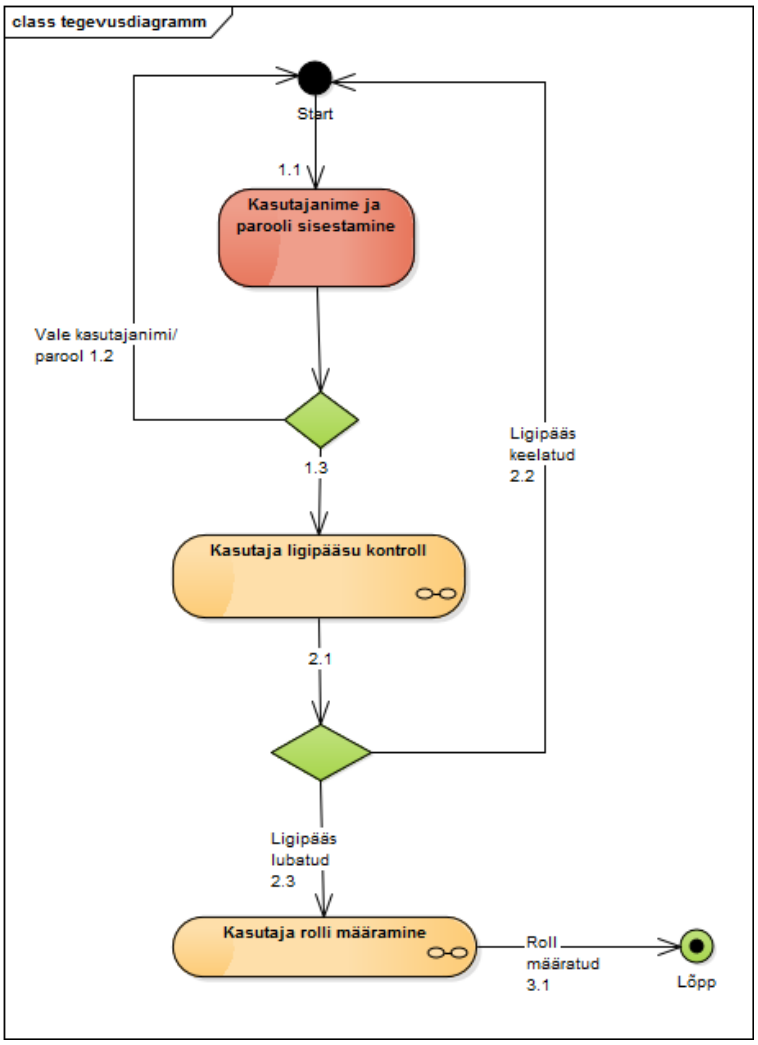
Tabel 1 - Infosüsteemi rollide kirjeldused

Rolli nimi	Kirjeldus
KASUTAJA	Kasutaja on töötaja, kellele rakenduse funktsionaalsus loodud on. Kasutaja teeb saab dokumente krüpteerida, dekrüpteerida ja allkirjastada ning seejärel konteineris sisalduvad failid allalaadida.
ANDMEBAASI ADMINISTRAATOR	Andmebaasi administraator on töötaja, kelle ülesandeks on tagada andmebaasi töö, viies läbi andmebaasi hooldust, jälgimist ja häälestamist. Tal on õigus vaadata kõiki

	andmeid.
ADMINISTRAATOR	Administraator saab teostada samasid operatsioone, mida kasutaja ning andmebaasi administraator, kuid lisaks saab administraator lisada süsteemi uue kasutaja.

3.5 Allsüsteemi dekrüpteerimine tegevusdiagramm

3.5.1 Sisse logimine



Joonis 8 - Sisse logimise tegevusdiagramm

1.1) Kasutaja sisestab oma kasutajanime ja parooli

1.2) Kasutajanimi ja/või parool ei sobi

1.3) Kasutajanimi ja parool on õiged.

2.1) Süsteem kontrollib andmebaasist, kas vastaval kasutajal on ligipääs süsteemi. Süsteemi ligipääsuks peab kasutaja olek vastama olekutüübile ’ACTIVE’.

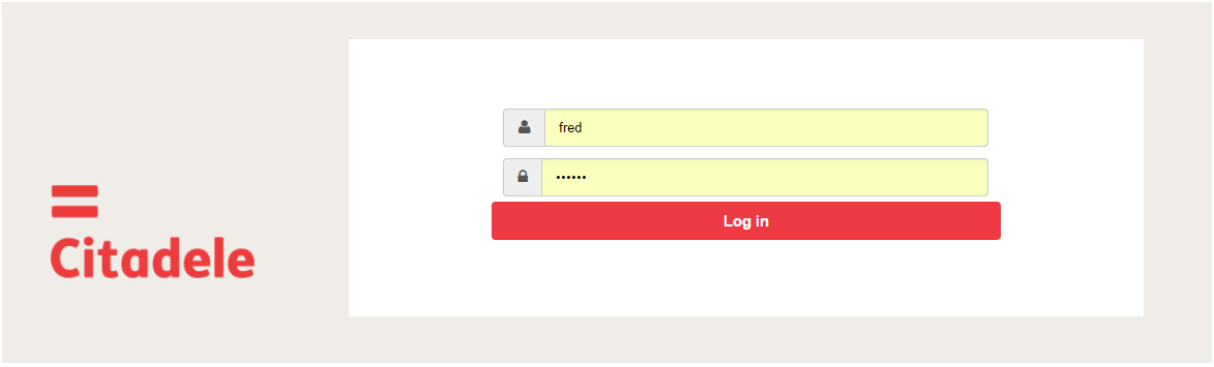
2.2) Kui kasutaja olek on midagi muud kui ’ACTIVE’ siis süsteem keelab kasutajale ligipääsu. Võimalikud olekud: active,inactive,locked,deleted. Inactive on olek, kus kasutajale on ajutiselt ligipääs keelatud. Locked on kasutaja ligipääsu keelamine mingi rikkumise tulemusel(parooli vale sisestamine). Deleted on kasutaja, kes ligipääsu enam ei vaja (inimene, kes enam ei tööta antud töökohal).

2.3) Ligipääsuks vajalikud toimingud on tehtud ning määratakse kasutaja roll.

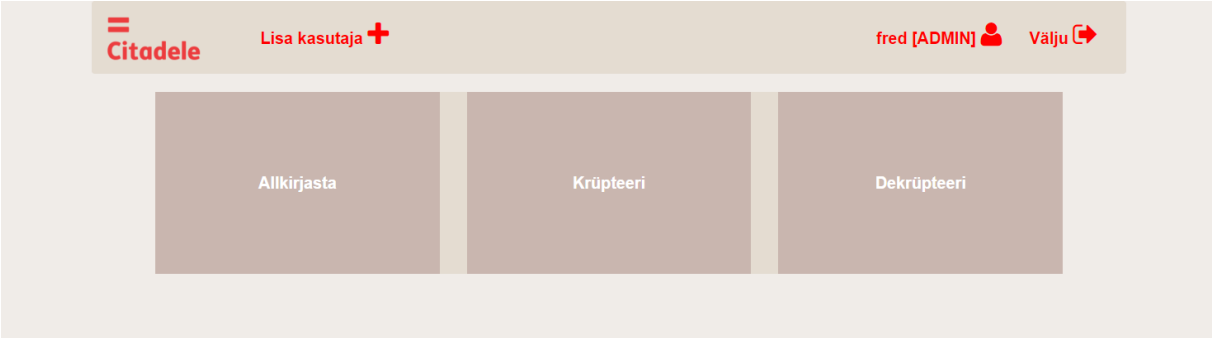
3.1) Kasutaja roll saab olla kas ’USER’,’DBA’ või ’ADMIN’ ning vastavalt sellele suunatakse ka kasutaja edasi.

Süsteemi logib kasutaja oma kasutajanime ja parooliga. Kasutajanime ja parooli saab luua administraator või andmebaasi administraator. Parool salvestatakse andmebaasi krüpteeritud kujul.

Antud rakenduse puhul kasutan Spring Security BcryptPasswordEncoder’it, mis on implementatsioon Bcrypt hash funktsiooni kasutavale Spring PasswordEncoder’ile.

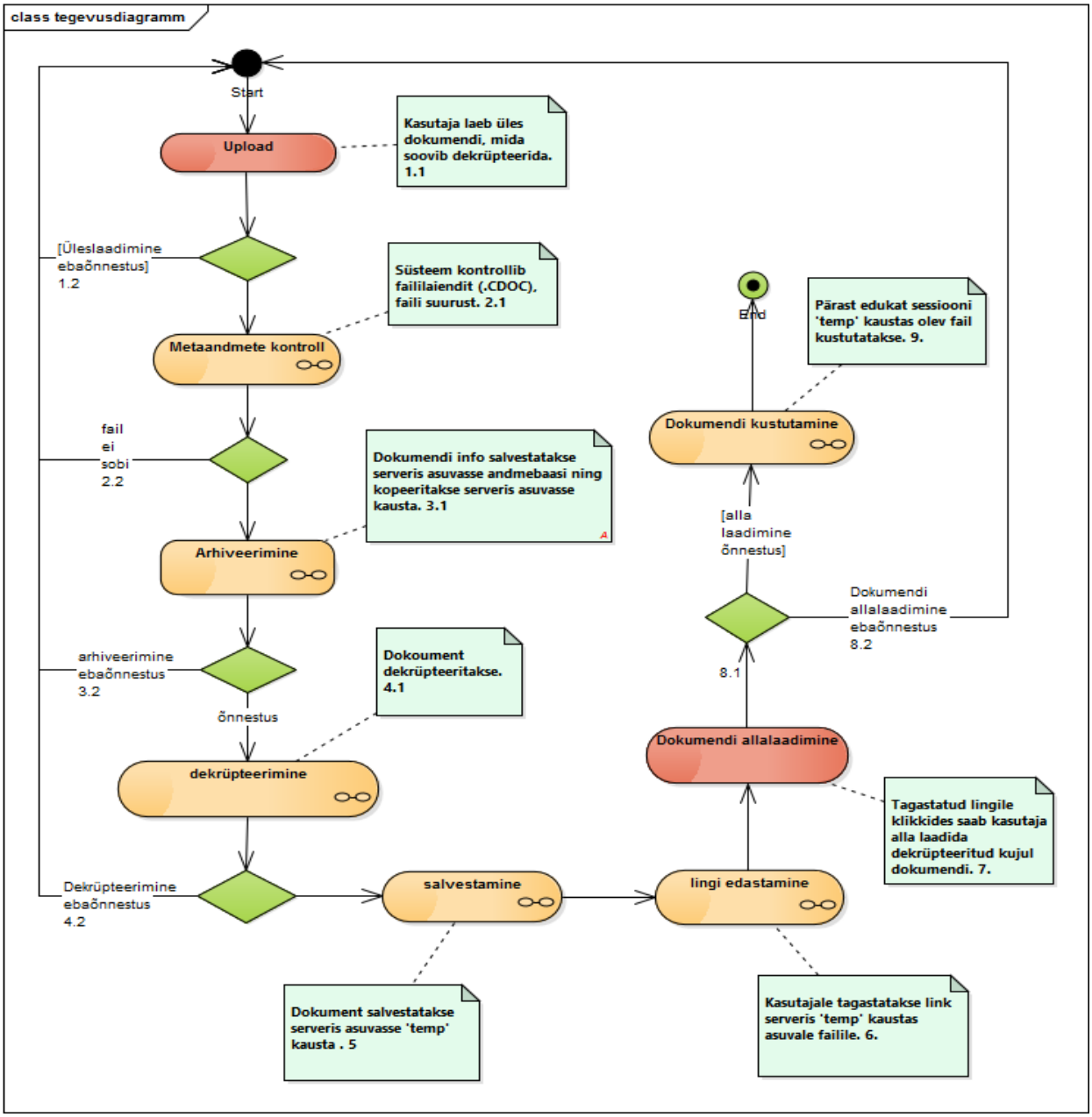


Joonis 9 - Rakendusse sisse logimine



Joonis 10 - Rakenduses operatsiooni valik

3.5.2 Dokumendi dekrüpteerimine



Joonis 11 - Dekrüpteerimise tegevusdiagramm

1.1) Kasutaja laeb üles dokumendi, mida soovib dekrüpteerida.

1.2) Kui mingil põhjusel dokumendi üleslaadimine ebaõnnestus, siis suunatakse kasutaja tagasi algusesse.

2.1) Faili metaandmete kontroll. Kontrollitakse, kas faili suurus sobib, kas faililaiend on .cdoc (krüpteeritud failide laiend).

2.2) Faili metaandmete kontrolli ebaõnnestumisel või faili andmete mittevastavusel suunatakse kasutaja algusesse ning tagastatakse vastav veateade. Samuti lisatakse selles punktis failile unikaalne tunnus, mis iseloomustab antud sessiooni (näiteks #123, mis viitab 123-ndale operatsioonile antud süsteemis). Info ebaõnnestunud faili üleslaadimisel kirjutatakse logisse.

3.1) Dokument arhiveeritakse. Dokument salvestatakse serverisse krüpteeritud kujul, et oleks võimalik hiljem kontrollida, millist faili üritati dekrüpteerida. Info faili ja faili asukoha kohta serveris salvestatakse andmebaasi.

4.1). Dokument dekrüpteeritakse. Süsteem pöördub tempelplus utiliidi poole ning dekrüpteerib ette antud dokumendi. Serveris asuvasse logifaili kirjutatakse antud operatsiooni kohta ka info koos info kasutaja kohta ning lisatakse teostatud toimingu aeg.

4.2) Kui süsteem ei saa mingil põhjusel dokumenti dekrüpteerida(puudub ühendus serveris olevate sertifikaatidega), siis süsteem lõpetab töö ning kirjutab ebaõnnestunud operatsioonist logifaili.

5) Dekrüpteeritud dokument salvestatakse ajutiselt serveris olevasse ’temp’ kausta, kus hoitakse dekrüpteeritud dokumenti kuni sessiooni lõppemiseni.

6) Kasutajale tagastatakse peale dekrüpteerimist link faili asukohta, milleks on ajutine kaust serveris.

7) Lingile klikkides saab kasutaja antud dokumendi alla laadida.

8.1) Kui alla laadimine õnnestus, siis dokument kustutatakse (9).

8.2) Kui alla laadimine ebaõnnestus, siis dokument kustutatakse ning kasutaja suunatakse algusesse.

9) Dokument kustutatakse serveris olevas ’temp’ kaustast.

Dokumendi dekrüpteerimiseks on programm käivitatud ooterežiimis. Sobiva faili ilmnemisel määratud kausta tuvastatakse see ning teostatakse automaatselt vajalik operatsioon. Seejärel nimetatakse dokument ümber unikaalse nimega ning kopeeritakse serverisse ning

dekrüpteeritud fail kopeeritakse ajutisse kausta, kust kasutaja dekrüpteeritud konteineri sisu saab alla laadida.

Joonis 12 - Dokumendi üleslaadimine

Dokumendi dekrüpteerimise kohta jääb info andmebaasi. Andmebaasist on võimalik näha:

- Kasutaja, kes dokumendi dekrüpteeris
- Kuupäev koos kellaajaga, millal toiming teostati
- Originaalse faili asukoht serveris

F_Id	created	name	size	type	url	fk_user_id
42	2015-12-20 23:29:12	1_test.cdoc	130219	cdoc	C:\Users\fredm\Desktop\server_folder\test.cdoc	1

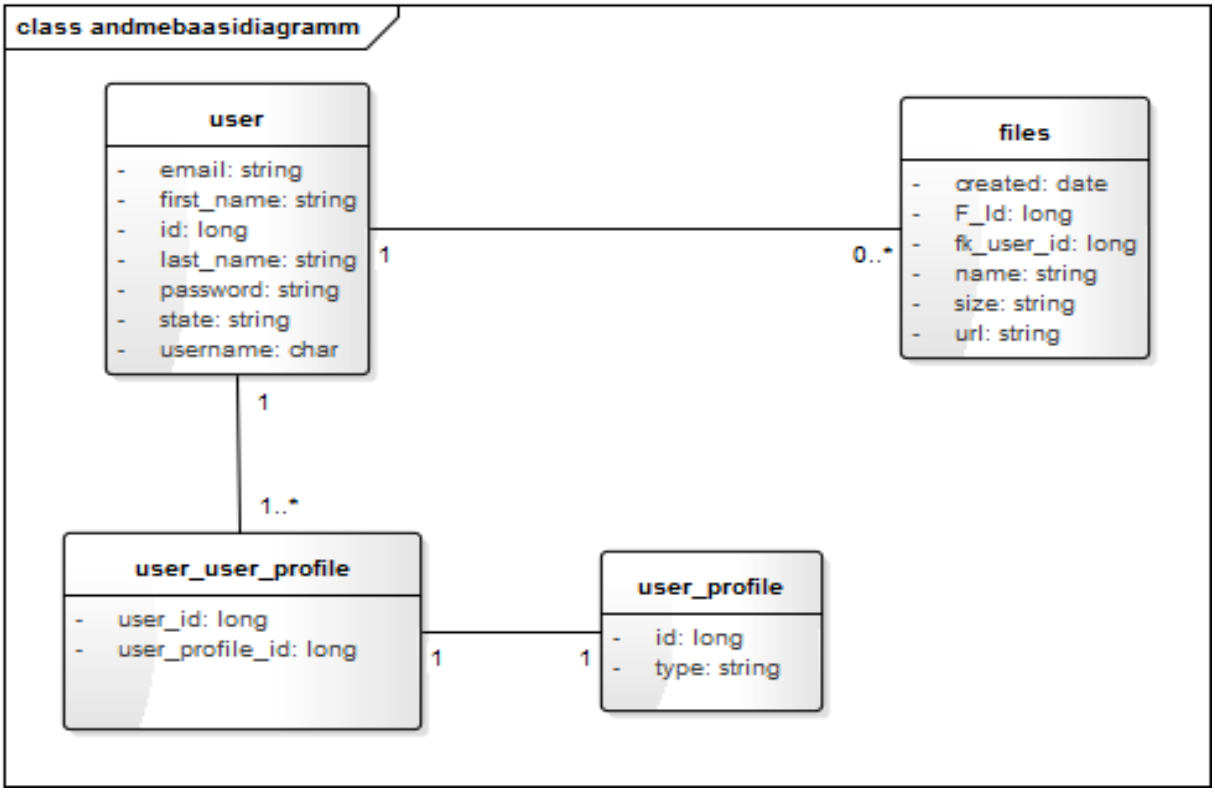
Joonis 13 - Andmebaas

Samuti kirjutatakse iga krüptopulgaga õnnestunud/ebaõnnestunud operatsioon eraldi logifaili.

```
2015-12-20 23:29:12 [PKCS11SignatureFactory,DEBUG] Open auth session for: AS
Citadele banka Eesti filiaal: Test id: 279ef04571e7d967 sign: false crypt:
true
2015-12-20 23:29:12 [PKCS11SignatureFactory,DEBUG] Login for: AS Citadele
banka Eesti filiaal: Test id: 279ef04571e7d967
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Get cert for token: 0
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Decrypting 256 bytes
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Decrypting with token: 0
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] session: Session Handle:
0x794000a
Token: Token in Slot: Slot ID: 0x0
Module: Module Name: C:\Windows\System32\eTPKCS11.dll
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Key 0 id: 656a4b243eb236d2
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Key 0 id: 279ef04571e7d967
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Using key 0 id:
279ef04571e7d967
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] decryptInit OK
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] value = [B@1a00fb9
2015-12-20 23:29:13 [PKCS11SignatureFactory,INFO] Decrypted 256 bytes, got: 16
2015-12-20 23:29:13 [PKCS11SignatureFactory,DEBUG] Closing card session
2015-12-20 23:29:13 [EncryptedStreamSAXParser,INFO] Total input: 126924
decrypted: 93886 decompressed: 0
2015-12-20 23:29:13 [Decrypt,INFO] Done: C:\Users\fredm\Desktop\temp_folder
\1_test.cdoc(1)\test.pdf
2015-12-20 23:29:13 [Decrypt,INFO] trying to delete file:1_test.cdoc
2015-12-20 23:29:22 [Decrypt,INFO] Decrypting file 1 of 1. Currently
```

Joonis 14– Logifail

3.6 Andmebaasi diagramm



Joonis 15 - Andmebaasi diagramm

3.6.1 Atribuutide kirjeldused

Tabel 2- Atribuutide kirjeldused

Olemitüübi nimi	Atribuudi nimi	Definitsioon	Näiteväärtus
USER	EMAIL	Elektroonilise posti aadress.	Fred.martmaa@gmail.com
USER	FIRST_NAME	Kasutaja eesnimi	Fred
USER	LAST_NAME	Kasutaja perenimi	Martmaa
USER	PASSWORD	Kasutaja parool. Kasutaja parool on krüpteeritud	Tamm1234

		kasutades Bcrypti.	
USER	STATE	Kasutaja olek, mis peab olema ’Active’, et kasutaja rakendusse sisse saaks logida.	Active
USER	USERNAME	Kasutajanimi, millega kasutaja süsteemi sisse logib.	fred
FILE	NAME	Dokumendi nimi, millega operatsiooni teostati. Sisaldab unikaalset identifikaatorit ning laiendit.	1_test.cdoc
FILE	SIZE	Faili suurus bittides.	130219
FILE	URL	Faili asukoht serveris.	C:\Users\fredm\Desktop\ server_folder\1_test.cdoc
FILE	CREATED	Kuupäev koos kellaajaga, millal operatsioon teostati. Formaat YYYY-MM-DD HH:MM:SS	2015-12-20 23:29:12

4. Rakenduse analüüs

4.1 Alternatiivsed lahenduskäigud

Kuna antud rakendus on loodud spetsiaalselt Citadele Banka Eesti filiaali nõuetele, siis sellist alternatiivi ei ole. Rakendus on eelkõige sellepärast, et oleks võimalik jälgida ja tõestada, kes millist dokumenti on näinud ja töödelnud.

Sertifitseerimiskeskuse poolt loodud TempelPlus käsureaprogramm on mõeldud eelkõige serverikeskkonnas käsureaprogrammina jooksutamiseks, näiteks pideva protsessina mingisse kindlasse kausta salvestatavate failide massiallkirjastamiseks/ krüpteerimiseks/ dekrüpteerimiseks. Selleks pannakse käima valitud protsess ooterežiimis ning jäädakse ootama faile, mis käsus määratud sisendkataloogi lisatakse.

Antud lahendus paraku jääb sellevõrra puudulikuks, et ei taga kontrolli kasutajate üle, kes tegelikult teatud failiga operatsiooni teostas. Samuti ei ole võimalik antud lahendusega piirata kasutajate ligipääsu üksteise poolt operatsioone teostatud failidele.

Kuna kõiki operatsioone, mida TempelPlus teeb, on võimalik teha JDigiDoc teeki otse kasutades, siis oleks olnud lahendus ka muuta Sertifitseerimiskeskuse poolt loodud rakendust piisavalt palju, et antud ooterežiimis töötav programm muuta massilisteks krüptooperatsioonideks ja töötleks määratud kausta laetud faile täpselt nii nagu soovime. Paraku oleks tegemist siis juba täiesti uue digitaalset templit võimaldava rakendusega TempelPlusi asemele, mis oleks olnud liiga mahukas ning samuti oleks see tunduvalt keerulisem kui kasutada olemasolevat lahendust.

Kolmas lahendus oleks veel rakendusest välja kutsuda vajalikke operatsioone, käivitades TempelPlus eraldi protsessina. Näiteks Javast otse käsureaprogrammi välja kutsudes.

```
final Runtime r;
r = Runtime.getRuntime();
Process process;
//käsu parameetrid
String[] cmdArray = new String[]{"cmd.exe", "/c",
        "tempelplus", "sign", "c:\\input","-output_folder", "c:\\output"};
process = r.exec(cmdArray , null, new File("working directory"));
//teises threadis kontrolli protsessi väljundit
/*....*/
process.destroy();
```

Joonis 16 - Javast käsurea poole pöördumine

Antud lahenduse kasuks siiski ei otsustanud, kuna inimesel, kes seda koodi loeb, on tõenäoliselt sellest raskem aru saada ning antud lahendus ei oleks välistanud kohustust luua siiski vajalik keskkond selle ümber.

Neljanda lahendusena pakun välja, et oleks võinud kutsuda TempelPlus *main* meetodi koos vajalike argumentidega, kuid kuna tegemist ei olnud ooterežiimis tegutseva lahendusega, siis oleks programm lõpetanud töö peale igat operatsiooni ning selle tulemusel poleks ka antud rakendus dokumendiga teostatud operatsioonist kaugemale jõudnud.

Andmebaasina langetasin otsuse MySQL kasuks, sest, programmi testimiseks kasutasin wamp serverit ning antud lahendus toetas MySQL andmebaasi. Sisuliselt kuna rakendus kasutab Hibernate ORM tehnoloogiat, mis toetab populaarsemaid andmebaase, võib kasutada ka PostgreSQL kui ka Oracle andmebaase.

4.2 Olemasolevad lahendused

Eesti ID poolt on loodud ID tarkvara, mis sisaldab ID-kaardi haldusvahendit, millega on võimalik kontrollida oma ID kaardi töötamist ja sertifikaatide kehtivust, muuta PIN ja PUK koode või vajadusel neid lahti blokeerida. Samuti sisaldab ID-kaardi tarkvara DigiDoc3 krüpto ehk krüpteerimise tarkvara ning DigiDoc3 klient ehk digiallkirjastamise tarkvara. ID-kaardi tarkvara saab kasutada nii asutuse digitempliga kui ka ID-kaardiga.

Arendajatele on loodud Eesti ID poolt näidisrakendus PHP-s, mis kasutab DigiDocService teenust. Antud näidisrakendus hõlmab konteineri loomist, andmefaili lisamist, eemaldamist, allkirjastamist ID-kaardi ja mobiili-IDga, allkirja eemaldamist ja allkirjade verifitseerimist. Antud Programm kasutab javascriptis loodud hwcrypto.js allkirjastamismoodulit. Siiski pidin langetama otsuse Java kasuks, kuna ettevõttes on põhilised rakendused tehtud Javas ning

seega on tulevikus antud rakenduse ühildamine teiste süsteemidega lihtsam ning tõrkeid esineb vähem.

4.3 Hinnang tehtud tööle

Antud rakendus kujunes mahukamaks ja keerulisemaks kui alguses hindasin. Keeruliseks tegi rakenduse tegemisel teadmatus, kuidas tegelikult taolised operatsioonid toimivad ning kuna tegemist on ainulaadse lahendusega, siis võttis iga etapp kauem aega.

Antud rakenduse ehitamise käigus tuli välja, et TempelPlus tarkvaral ei ole tuge otse krüpteeritud CDOC failide dekrüpteerimiseks ning ID-tarkvara versioon 3.10 krüpto on vaikimisi seadistatud otse krüpteerima. See tähendab, et ID-tarkvara 3.10 ja uuem DigiDoc3 krüpto toodab CDOC faile, mida TempelPlus ei oska dekrüpteerida. Hetkel ainuke alternatiiv TempelPlusi kasutajatele on igale krüpteerijale eraldi ütlema, et nad enne krüpteerimist seadistuse ära muudaksid.

Sarnane probleem Windows XP ja Windows Vista kasutajatel, sest nende viimane võimalik versioon ID-tarkvarast on 3.8.2, millele puudub samuti tugi. Samuti puudub tugi ka Windows 10 puhul, millega TempelPlus veel ei tööta.

Sertifitseerimiskeskus on probleemist teadlik ning tõenäoliselt on uuemal TempelPlus tarkvara versioonil probleem lahendatud.

4.4 Võimalikud arendusvaldkonnad

Antud rakendus on mõeldud edasiarenduseks ning planeeritud on ka täiendavat funktsionaalsust. Kuna rakenduse põhiline sihtgrupp ettevõttes oleks alguses ca 10 inimest, siis arvestades filiaali kiiret arengut, tuleb arvestada kasvava kasutusega.

Kindlasti tuleks teostada andmebaasi jaotamine vastavalt digitaalse templi operatsioonidele. See muudaks info otsimise efektiivsemaks ning kiiremaks. Samuti tuleks serveris olevaid faile sorteerida vastavalt kasutajate järgi individuaalsetesse kaustadesse. Antud lahenduse puhul on siis hea näha, milliseid faile keegi on käsitlenud.

Kasutajaliidesesse on vaja lisada vorm, mille läbi oleks uute kasutajate lisamine lihtsam. Antud rakenduse näitel on kasutajad lisatud otse andmebaasi. Lisaks on plaan lisada

kasutajaliidesesse võimalus administraatoril näha tehtud operatsioone ja teha päringuid kindlate parameetrite järgi.

Kuna pangasiseseid programme on palju ning paroole, mida meeles pidada lisandub aina juurde, siis on plaanis teha antud rakendusse sisselogimine ID-kaardi põhiseks, mis jätaks parooli unustamise korral alternatiivse lahenduse. Kolmas lahendus oleks siduda rakendus domeeni Active Directory'ga ning teostada rakendusse sisselogimine vastavalt arvutis sees oleva domeenikasutajaga. Lisada oleks vaja veel kasutajapoolne võimalus seadistada enda profiili ning läbi selle ka vahetada oma parooli. Turvalisuse huvides on kindlasti ka lisada andmebaasi väli viimati muudetud parooli kohta, mis sunniks kasutajat mingi teatud perioodi tagant oma parooli vahetama.

Lisaks on plaanis arendada kasutajaliides veakindlamaks, mis tähendab kasutajapoolsete valikute (digitempli operatsioonide valiku) eemaldamist ning lisades süsteemipoolse faili tuvastuse. Selline lahendus suudab ise tuvastada üleslaetud faili põhjal, millise operatsiooni juurde kasutaja edasi suunata.

Suurema kasutuse korral võib tekkida olukord, kus ühte krüpteeritud faili sisu on vaja näha rohkemal kui ühel kasutajal. Seega peaks olema võimalus failide linke turvaliselt jagada teiste kasutajatega. Selline lahendus eeldab, et kasutajatel on võimalus laadida alla faile otse serverist ning andmebaasi tuleks juurde lisada grupid koos õigustega, mille läbi siis ka failid serveris jaotatud on. See tagab võimaluse sorteerida, millisel grupil millistele kaustadele õigused on. Keerulisem edasiarendus oleks rakenduse sidumine domeeniga ning pärida läbi Active Directory kasutajate õiguseid.

5. Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua AS Citadele Banka Eesti filiaalile rakendus sissetulevate failide krüpteerimiseks, dekrüpteerimiseks ja allkirjastamiseks ning arhiveerida töödeldud failid. Rakendus on loodud vastavalt AS Citadele Banka Eesti filiaali poolt esitatud nõuetele ning on kasutatav ainult selle asutuse poolt.

Antud töö raames uuriti Sertifitseerimiskeskuse poolt loodud TempelPlus rakendust vastavate digioperatsioonide tegemiseks ning krüpteerimise protsessi. Samuti uuriti erinevaid failitüüpe, mida antud operatsioonide käigus töödeldakse.

Autori poolt seatud eesmärk sai täidetud ning bakalaureusetöö raames valmis ka rakendus failide arhiveerimiseks, mida seoti Sertifitseerimiskeskuse poolt loodud TempelPlus käsureaprogrammiga, mis võimaldab dokumente krüpteerida, allkirjastada ning dekrüpteerida.

Antud bakalaureusetöö raames loodud rakendus täidab peamist funktsionaalsust, kuid vajab edasiarendust parema kasutajaliidese osas ning on arvestatud ka tulevase lisafunktsionaalsusega.

Summary

The purpose of this thesis was to create an application for AS Citadele Banka for document encryption, decryption and signing and to archive those files. Application is created according to functional requirements given by AS Citadele Banka and can only be used by that institution.

Main subjects analyzed in this thesis were application for digital operations by Sertifitseerimiskeskus and investigation of different file types used in those operations.

The aim of this thesis was accomplished and as a result of this thesis an application for archiving files that were processed using TempelPlus command line utility which allows files decryption, encryption and signing was created.

The created application fills the main functionality but needs to be improved by better user interface and by future extra functionality.

Kasutatud kirjandus

[1] Diffie, W., & Hellman, M. (6. November 1976. a.). *New Directions in Cryptography.* Allikas: Invited Paper: https://ee.stanford.edu/~hellman/publications/24.pdf

[2]*Digitaalne tempel*. (kuupäev puudub). Allikas: Sertifitseerimiskeskuse koduleheküIg: https://www.sk.ee/teenused/digitempli-teenus

[3] *eid.eesti.ee - failivormingud*. (kuupäev puudub). Allikas: https://eid.eesti.ee/index.php/%C3%9Cldteavet_arendajatele#CDOC

[4] Gutmann, P. (October 2000. a.). X.509 Implementation notes. *an Overview of PKI.*

[5] *ID koduleheküIg*. (kuupäev puudub). Allikas: http://www.id.ee/

[6] Kasten, J., Durumeric, Z., Bailey, M., & Halderman, A. (2013). Analysis of the HTTPS Certificate Ecosystem. *The Internet Measurement Conference.* Detroit: SIGCOMM.

[7] *SafeNet koduleheküIg*. (kuupäev puudub). Allikas: http://www.safenet-inc.com/multi-factor-authentication/authenticators/pki-usb-authentication/etoken-pro/

[8] Sinivee, V. (29. June 2015. a.). *JDigiDoc Programmer's guide.* Allikas: http://id.ee/public/SK-JDD-PRG-GUIDE.pdf

[9] *Tallinna Ülikooli Krüptograafia loeng*. (kuupäev puudub). Allikas: https://www.tlu.ee/~matsak/crypto/loeng9.pdf

[10] *Tartu Ülikooli Infoturve*. (kuupäev puudub). Allikas: https://courses.cs.ut.ee/2013/infoturve/fall/Main/PKIEhkAvalikuV%C3%B5tmeInfrastruktuur

[11] *TempelPlus - Digitembeldamise tarkvara*. (kuupäev puudub). Allikas: Sertifitseerimiskeskuse koduleheküIg: https://www.sk.ee/teenused/digitempli-teenus/tempelplus/

[12] What is a Public Key Infrastructure? (17. April 2015. a.). *A Simple Overview*, lk 1.