KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Paulius Jasas

E-SĄSKAITŲ SISTEMOS SUKŪRIMAS

Baigiamasis magistro darbas

Vadovas
prof. dr. Eligijus Sakalauskas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

E-sąskaitų sistemos sukūrimas

Baigiamasis magistro darbas
Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
Prof. dr. Eligijus Sakalauskas

Recenzentas
Dr. Aleksejus Michalkovič

Projektą atliko
Paulius Jasas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Paulius Jasas

(Studento vardas, pavardė)

„Informacijos ir informacinių technologijų sauga“ (621E10003)

(Studijų programos pavadinimas, kodas)

„Baigiamojo projekto pavadinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 17 m. gegužės 22 d.

Kaunas

Patvirtinu, kad mano **Pauliaus Jaso** baigiamasis projektas tema „E-sąskaitų sistemos sukūrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka) *(parašas)*

Jasas, P. „E-sąskaitų sistemos sukūrimas“. Magistro baigiamasis projektas / vadovas prof. Dr. Eligijus Sakalauskas; Kauno technologijos universitetas, Informatikos fakultetas, Kompiuterių katedra.

Kaunas, 2017. 62 p.

SANTRAUKA

Verslo operacijose vis dažniau naudojant elektroninius dokumentus, iškyla poreikis tų dokumentų pasirašymui ir dokumentų patikrinimui. Taip pat šiuo metu darbo kultūra paremta efektyvumu, lankstumu ir mobilumu. Didesnis greitis pasiekiamas naudojant mobilius telefonus, nes jie visada yra nešiojami kartu su savimi. Todėl šiame darbe siekiama įgyvendinti sistemą, kuri galėtų pasirašyti bei patikrinti parašus ant elektroninių bei atspausdintų dokumentų ir tai atlikti naudojantis mobiliais įrenginiais.

Darbui pasiruošti buvo išanalizuotas e. dokumentų sistemų veikimas, skirtingų kriptografinių algoritmų ypatybės, *QR* kodų duomenų talpinimo galimybės, parašų laikymo dokumentuose taisyklės ir kita su dokumentų pasirašymu susijusi informacija. Ištirtos kitos panašios sistemos parodė, kad stengiamasi didinti sistemų integravimą su mobiliais įrenginiais, tačiau patikrinus pasirašymo ir dokumentų tikrinimo galimybes matoma, kad trūksta sistemų, kurios tai galėtų daryti greitai ir efektyviai. Parašams kurti pasirinkti *ECDSA* 256 bitų ilgio raktai, leisiantys pasirašyti dokumentus, užtikrinant parašo saugumą ir minimizuojant sukuriamos informacijos kiekį. Kuriama sistema remiasi *QR* kodo sugebėjimu talpinti informaciją mažame plote. Jame užšifruojama parašo patikrinimui reikalinga informacija, o tada mobilusis įrenginys skanuodamas *QR* kodą gali patikrinti šį parašą, ar tikrai dokumentą pasirašė tas asmuo, kuris ir turėjo. Nuskanavus dokumentą suteikiama galimybė pačiam padėti parašą ant šito dokumento. Pats dokumentas laikomas serveryje, todėl pasirašymui ir pilnam patikrinimui ši sistema remiasi serverio ir mobilaus įrenginio sugebėjimu komunikuoti tarpusavyje.

Pirmo tyrimo rezultatai parodo *ECDSA* rakto pranašumą, sukuriant mažesnį *QR* kode šifruojamų duomenų kiekį taip sutaupant vietos dokumente. Kituose tyrimuose atliktas *QR* kodų skanavimas pademonstruoja *QR* kodo dydžio ribas, kad jis dar būtų nuskaitomas ir būtų įmanoma patikrinti dokumento parašą.

Jasas, Paulius. E-Invoices Management System: Master‘s thesis in “Information and Information Technology Security” / supervisor prof. dr. Eligijus Sakalauskas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: e-invoices signing with e-signatures using mobile devices

Key words: e-invoice, digital signature, XAdES, QR code

Kaunas, 2017. 62 p.

SUMMARY

As more and more e-documents are being used in business operations, there is a growing need for the signing of those documents and checking signatures. Additionally, nowadays, work culture relies on effectiveness, flexibility and mobility. Higher mobility and flexibility are achieved by using mobile devices, which are always by our side. As a result, the goal of this work is to create a system which could sign and check signatures on electronic and printed documents while using mobile devices for that.

In order to prepare for work it was firstly analyzed e-document systems working principles, different cryptography algorithms properties, signature storing in a document rules and other data related to working with e-signatures. The analysis of other similar systems, has shown that there is an effort put into integrating mobile devices with such systems, although after checking the capability of signing and checking documents it becomes apparent that there is a lack of systems which could do those things rapidly and effectively. For creating signatures, it was chosen to use *ECDSA* 256 bit long keys which allow to sign documents ensuring enough security and minimizing the amount of data created while signing. This system relies on the capability of *QR* codes to fit information in small space. Data necessary for checking the signature is encoded in a *QR* code and then mobile device scans the *QR* code allowing an app to perform signature validity checking. After scanning, and ability to sign the document is granted. The document itself is held in the server, so the whole performance relies on the capability of communication between the server and mobile device.

Results of the first experiment show the superiority of *ECDSA* key when creating less data for encoding into *QR* code thus saving space in the document. *QR* code scanning in other experiments has shown the limits of *QR* code where it still can successfully read data and check the signature.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

E. dokumentas – elektroninis dokumentas

E. parašas – elektroninis parašas

EDVS – elektroninių dokumentų valdymo sistema

DVS – dokumentų valdymo sistema

PDF (angl. *Portable Document Format*) – portatyvaus dokumento formatas

QR kodas (angl. *Quick Response code*) – greito atsakymo dviejų dimensijų brūkšninis kodas

Šifrograma – užšifruotas tekstas

MAC (angl. *Message Authentication Code*) – pranešimo autentifikavimo kodas

NIST (angl. *National Institute of Standards and Technology*) – Nacionalinis standartų ir technologijos institutas JAV

XML (angl. *Extensible Markup Language*) – bendrosios paskirties duomenų struktūrų bei jų turinio aprašomoji kalba

XMLDSIG (angl. *XML Signature*) – XML tipo sintaksė naudojama elektroninių parašų laikymui faile

AdES (angl. *Adva*nced Electronic Signatures) – pažangūs elektroniniai parašai, kurie gali būti laikomi įvairiais formatais (pvz., XAdES)

XAdES (angl. *XML Advanced Electronic Signatures*) – pažangūs XML elektroniniai parašai

PAdES (angl. *PDF Advanced Electronic Signatures*) – pažangūs elektroniniai parašai PDF dokumentams.

CAdES (angl. *CMS Advanced Electronic Signatures*) – papildiniai kriptografinių žinučių sintaksei, pažangių elektroninių parašų įgyvendinimui

MIME (angl. *Multipurpose Internet Mail Extensions*) – interneto standartas išplečiantis el. pašto protokolą, kuris leidžia dalintis įvairaus tipo failais internetu

SF – sąskaita faktūra

URL (angl. *Uniform Resource Locator*) – internetinio tinklalapio adresas

SSL (angl. *Secure Socket Layer*) – dar kitaip vadinamas TLS tai yra kriptografinis protokolas, numatantis apsaugotą duomenų perdavimą tarp mazgų pasauliniame kompiuterių tinkle internete

HTTP (angl. *Hypertext Transfer Protocol*) – pagrindinis metodas informacijai pasauliniame tinkle (WWW) pasiekti

JSON (angl. *JavaScript Object Notation*) – atviro standarto formatas perduodantis duomenų objektus

ĮVADAS

Šiuo metu vis labiau kompiuterizuojamos visos gyvenimo sritys. Ne išimtis ir verslo procesai. Vykdant transakcines operacijas būna naudojamos sąskaitos faktūros todėl, šiuos, dažnai popierinius, dokumentus galima paversti į elektroninį formatą ir sutaupyti laiko, pinigų ir gauti kitokios papildomos naudos [1]. Elektroniniai dokumentų pasirašymui naudojami elektroniniai parašai, kurie yra sunkiau suklastojami nei ranka rašomas parašas (jeigu pasirinktas rekomenduojamas saugus kriptografinis metodas), o klastotes aptikti yra lengviau [2]. Kadangi dabar mobilieji telefonai populiarūs kaip niekada anksčiau, jų panaudojimas dokumentų valdyme gali pagreitinti verslo procesus, paspartinant dokumentų dalijimąsi, nes mobilieji įrenginiai visada nešiojami šalia žmogaus.

Šis darbas priklauso Informacijos ir informacinių technologijų saugos studijų programai. Jame bus aprašoma e-sąskaitų sistemos sukūrimas ir jos veikimo principai. Šiame darbe aprašoma sistema turėtų būti aktuali įmonėms, kurios nori suteikti galimybę pasirašinėti sąskaitas faktūras savo klientams nenaudojant popieriaus, o tiesiog greitu būdu tai atlikti naudojant mobilius įrenginius. Pagrindiniai vartotojai bus verslo įmonės bei jų klientai, kurie dirba su tokio pobūdžio dokumentais.

Darbo problematika ir aktualumas

Internete galima rasti sistemų, kurios siūlo e. dokumentų pasirašymą ir parašų tikrinimą. Dauguma iš jų net negali pasiūlyti kriptografinių raktų naudojimo. Jaučiamas trūkumas tokių sistemų, kurios turėtų galimybė vykdyti šiuos veiksmus naudojant mobilųjį įrenginį. Taip pat nors yra sistemų tikrinančių kriptografinius parašus e. dokumentuose, tačiau popieriniai dokumentai niekur nedings, todėl negalima jų palikti nuošaly, nes jie užima vis dar svarbią dalį verslo pasaulyje. Reikia metodo, kuris galėtų atlikti sąskaitų faktūrų išrašymą ir jų pasirašymą elektroniniais parašais, su galimybe atvaizduoti ir patikrinti parašus tiek ant e. dokumento, tiek ant popieriaus lapo. Visa tai turi būti galima atlikti mobiliuoju įrenginiu, nes jie yra plačiai išplitę, o jų naudojimas gali paspartinti verslo procesus.

Darbo tikslas ir uždaviniai

Sukurti e-sąskaitų faktūrų sistemą, įgyvendinančią pasirašymą skaitmeniniais kriptografiniais parašais ir jų tikrinimą su mobiliaisiais įrenginiais.

Tam tikslui įgyvendinti reikės atlikti šiuos uždavinius:

1) atlikti probleminės srities analizę;
2) suprojektuoti sistemos struktūrą, parodančią skirtingų dalių sąveiką;
3) išrinkti tinkamą kriptografinį metodą parašų generavimui;
4) pritaikyti *QR* kodų informacinę struktūrą parašo informacijai saugoti;
5) sukurti sistemos prototipą, remiantis suprojektuota struktūra ir pasirinktais kriptografiniais raktais;

6) atlikti parašo duomenų dydžio priklausomybės nuo kriptografinių raktų, jų ilgio ir *QR* kodo nuskaitomumo tyrimus bei aprašyti rezultatus.

Darbo rezultatai ir jų svarba

Sukurta e-sąskaitų faktūrų parašų tikrinimo ir pasirašymo sistema suteikianti galimybę klientams išrašyti bei pasirašyti sąskaitas faktūras pagal pasirinktą šabloną, taip pat patikrinti parašų teisėtumą, naudojant mobilius įrenginius. Dokumentų valdymas virtualioje erdvėje leis greitai persiųsti dokumentus, atlikti paiešką, greitą parašų teisėtumo tikrinimą, taip sutaupant laiką ir pinigus klientams.

Darbo struktūra

Šį darbą sudaro trys pagrindiniai skyriai.

Pirmajame skyriuje yra patiekta visos probleminės srities analizė, išanalizuojami kriptografiniai raktai, el. parašų standartai ir palyginami šiuo metu rinkoje esantys produktai. Skyriaus tikslas yra paruošti reikalingą informaciją sistemos projektavimui, išanalizuoti technologijas, kurios bus naudojamos projekte.

Antrajame skyriuje pateikiamas sukurtas sistemos projektas, kuriame pateikiamos sistemos architektūros, panaudojimo atvejų, veiklos diagramos su paaiškinimais. Skyriuje atskleidžiamas naudojamas kriptografinis algoritmas, pateikiama *QR* kodo struktūra.

Trečiame skyriuje aprašytas sukurtas prototipas ir naudojami įrankiai jo kūrimui. Taip pat pateikiami tyrimai, kurie parodo, kuriamos sistemos panaudojimo galimybes ir metodų veiksmingumą.

Darbo pabaigoje pateikiamos išvados, apibendrinančios viso darbo rezultatus, panaudotos literatūros sąrašas. Prieduose pridedamas skyrelis, kuriame pateikiamas sukurto prototipo aprašymas su vartotojo sąsaja.

1. PROBLEMINĖS SRITIES ANALIZĖ

1.1. Analizės tikslas

Šios analizės tikslas – paruošti informaciją ir metodą sistemos įgyvendinimui. Suprasti sistemą sudarančių technologijų, protokolų veikimą ir kaip juos panaudoti, kad jų sąveika pateiktų norimą rezultatą.

1.2. Tyrimo objektas, sritis ir problema

El. sąskaitų faktūrų pasirašymas integruotoje el. parašo sistemoje. Patogaus, saugaus, įrankio, kuris leistų atlikti veiksmus su sąskaitomis faktūromis ir elektroniniais parašais, internetu trūkumas.

1.3. E-dokumentai, dokumentų valdymo sistemos

Elektroniniai dokumentai iš esmės yra bet kokia elektroninė informacija ar kitaip sakant failai (neskaitant programinės įrangos ar sisteminių failų), kurie yra skirti vaizduoti ar kaupti elektroninėje formoje ar gali būti atspausdinti popieriaus lape. Anksčiau, bet kokia skaitmeninė informacija buvo laikoma tik vidine kompiuterio dalimi, kurios turinį pamatyti atpažįstamame formate vienintelis būdas yra atspausdinti ją ant popieriaus. Tačiau kompiuterinių tinklų tobulėjimas lėmė, jog dabar daugumoje atvejų yra patogiau dalintis elektroniniais dokumentais nei spausdintais. Taip pat patobulėjimas atvaizdavimo srityje reiškia, kad dabar norint pamatyti dokumento turinį galima jį tiesiog atvaizduoti ekrane užuot spausdinus ant popieriaus lapo taip sutaupant popieriaus ir vietos reikalingos laikyti atspausdintus popierius. Be to einant laikui ypač per pastaruosius 50 metų išryškėjo tendencija, kad dokumentai gali būti ne tik saugomi ir kaupiami, bet jais gali būti dalinamasi, jie gali būti publikuojami, o tam tikslui pasiekti dokumentų skaitmenizavimas labai stipriai gali padėti.

Esant dideliam dokumentų kiekiui juos tvarkyti gali pasidaryti sunkoka ir nepatogu, jeigu jie yra atspausdinti ir laikomi fiziniame lygmenyje. Per pastaruosius 20-30 metų, kai kompiuterinė technika ištobulėjo pakankamai, kad kompiuterių naudojimui nebereikia programavimo žinių, gamybos ir paslaugų įmonių darbuotojų skaičius naudojančių kompiuterius darbe, 2016 metais siekė 43,5%, o įmonių dalis siunčiančių ir gaunančių elektronines sąskaitas faktūras siekė atitinkamai 60,1% ir 82,4% [3]. Tai rodo pakankamai paplitusį elektroninių dokumentų naudojimą darbinėje aplinkoje. Tokių dokumentų valdymui bei priežiūrai pasitelkiami kompiuteriuose veikiantys programinės įrangos paketai vadinami elektroninių dokumentų valdymo sistemomis.

EDVS – elektroninių dokumentų valdymo sistema yra programinė įranga, kuri leidžia kurti, laikyti, archyvuoti ir valdyti dokumentų turinį elektroniniu būdu. Pirminė DVS funkcija yra valdyti informaciją organizacijos veiklos procesuose. Bet kuri bazinė DVS turėtų leisti dokumentų valdymą(turinio kontroliavimą) ir jų versijavimą, teksto redagavimą, failų archyvavimą ir

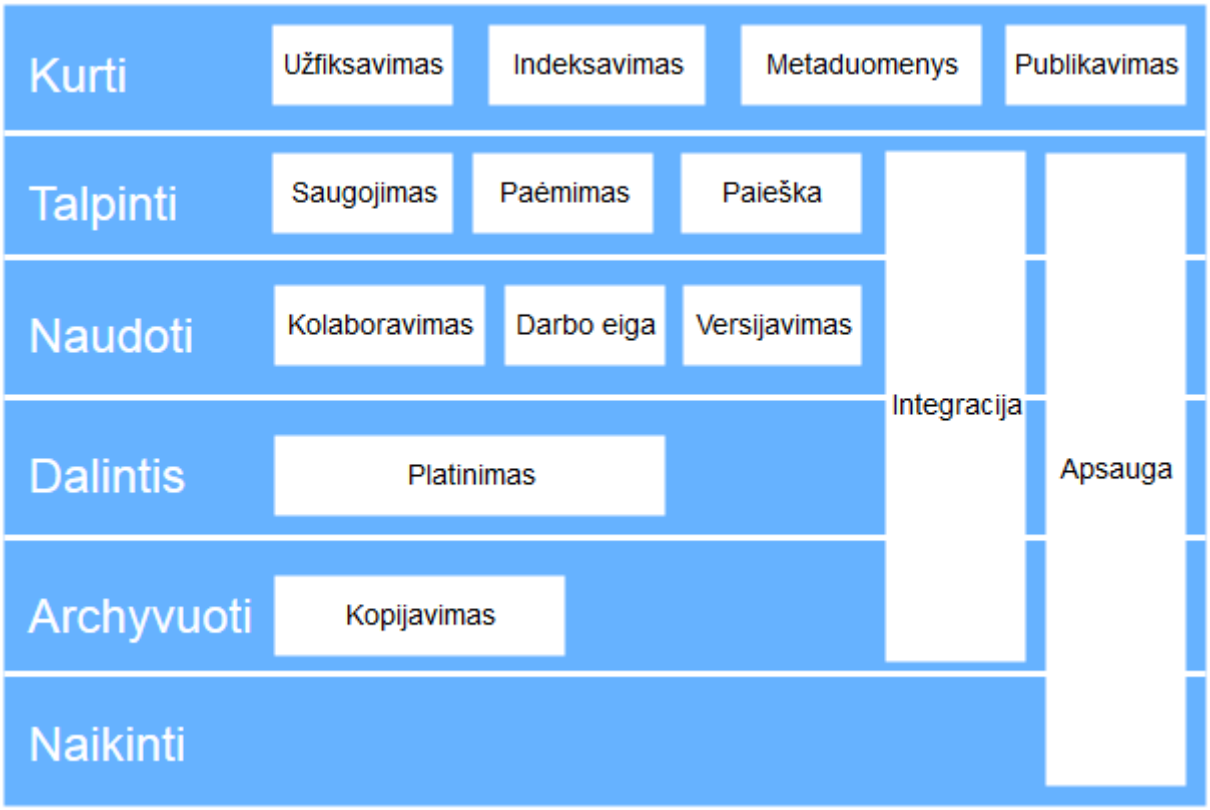
atvaizdavimą. DVS taip pat privalo galėti užtikrinti saugų priėjimą išsaugant failų vientisumą ir įvairių failo disponavimo funkcijų vykdymo galimybę visiems sistemoje esantiems failams. [4]

Dokumentų valdymo supaprastinimas gali būti vienas iš kertinių rūpesčių organizacijai nepaisant jos dydžio. Šių dienų vis griežtinamų taisyklių ir standartų aplinkoje atitinkamas dokumentų valdymas nesvarbu ar jis yra popierinis ar elektroninis gali išnaudoti daug pinigų ir laiko, kurie galėtų būti geriau panaudoti kitais tikslais. Įgyvendinant tokią sistemą organizacijos gali pastebimai padidinti atliekamų procesų efektyvumą. Tokios sistemos gali jums padėti:

- taupant pinigus;
- taupant laiką;
- padidinant efektyvumą;
- pagerinant komunikaciją tarp atskirų organizacijoje esančių veiklos vienetų(pvz. skyrių);
- įgalinant automatizavimą. [5]

Pagal *Gartner Group* atliktą tyrimą dokumentų valdymo sistemos įdiegimas ir naudojimasis organizacijai gali su dokumentais susijusias išlaidas gali padėti sumažinti net iki 40%. [6]

Visos EDVS sistemos turi keletą bendrų komponentų tokių kaip metaduomenys, integracija, užfiksavimas, saugojimas, paėmimas, platinimas, apsauga, darbo eiga, kolaboravimas, versijavimas, paieška, publikavimas, ir kopijavimas. Jie persidengia su šešiomis informacijos gyvenimo ciklo fazėmis: kurti, talpinti, naudoti, dalintis, archyvuoti, naikinti. Kaip tai atrodo galima pamatyti 1 pav.



1 pav. Informacijos gyvenimo ciklo ir DVS komponentų sulyginimas

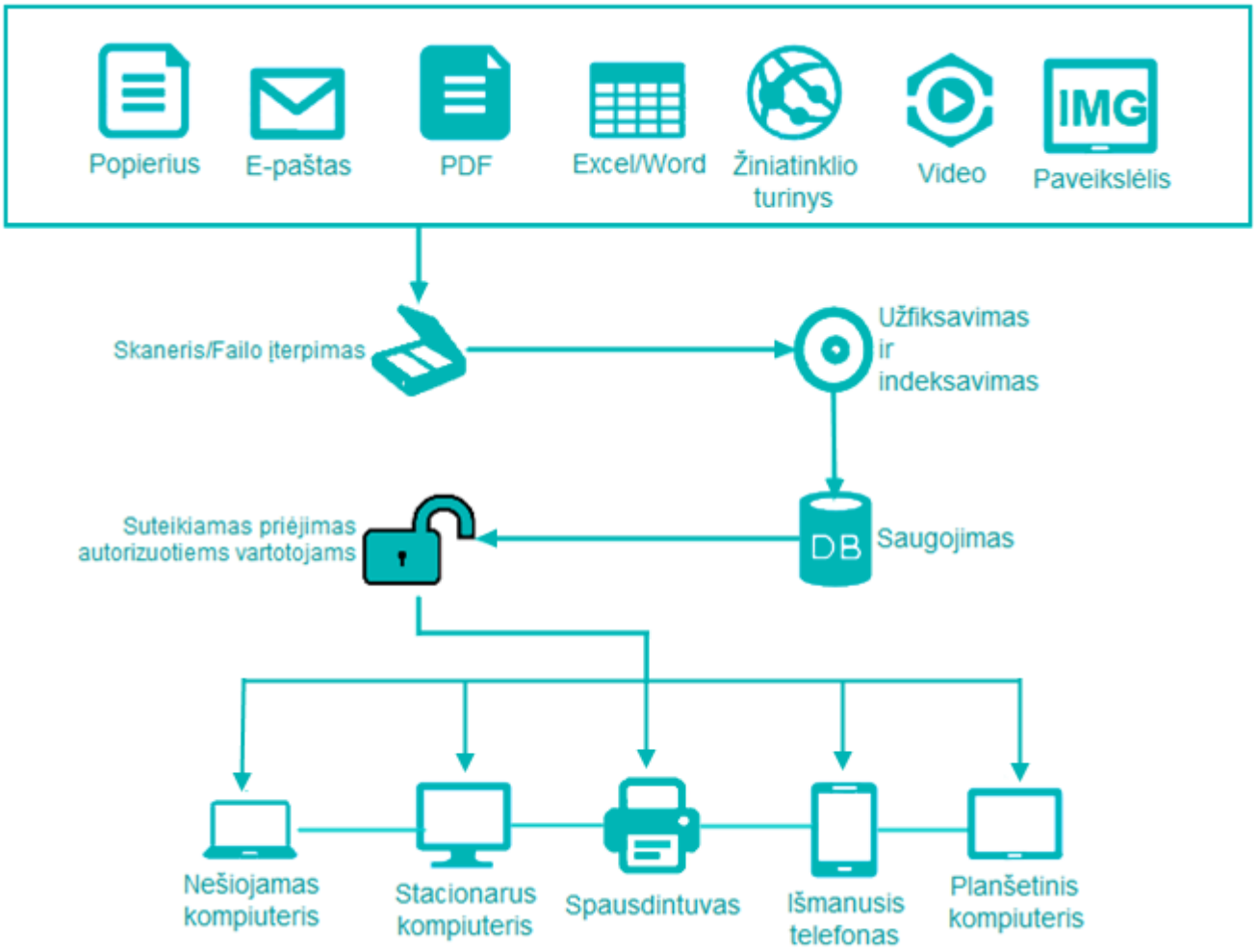
Dažniausiai yra paprasta susieti dokumentų valdymo komponentus su informacijos gyvenimo ciklo fazėmis. Pavyzdžiui, dokumentų skanavimas ir užfiksavimas vyksta kūrimo metu, skanuojant popieriaus lapą taip jį skaitmenizuojant. Šie skaitmenizuoti failai tuo pačiu metu yra publikuojami, indeksuojami ir pažymimi metaduomenimis. Kai dokumentai jau yra sukurti jie yra pastumiami į DVS saugojimo komponentą, kuri integruojama kartu su paieškos ir paėmimo funkcionalumais tam, kad vartotojai galėtų rasti ko ieško ir panaudoti failą su kitais komponentais [7].

DVS integravimo ir saugumo moduliai užima kelias ciklo fazes. Jie pasirūpina visos sistemos veikimo pamatu, kuris įgalina sistemos veikimą ir jos saugumą.

Gilinantis į komponentų savybes galima išskirti, kad užfiksavimas vyksta nebūtinai skanuojant popieriaus lapą. Tai gali būti ir tiesiog naujo dokumento sukūrimas, jo importavimas ar dar kitas būdas priimtinas vartotojui ir įgyvendintas sistemos gamintojo. Indeksavimo komponentas taip pat gali veikti keliais skirtingais būdais. Priklausomai nuo gamintojo norų gali būti įgyvendinami keletas skirtingų indeksavimo būdų. Pavyzdžiui, galima rinktis pilno teksto indeksavimą, kuris visą leidžia ieškoti dokumento tikrinant kiekvieną žodį, tačiau tai užima labai daug vietos diske ir paima daug procesoriaus resursų. Arba galima naudoti pastaruoju metu *Google* proteguojamą būdą indeksuoti naudojant raktažodžius, tačiau tam gali prireikti ištiso raktažodžių žodyno.

Platinimui dokumentų valdymo sistemoje galima pasirinkti įvairių būdų, tačiau tokie kaip spausdinimas, e-paštas ir faksas yra dažniausiai įgyvendinami dėl savo populiarumo ir prieinamumo (dabar faksas jau kiek rečiau). Kadangi dabar viskas susieta tinklais (intranetas, ekstranetas ar

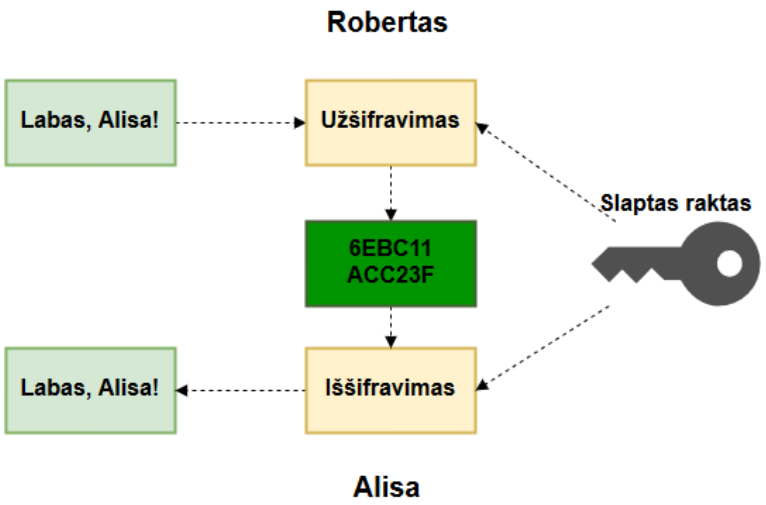
internetas) kliūčių tam neiškyla nors priklausomai nuo tinklo tipo reikia pagalvoti kokių platinimo galimybių gali prireikti. Vieną iš galimų DVS naudojimosi būdų galite išvysti 1.2 paveikslėlyje.



2 pav. DVS veikimo procesai

1.4. Simetrinė kriptografija

Simetrinėje kriptografijoje tas pats raktas naudojamas žinutės užšifravimui ir iššifravimui. Pasinaudojant raktu *k* užkoduojama žinutė *m* ir gaunamas užšifruotas tekstas *c*. Norint iššifruoti užšifruotą tekstą reikia pasinaudoti tuo pačiu raktu ir tada gaunama pirminė žinutė *m*.



3 pav. Simetrinio šifravimo mechanizmas

Simetrinis šifravimas tipiškai veikia dviem būdais: šifruojant blokais arba srautais. Šifruojant blokais visi duomenys yra suskirstomi į tam tikro dydžio duomenų blokus. Duomenys, kurie priklauso nuo bloko ilgio ir naudojamo rakto yra pateikiami šifravimui. Srautinių šifrų atveju, duomenys yra padalinami į atskirus bitus, tada jie sumaišomi ir pateikiami šifravimui. Tam pačiam saugumo lygiui užtikrinti simetriniai raktai tipiškai yra trumpesni už asimetrinius raktus. Simetrinis šifravimas yra gerokai greitesnis už asimetrinį šifravimą [8]. Dėl to jis yra labai tinkamas šifruoti failus, ar kitus duomenis, kurie yra didelio dydžio (>1 MB).

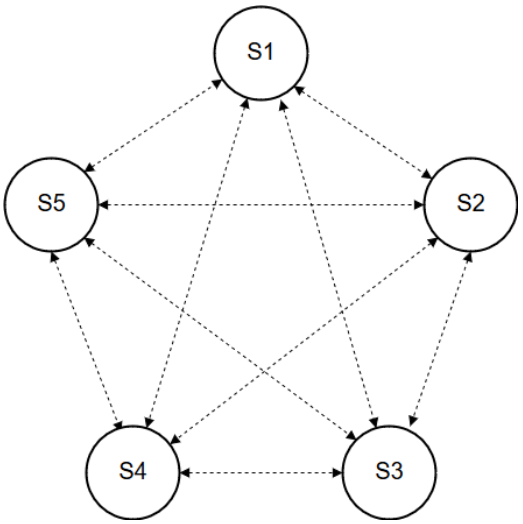
Norint, kad kitas asmuo perskaitytų užšifruotą žinutę simetriniai raktas turi būti persiunčiamas kitam duomenų mainų dalyviui. Jeigu tas dalyvis yra patikimas ir raktas yra pakankamo ilgio, kad nebūtų nulaužiamas per užpuolikams priimtiną laiko tarpą, o persiunčiant jį pavyko išlaikyti paslaptyje, tuomet galima būti įsitikinus, kad siunčiamą žinutę galės perskaityti tik tas adresatas, kuriam ir siunčiama žinutė. Raktas privalo būti laikomas paslaptyje. Raktui patekus į svetimas rankas, jis tampa nebetinkamas, nes tada žinutes gali perskaityti ir neautorizuotas asmuo. Iškylanti problema yra pats rakto persiuntimas. Tam, kad būtų garantuojama, jog persiuntimo metu raktas būtų neperskaitomas, jis gali būti užšifruojamas su kitu simetriniu raktu, kurį jau turi gavėjas. Jeigu naudojamasi antru simetriniu raktu, o gavėjas jo neturi, tada jis turi būti persiųstas adresatui, kad jis galėtų išsišifruoti pirmąjį simetrinį raktą. Tai gali nuvesti iki begalinio raktų šifravimo. Su šia problema susidoroti dažnai naudojama asimetrinė kriptografija, kai siunčiamas simetrinis raktas užšifruojamas gavėjo viešuoju raktu, o jis gautą užšifruotą raktą iššifruoja savo turimu privačiuoju raktu.

Sistemose naudojančiose simetrinius raktus duomenų šifravimui, tarpusavyje gali būti sudėtinga valdyti visą raktų infrastruktūrą. Taip yra dėl to, kad norint užtikrinti kiekvienam sistemos vartotojui galimybę užšifruoti žinutes ir siųsti jas kitam vartotojui reikia suteikti kiekvienam ryšiui tarp dviejų vartotojų po raktą. 4 pav. pateikiama schema, parodo kaip sistemoje turint 5 vartotojus, tai galima įsivaizduoti kaip grafą kuris turi penkias viršūnes. Šiuo atveju simetriniai raktai tai būtų

briaunos arba linijos tarp jų. Šiame pavyzdyje reikalingų raktų kiekis yra 10 raktų. Norint sužinoti raktų kiekį sistemoje su tam tikru vartotojų skaičiumi prieinama prie formulės:

$$r = \frac{n * (n-1)}{2} \quad (1)$$

Ji leidžia apskaičiuoti reikalingų raktų kiekį sistemai, kur raidė *r* reiškia raktų kiekį, o *n* reiškia vartotojų kiekį sistemoje. Jeigu dar būtų bandoma suteikti ir daugiau raktų tarp vartotojų, pvz., grupėms po tris vartotojus, tada skirtingų raktų kiekis dar labiau padidėtų.



4 pav. Raktų valdymas tarp vartotojų

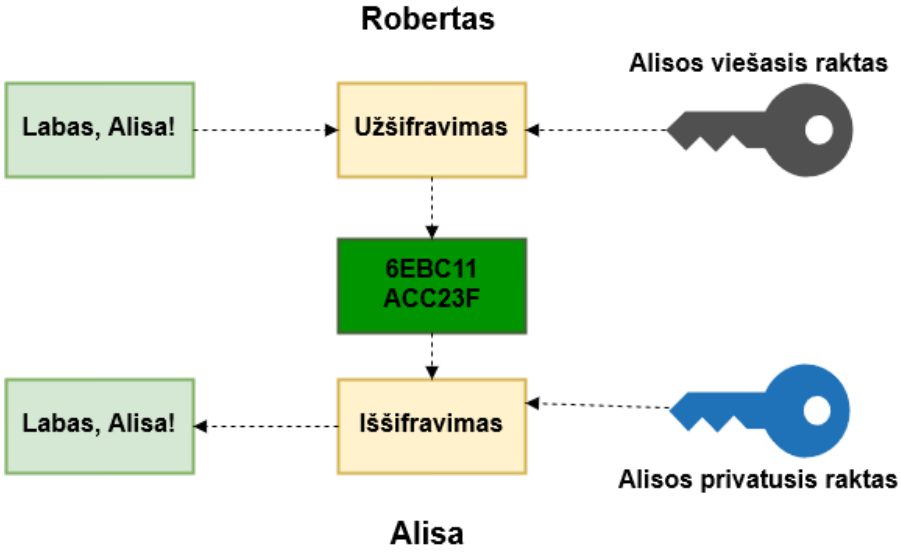
Naudojant kriptografinius raktus dokumentų pasirašymui reikia užtikrinti pasirašomo dokumento vientisumą, autentiškumą ir neišsiginamumą. Autentiškumą ir vientisumą galima pasiekti pasinaudojant žinučių autentifikavimo kodais (*MAC*). Tačiau, net pasinaudojus *MAC* autentiškumas gali būti pasiekiamas tik dalinai ir tik tuo atveju, jeigu raktas yra gerai saugomas, nes kitu atveju, bet kas turintis raktą gali siųsti užšifruotas žinutes adresatui, ir tam nebūtina žinoti koks turėjo būti neužšifruotas pranešimas pačioje pradžioje. Vis dėlto, net ir laikant raktą saugiai to gali neužtekti patvirtinti autentiškumui. Naudojant bendrą raktą šifravimui ir iššifravimui ir jį persiunčiant prieš tai užšifravus gavėjo viešuoju raktu (kaip daroma naudojantis viešojo rakto infrastruktūra), vis tiek atsiveria akivaizdi saugumo spraga, nes bet koks asmuo turinti prieigą prie gavėjo viešojo rakto gali užkoduoti, savo raktą, žinutę ir persiųsti gavėjui. Kadangi gavėjas neturi jokios galimybės pasitikrinti siunčiančio asmens tapatybės, jis negali būti įsitikinęs rakto bei žinutes autentiškumu. Dar viena problema yra simetrinių raktų nesugebėjimas garantuoti parašo neišsiginamumą. Taip yra todėl, kad tiek siuntėjas, tiek gavėjas turi vienodą raktą ir nėra jokio būdo įrodyti, kad siuntėjas pasirašė tą žinutę, o ne gavėjas.

Dėl šiame skyrelyje aprašytų priežasčių nutariama, kad simetrinis raktas pasirašymui netiks.

1.5. Asimetrinė kriptografija

Asimetrinė – dar kitaip vadinama viešųjų raktų kriptografija, tai būdas šifruoti pranešimus vienu raktu, o iššifruoti kitu. Asimetrinių raktų porą sudaro viešasis (*VR*) ir privatusis raktai (*PR*). Dėl jų gebėjimo šifruoti su vienu, o iššifruoti žinutę su kitu išryškėja du pagrindiniai asimetrinės kriptografijos panaudojimo būdai: užšifruoti žinutę taip, kad ją galėtų iššifruoti tik tas asmuo, kuriam žinutė yra skirta ir užšifruoti dokumentą taip, kad bet kas galėtų perskaityti žinutę ir pamatyti, kas užšifravo žinutę.

Norint užšifruoti žinutę taip, kad tik adresatas galėtų ją perskaityti reikia atlikti 5 pav. parodytus veiksmus. Pirma paimamas viešai prieinamas adresato (Alisos) viešasis raktas ir juo užšifruojama žinutė. Gavus užkoduotą žinutę ji persiunčiama adresatui, o šis savo ruožtu pasinaudodamas privačiuoju raktu jį iššifruoja ir perskaito žinutę.



5 pav. Asimetrinio šifravimo mechanizmas

Norint pasirašyti žinutę reikia atlikti priešingus veiksmus nei nurodyta 5 pav. Alisa užšifruoja (pasirašo) žinutę su privačiuoju raktu, o tada visi žmonės turintys prieigą prie Alisos viešojo rakto gali iššifruoti žinutę ir jeigu iššifravimas pavyks tada bus įsitikinta, kad duomenis pasirašė būtent Alisa.

Asimetrinė kriptografija yra pagrįsta vienkrypčių funkcijų veikimu, kai užšifravus žinutę būtų neįmanoma ją iššifruoti ir perskaityti per prieinamą laiko tarpą. Turima privačiojo ir viešojo raktų pora yra matematiškai susijusi, tačiau šis ryšys yra toks, kad iš viešojo rakto sužinoti duomenis apie privatųjį raktą yra labai sudėtinga. Tarp galimų būdų įveikti šį matematinį ryšį ir sužinoti privačią informaciją būtų bandymai vykdyti skaičių faktorizaciją ieškant dviejų pirminių skaičių, kurie yra rakto sandaugos daugikliai, vykdyti diskrečiuosius algoritmus arba nagrinėti eliptinių kreivių ryšius. Tačiau šie išvardyti būdai yra neefektyvūs. Lyginant su simetrine kriptografija raktai tipiškai yra ilgesni. Populiarios *RSA* kriptografinės sistemos saugiu laikomas raktų ilgis šiuo metu yra 2048 bitai. 2009 metų pabaigoje buvo sėkmingai faktorizuotas 768 bitų ilgio *RSA* raktas [9]. Dar visai neseniai 1024

bitų ilgio *RSA* raktai buvo laikomi saugiais, tačiau tobulėjantys kompiuteriai ir matematiniai metodai leidžia faktorizuoti vis didesnius raktų ilgius. Dėl šios priežasties jau 2011 metais NIST aprašė poreikį nebeišduoti 1024 bitų ilgio *RSA* raktų ir jau nuo 2013 metų galutinai pereiti prie dvigubai didesnių raktų ilgių [10]. Palyginus simetrinius ir asimetrinius šifravimus yra ištirta, kad asimetriniai šifravimai yra gerokai lėtesni daugiau nei 10–100 kartų. Tačiau eliptinės kreivės nuo simetrinių šifravimų stipriai neatsilieka, o lėtesnius, pavyzdžiui, *DES, 2DES* simetrinius šifrus lenkia [11].

Viešųjų raktų kriptografija leidžia išspręsti apsikeitimo raktais problemą su kuria susiduriama simetrinėje kriptografijoje, kadangi viešasis raktas gali būti prieinamas visiems. Taip nebereikia rūpintis saugių kanalų palaikymu. Bandant valdyti visą raktų infrastruktūrą, kai asmenų sistemoje gali būti įvairus kiekis, irgi yra paprasčiau. Taip yra todėl, kad čia kiekvienas vartotojas turi turėti tik du raktus (viešąjį ir privatųjį). Tai reiškia, kad sistemoje esant *n* kiekiui asmenų visas raktų *r* kiekis prilygs

$$r = n * 2 \quad (2)$$

o tai yra mažesnis kiekis nei sistemose naudojančiose simetrinius raktus. Nereikia pamiršti ir kitų asimetrinės kriptografijos stiprybių, kurie šiame darbe yra esminiai sėkmingam sistemos veikimui: tai yra galimybė užtikrinti pasirašyto turinio vientisumą, autentiškumą ir parašo neišsiginamumą.

Pasirašant dokumentus vientisumas yra užtikrinamas pasinaudojant maišos funkcijų ir viešųjų raktų kriptografijos kombinacija. Pirmiausia žinutė yra užšifruojama naudojantis maišos funkcijomis ir taip įgauna du svarbius atributus:

1) gauta maišos reikšmė yra mažesnė už pačią žinutę;
2) bent menkiausias pakeitimas dokumento turinyje iškart pakeis ir pačią maišos reikšmę.

Gautoji maišos reikšmė yra užšifruojama (pasirašoma) su asmens privačiuoju raktu. Parašas yra prikabinamas prie žinutės ir tada gavėjas pasinaudodamas siuntėjo viešuoju raktu iššifruoja parašą ir gauna maišos reikšmę. Tuomet jis pasinaudodamas gauta žinute ir siuntėjo naudota maišos funkcija, pats sugeneruoja savo dokumento santrauką ir palygina ją su ta, kuri buvo gauta iššifravus siuntėjo atsiųstą parašą. Jeigu sutampa, tada žinutė nuo pasirašymo momento pakeista nebuvo.

Autentiškumas ir neišsiginamumas viešųjų raktų kriptografijoje iš dalies užtikrinamas tuo, kad dokumentas turi būti pasirašytas privačiuoju raktu ir tik vienas asmuo gali turėti tokį raktą. Pasirašytą tekstą, kaip jau minėta, galima iššifruoti tik pasinaudojant to asmens viešuoju raktu. Taigi jeigu iššifravimas pavyksta su pateiktu viešuoju raktu tuomet galima manyti, kad privatusis raktas yra tikrai to žmogaus. Tačiau tai negarantuoja visiško autentiškumo, nes bet koks piktavalis gali susigeneruoti savo raktą, pasirašyti norimą tekstą ir taip apsimesti kitu žmogumi susikurdamas panašaus turinio skaitmeninį sertifikatą viešųjų raktų infrastruktūroje. Šie sertifikatai saugo visą informaciją (vardas, pavardė, organizacija ir kiti duomenys) apie asmenį, kuris turi tam tikrą privatųjį raktą. Taip pat kartu yra saugomas ir viešasis raktas. Sertifikatą galima susikurti ir pačiam asmeniškai

arba jį išduoda specialūs sertifikavimo centrai. Ši sistema yra pagrįsta pasitikėjimu. Jeigu asmuo pats sau susikuria sertifikatą, tai neatrodys patikimai ir žmogus, tikrinantis parašą gali pamanyti, kad parašas yra netikras, nes sertifikatas nėra išduotas patikimo sertifikavimo centro. Todėl tikrinant parašą būtinai yra patikrinama ir įvertinama, ar sertifikatas kuris slepiasi už parašo, yra išduotas patikimo sertifikavimo centro, taip pat sutikrina viešąjį raktą su sertifikate esančiu raktu ir patikrina, ar pavyksta iššifruoti pasirašytą tekstą. Tokiu atveju tikrintojas gali įvertinti, ar galima pasitikėti šiuo parašu, ir tuo pačiu susieja šį parašą su pasirašiusiuoju taip, kad šis negalėtų išsiginti savo parašo.

Asimetrinė turi daug įvairių kriptosistemų, tačiau darbe bus apžvelgiamos, dėl jų portabilumo tik pačios populiariausios.

1.5.1. RSA kriptosistema

Tai yra viena iš pirmųjų sukurtų viešųjų raktų kriptografijos kriptosistemų. Jos saugumas pagrįstas ta idėja, kad yra labai sudėtinga per prieinamą laiką faktorizuoti didelius sveikuosius skaičius. *RSA* raktai dažniausiai būna tokių ilgių kurie yra skaičiaus 2 kartotiniai, kaip tradiciškai yra skaičiuojama kompiuterijoje, pvz., 512, 1024, 2048 ir t. t. [12] Šiuo metu jau rekomenduojami yra 2048 bitų ilgio (arba ilgesni) raktai, nes trumpesni variantai jau laikomi nebesaugiais arba greit tapsiančiais nesaugiais.

Norint sukurti *RSA* raktą yra atliekami šie veiksmai:

a. sukuriami du dideli pirminiai skaičiai p ir q. Tuomet yra suskaičiuojamas

$$n = p * q \tag{3}$$

kur n yra parašo modulis;

b. sugalvojamas reliatyviai pirminis e reikšmei

$$\varphi = (p - 1)(q - 1) \tag{4}$$

Reliatyviai pirminis e yra toks skaičius, kad skaičių e ir φ didžiausias bendras daliklis būtų lygus 1. e gali būti ir nedidelis skaičius, dažnai renkami variantai yra *3, 65537*, tačiau jis turi būti mažesnis už φ reikšmę;

c. Viešasis raktas yra *(e, n)*, o privatusis – *(d, n)*, kur d yra būtų galima panaudoti šioje lygtyje

$$e * d = 1 \, mod \, (\varphi) \tag{5}$$

t. y. e ir d sandauga, padalyta iš φ liekana lygi 1 [13].

Bandant užšifruoti žinutę m yra apskaičiuojama lygtis

$$c = m^e \, mod \, n \tag{6}$$

kur c yra gautasis užšifruotas tekstas, e yra naudojama eksponentė, n yra rakto modulis.

Bandant iššifruoti tekstą c apskaičiuojama lygtis

$$m = c^d mod\ n \quad (7)$$

kur *m* yra gautoji iššifruota žinutė, *c* užšifruotas tekstas, *d* – privatusis raktas, o *n* yra rakto modulis.

Žinutės pasirašymas veikia panašiai kaip ir užšifravimas. Pasirašant apskaičiuojama

$$s = m^d mod\ n \quad (8)$$

ir gaunamas pasirašytas tekstas *s*. Tada yra vykdomas patikrinimas su gautu žinutės tekstu ir iššifruotu tekstu

$$m = s^e mod\ n \quad (9)$$

1.5.2. ElGamal kriptosistema

ElGamal panašiai kaip *RSA* saugumo garantui naudojasi tuo, kad reikia per daug didelių kompiuterių resursų, kad būtų galima jas išspręsti. Skirtumas yra tas, kad šiuo atveju apskaičiuoti yra sunku diskretųjį logaritmą. Ši problema apibūdinama taip, kad jeigu yra turimas *p* pirminis skaičius ir sveikieji skaičiai *g* ir y tuomet būtų sunku surasti skaičių *x* lygtyje

$$g^x = y\ (mod\ p) \quad (10)$$

ElGamal raktų sukūrimas pagrįstas šiais veiksmais:

a. sukuriamas didelis pirminis skaičius *p*;
b. Tada pasinaudojant atsitiktinių skaičių generatoriumi sukuriami sveikasis skaičius *g* ($1 \leq g \leq p-1$) [14];
c. sukuriamas atsitiktinis skaičius *x*, kur ($1 \leq x \leq p-2$). Tai yra privatusis raktas;
d. tuomet yra viešojo rakto dalis *y* lygtyje:

$$y = g^x\ mod\ p \quad (11)$$

Ją apskaičiavus gaunamas viešasis raktas yra *(p, g, y)*.

Norint užšifruoti tekstą *m* jis yra padalinamas į blokus (m_1, *m2*, *m...*) taip, kad kiekvieno bloko reikšmė *a*, kad ($0 \leq a \leq p - 1$). Tuomet yra sugalvojamas atsitiktinis skaičius *k*, ($0 \leq k \leq p - 1$), kuris būtų reliatyviai pirminis su skaičiumi $p - 1$. Tada kiekvienas žinutės *m* blokas yra užšifruojamas lygtimis:

$$a = g^k\ mod\ p \quad (11)$$

$$b = y^k m\ mod\ p \quad (12)$$

Skaičių *a* ir *b* pora yra žinutės bloko *m* šifrograma, Tai reiškia, kad užšifruoto teksto dydis yra dvigubai didesnis nei *m*.

Norint iššifruoti *a* ir *b* reikšmes reikia pasinaudoti privačiuoju raktu *x*. Žinutė *m* yra gaunama pasinaudojant šia lygtimi:

$$m = b/a^x m\ mod\ p \quad (13)$$

Turint omeny, kad

$$a^x \equiv g^{kx} (mod\ p) \tag{14}$$

tada išeina, kad

$$\frac{b}{a^x} \equiv \frac{y^k m}{a^x} \equiv \frac{g^{kx} m}{g^{xk}} \equiv m\ (mod\ p) \tag{15}$$

ir taip gaunama žinutė *m* pasinaudojant anksčiau gautais *a* ir *b* [15].

1.5.3. EC kriptosistema

Eliptinių kreivių kriptosistema veikia sukurdama baigtinį reikšmių lauką iš eliptinių kreivių lygties

$$y^2 = x^3 + ax + b \tag{16}$$

sprendimų aibės su pridėtiniu atpažinimo elementu. Egzistuoja daug skirtingų būdų sukurti eliptinių kreivių parašus [12].

Eliptinių kreivių parašai yra vertinami dėl to, kad manoma, jog jie yra sunkiau apskaičiuojami nei kitose kriptosistemose taikomi metodai. Dėl šios priežasties eliptinių kreivių raktai gali būti trumpesni nei kitų kriptosistemų raktai, tam kad būtų užtikrinamas tas pats saugumo lygis. Šiame projekte tai yra labai naudinga savybė.

1.6. Kriptografinių algoritmų saugumo palyginimai

Kriptografiniai algoritmai gali pasiūlyti skirtingo lygio saugumą. Žinoma rakto stiprumas priklauso nuo pasirinkto algoritmo ir kuriamo rakto ilgio. Raktų palyginimai yra atlikti remiantis prielaida, kad raktai buvo generuoti ir valdomi laikantis tam tikrų rekomendacijų (pvz., generavimui naudojant atsitiktinių skaičių generatorius su pakankamai geru atsitiktinumo užtikrinimu – entropija).

Algoritmai laikomi panašaus stiprumo, jeigu algoritmo nulaužimas arba rakto išsiaiškinimas užtrunka maždaug tokį patį laiko tarpą naudojant tuos pačius resursus. Kriptografinio algoritmo stiprumas apibūdinamas reikalingų atlikti kombinacijų kiekiu, kuris leistų pabandyti visas galimus būdus atspėti *X* ilgio simetrinį raktą. Taip būtų tuo atveju, jeigu efektyviausias atakos būdas yra paprasta brutalios jėgos ataka. Kitais atvejais jeigu rakto ilgis yra *Y*, tačiau yra efektyvesnių atakos būdų nei visų galimų raktų bandymas, tai jų saugumas apibūdinamas bandant palyginti su *X* ilgio simetriniu raktu. Tai reikštų, kad *Y* ilgio raktas gali suteikti maksimalų *X* bitų ilgio saugumo stiprumą. Saugumo bitams paaiškinti galima pasitelkiant keletą neužšifruoto ir užšifruoto teksto blokų. Algoritmas, kuris užtikrina *X* bitų saugumą, vidutiniškai turėtų užtrukti apie 2^{x-1} *T* laiko vienetų kol būtų sėkmingai įvykdyta ataka. Šiuo atveju *T* laiko vienetas yra laikas, kurio reikia atlikti vieną teksto užšifravimą ir rezultato palyginimą su atitinkama užšifruota reikšme.

Skirtingų algoritmų apytiksliai palyginimai matomi 1 lentelėje.

1. Pirmas stulpelis parodo saugumo kiekį bitais. Tai atitiktų skaičių *X*. Jis keičiasi kiekvienoje eilutėje priklausomai nuo naudojamo rakto ilgio.
2. Antrasis stulpelis parodo naudojamus simetrinių raktų algoritmus, kurie suteikia saugumo lygį parodytą pirmame stulpelyje.
3. Trečiame stulpelyje minimalus parametrų susijusių su standartais naudojančiais baigtinių laukų kriptografiją (*FFC* angl. *finite-field cryptography*) dydis. Tokių algoritmų pavyzdžiai yra *DSA* skaitmeniniams parašams ir *Diffie-Helman* raktų apsikeitimas, kur *L* yra viešasis raktas, o *N* – privataus rakto dydis.
4. Ketvirtas stulpelis nurodo reikšmę *k* (taip pat ir pirminio skaičiaus *n* ilgį) algoritmams, kurie pagrįsti sveikųjų skaičių faktorizaciją (*IFC* angl. *integer-factorization cryptography*). Pagrindinis tokio tipo algoritmas yra *RSA*. Reikšmė *k* yra dažniausiai laikoma rakto ilgio atitikmeniu.
5. Penktas stulpelis nurodo reikšmių rėžį *f* (dydį *n*, kur *n* yra bazinio taško *G* laipsnis) algoritmams pagrįstiems eliptinių kreivių kriptografija (*ECC*). Reikšmė *f* yra laikoma rakto dydžiu.

1 lentelė Kriptografinių algoritmų saugumo stiprumo palyginimai

Saugumo stiprumas bitais	**Simetrinių raktų algoritmai**	**FFC (pvz. DSA, D-H)**	**IFC (pvz. RSA)**	**ECC (pvz. ECDSA)**
≤ 80	2TDEA	$\|L\| = 1024$ $\|N\| = 160$	$\|k\| = 1024$	$f = [160,..,223]$
112	3TDEA	$\|L\| = 2048$ $\|N\| = 224$	$\|k\| = 2048$	$f = [224,..,255]$
128	AES-128	$\|L\| = 3072$ $\|N\| = 256$	$\|k\| = 3072$	$f = [256,..,383]$
192	AES-192	$\|L\| = 7680$ $\|N\| = 384$	$\|k\| = 7680$	$f = [384,..,511]$
256	AES-256	$\|L\| = 15360$ $\|N\| = 512$	$\|k\| = 15360$	$f = 512+$

Pateikti 1 lentelės duomenys rodo, kad dabartinis rekomenduojamas *RSA* rakto ilgis (2048 bitai) suteikia maždaug 112 saugumo bitų. Tai yra laikoma šiuo metu pakankamai saugiu raktų ilgiu, tačiau tobulėjant technologijoms tai gali būti įveiktas ir tokio ilgio *RSA* raktai. Ieškant dar saugesnių sprendimų, kad būtų užtikrintas rakto saugumas ilgesniam laikui galima didinti *RSA* rakto ilgį iki 3072 bitų. Tai atitiktų *AES-128* blokinio šifro stiprumą t. y. jo įveikimas užtruktų 2^{127} *T* laiko vienetų, kai *T*

yra laikas trunkantis užšifruoti tekstą ir palyginti su turimu užšifruotu tekstu. Toks saugumo lygis tikrai yra pakankamas, tačiau jeigu yra poreikis turėti trumpesnį asimetrinį raktą, tuomet tam puikiai tinka *ECDSA* raktai. Minėtam *AES-128* raktui pagal saugumą prilygsta *ECDSA* 256 bitų ilgio raktas. Labai pageidaujant galima susikurti ir dar saugesnių raktų, tačiau tam vargu ar yra prasmė, nes ir taip niekas nesugebėtų nulaužti tokio ilgio raktų.

Oranžine spalva ir punktyrine linija pažymėta pirmoji eilutė rodanti į mažesnio arba lygų 80 saugumo bitų yra jau neberekomenduojami raktų ilgiai, dėl nepakankamo saugumo užtikrinimo. Tačiau juos dar galima naudoti siekiant patikrinti senus parašus arba iššifruojant informaciją, tuo pačiu turint omeny galimas rizikas atliekant šiuos veiksmus. Apačioje geltona spalva ir stora juoda linija išskirti keturi langeliai parodo į *NIST* standartus neįtrauktus raktų ilgius dėl jų efektyvumo trūkumo.

Verta pažymėti, kad lentelės duomenys gali ir būtinai keisis bėgant laikui. Mūro dėsnis vis dar galioja, o tai reiškia sparčiai tobulėjančias procesorių galimybes atlikti skaičiavimo veiksmus. Taip pat įtaką daro ir geresnių faktorizavimo algoritmų sukūrimas, eliptinių kreivių diskrečių logaritmų atakavimo tobulinimas. Jeigu bus gerai ištobulinti kvantiniai kompiuteriai, kuriais galės prieinamu būdų vykdyti atakas tuomet asimetriniai raktai gali visiškai nebeužtikrinti jokio saugumo [16].

1.7. Kriptografinės maišos funkcijos

Norint pasirašyti dokumentus dažniausiai, resursų ir vietos taupymo sumetimais tekstas yra sutrumpinamas. Tam puikiai pasitarnauja kriptografinės maišos funkcijos.

Maišos funkcijos *H*, kad jos būtų saugios turi būti vienkryptės t. y. tokios, kurias būtų galima atkoduoti, o atkoduoti būtų praktiškai neįmanoma. Tokia funkcija sugeba paversti, bet kokio ilgio tekstą į fiksuoto ilgio šifrogramą. Tam, kad maišos funkcija būtų laikoma kriptografiškai saugi turi būti įgyvendinama ir dar viena sąlyga: turi būti išvengiama galimos maišos funkcijos reikšmių kolizijos. Maišos reikšmių kolizija pasitaiko, kai užšifravus du skirtingus tekstus gaunama ta pati maišos funkcijos reikšmė, išreiškiama $H(a) = H(b)$, kai $a \neq b$ [13]. Kiekviena maišos funkcija gali susidurti su kolizijomis. Pavyzdžiui, šiuo metu populiari *SHA-256* maišos funkcija užkoduoja tekstą į 256 bitų ilgio reikšmę. Kadangi galimų būdų užkoduoti tekstą yra 2^{256}, tai reiškia, kad galų gale vis tiek gali pasitaikyti galimybė gauti tas pačias reikšmes net užkodavus skirtingą tekstą. Dėl to nėra visiško atsparumo nuo tokios problemos, o tiesiog labai maži šansai jai atsitikti.

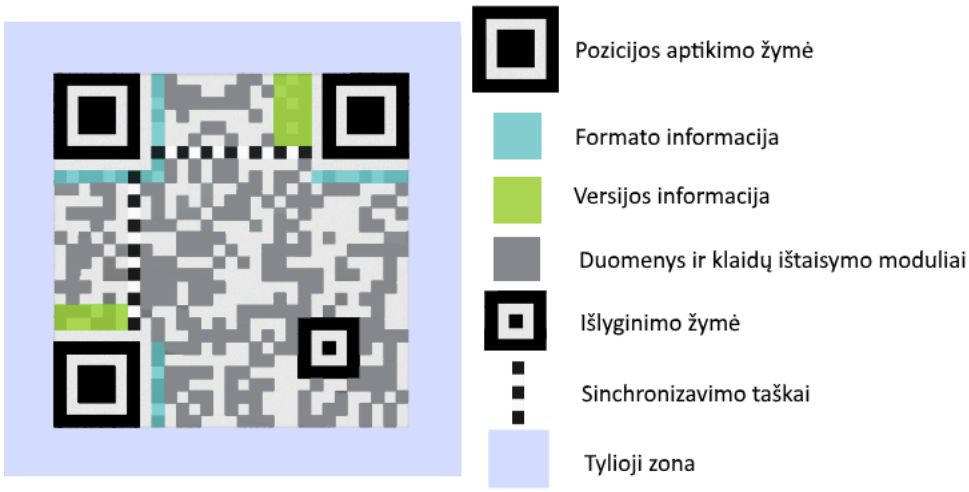
Vadinamasis „gimtadienio paradoksas“ parodo atsparumo kolizijai viršutines ribas. Jeigu maišos funkcija sugeneruoja *N* ilgio kodą tai reiškia, kad atakuotojas sugeneravęs vidutiniškai) $2^{N/2}$ skirtingų maišos operacijų su skirtingomis reikšmėmis turi ganėtinai gerus šansus aptikti du sutampančius maišos funkcijos rezultatus. Tačiau tai yra tik paprasta, neefektyvi „brutalios jėgos“ ataka. Jeigu būtų koks lengvesnis būdas atrasti kolizijai, tada maišos funkcija būtų laikoma kaip turinti didelių saugumo skylių [17].

Visos kokybiškos maišos funkcijos yra pradžioje sukuriamos tokios, kurioms nulaužti nėra kitų būdų kaip tik „brutalios jėgos“ ataka. Tačiau tobulėjant technologijoms ir matematinėms formulėms, kai kurioms maišoms funkcijoms yra atrandama efektyvesnių būdų nulaužimui nei „brutalios jėgos“ ataka. Pastaruosius 15 metų labai populiarios maišos funkcijos *MD5, SHA-1*, jau yra sukompromituotos [18] [19] ir jų palengva atsisakoma ten, kur saugumas yra labai vertinamas. Kitos dabar populiarios ir rekomenduojamos maišos funkcijos yra *SHA-2(SHA-224, SHA-256, SHA-512 ir t.t.)* bei *SHA-3(SHA3-224, SHA3-256, SHA3-512 ir t. t.).*

1.8. QR kodai

QR kodai yra dvimačiai brūkšniniai kodai sukurti 1994 metais, ieškant būdų sutalpinti daugiau informacijos per išnaudojama vietą nei iki tol buvę 1D brūkšniniai kodai arba kiti panašūs duomenų užkodavimo įrankiai.

QR kode duomenys yra užkoduoti tam tikros spalvos kvadratiniais taškais vienspalviame fone. 6 pav. matosi tokio kodo struktūra. Kampuose esantys keturkampiai (pozicijos aptikimo žymės) rodo informaciją apie poziciją. Kamerai nuskaičius *QR* kodą šie pozicijos aptikimo žymės yra reikalingos norint jį teisingai atvaizduoti, kad būtų galima atlikti duomenų skaitymą. Kvadratas apačioje dešinėje yra išlyginimo žymelė. Kuo didesnis *QR* kodas tuo daugiau tokių keturkampių. Jie yra skirti tam, kad būtų išlyginti visi galimi *QR* kodo iškraipymai, kai *QR* kodas yra skaitomas kaip nors pakreiptas. Šitokių žymelių gali būti daugiau tuose *QR* koduose, kuriuose užkoduota daugiau informacijos. Ryškiai juodi nuosekliai einantys juodi taškeliai kairėje vertikaliai ir viršuje horizontaliai yra sinchronizavimo taškeliai, leidžiantys aptikti kiekvieną taškelį *QR* kode. Šviesiai žalia spalva yra nudažyta informacija apie *QR* kodo versiją. Šviesiai mėlyna spalva pažymėta informacija apie užkoduotų duomenų formatą, pavyzdžiui, tekstas, vizitinė kortelė ir t. t. Tai taip pat yra ir informacija apie klaidų ištaisymo lygį. Šviesiai melsvai nudažytas plotas aplinkui *QR* kodą yra vadinamoji „tylioji zona“. Ji skirta tam, kad izoliuotų *QR* kodą nuo galimų pašalinių objektų, kurie galėtų pakenkti *QR* kodo nuskaitymui. Visa kita likusi informacija t. y. pilki taškeliai *QR* kodo moduliai yra užkoduota informacija kartu su klaidų ištaisymo taškais.



6 pav. QR kodo struktūra

QR kodai palyginus su paprastais brūkšniniais kodais gali tame pačiame plote talpinti informacijos daugiau nei 10 kartų. Maksimalūs duomenų kiekio apribojimai *QR* kodui yra:

a. 7089 simboliai skaitiniams duomenims;
b. 4296 simboliai mišriems skaitiniams/raidiniams duomenims;
c. 2593 simboliai baitų (8 bitų) duomenims;
d. 1817 kanji (japoniški/kiniški/korėjietiški hieroglifai) simboliai [20].

Nuo talpinamų duomenų kiekio priklauso ir versijos skaičius. Versijos eina nuo 1 iki 40. 1 *QR* kodo versija talpina po 21 tašką horizontaliai ir vertikaliai. Kiekvieną kartą didėjant versijai *QR* kodo matmenys taškais padidėja po 4, pvz., 2 versijos *QR* kodas jau bus 25x25 taškų pločio. Maksimaliai galima pasiekti 40 versijos *QR* kodą su 177x177 modulių kiekiu [21].

Talpa gali dar labiau sumažėti jeigu bus naudojama didesnio lygio klaidų korekcija. Ji skirta tam, kad jeigu *QR* kodas šiek tiek susigadina būtų galima nuskaityti duomenis nepaisant poveikio *QR* kodui. Yra keturi skirtingi klaidų ištaisymo lygiai, kurie leidžia atstatyti *QR* kodą pagal skirtingą poveikio *QR* kodui dydį:

a. žemas (L – *Low*) – 7%;
b. vidutinis (M – *Medium*) – 15%;
c. kokybiškas (Q – *Quality*) – 25%;
d. aukštas (H – *High*) – 30%.

QR kodai su dideliu klaidų ištaisymo lygiu, gali būti naudingi aplinkose, kur jie gali susitepti ar kitaip susigadinti, tačiau švariose aplinkose kaip planuojama šiame darbe dirbant su el. dokumentais klaidų ištaisymo lygis gali būti ir L arba M, o tai yra naudinga, nes tai leis sudėti daugiau informacijos į mažesnį plotą.

1.9. XMLDSIG, XAdES analizė

Pasirašytiems elektroniniams dokumentams būna taikomi, elektroninių parašų standartai. Vienas iš tokių yra *XMLDSIG*. Tai yra elektroninių parašų valdymo, apdorojimo taisyklės. Specifikacijoje nurodoma, kad būtų galima pasirašyti visą *XML* kodą. Pasirašomi duomenys gali būti tame pačiame dokumente, o kitu atveju bet kur išorėje. Pirmuoju atveju tikrinant parašą tenka atmesti parašo turinį, o antruoju parašas yra pasiekiamas kitokiais būdais. Pasirašant duomenys turi būti transformuojami t. y. supaprastinami, kad jie visada būtų lengvai prieinami ir suprantami [13].

XMLDSIG elementai pateikiami 7 pav. Jie visi susiję su pačiu el. parašu. *SignedInfo* elementas nurodo informaciją, kuri yra pasirašoma. Tinkama parašo validacija vykdoma dviem etapais: parašo tikrinimu naudojantis *SignedInfo* informacija ir tikrinant *Reference* maišos reikšmes. *CanonicalizationMethod* elementas nurodo kanonizacijos algoritmą naudojamą kaninizuoti *SignedInfo* elementą prieš pasirašant. *SignatureMethod* elementas nurodo algoritmo metodą kuriuo naudojamasi pasirašyti norimą reikšmę į parašo reikšmę(*SignatureValue*). Į šį elementą taip pat įeina informacija apie maišos algoritmą. *Reference* elementuose yra laikoma informacija apie maišos metodą su adresu į duomenų objektą. *KeyInfo* elemente saugo informaciją apie patį raktą, kuriuo galima patvirtinti parašą. Čia yra pridedamas viešasis raktas, tos raktų poros, kuria buvo pasirašytas dokumentas. Taip pat dar gali būti pridedamas elementas saugojantis informaciją apie X509 sertifikatą [22] .

```
<Signature ID?>
   <SignedInfo>
     <CanonicalizationMethod/>
     <SignatureMethod/>
     (<Reference URI? >
       (<Transforms>)?
       <DigestMethod>
       <DigestValue>
     </Reference>)+
   </SignedInfo>
   <SignatureValue>
  (<KeyInfo>)?
  (<Object ID?>)*
 </Signature>
```

7 pav. Bazinė XMLDSIG struktūra

Apačioje papildomai gali būti pridedamas *Object* elementas talpinantis informaciją apie dokumento turinį apvelkančio parašo atveju. Standarte daug apie tai neužsimenama, vartotojui veiksmų laisvė kaip jis nori įgyvendinti savo el. parašo dokumentą.

Šitokia veiksmų laisvė leido atsirasti *XAdES* standartui [13]. Jis nurodo galimas sintakses *XML* struktūroms įgyvendinant elektroninius parašus. Jis turi keletą atmainų, kurių kriptografinis saugumo lygis palaipsniui didėja. *XAdES* savo struktūra atrodo kaip praplėstas *XMLDSIG* standartas.

Jame išplėstas *Object* elemento panaudojimas, kuriame papildomai saugoma pasirašomoji ir nepasirašomoji informacija. *XAdES* išsaugoja visas *XMLDSIG* savybes ir dar prideda savų papildomų galimybių. Svarbiausia, kad tiek, *XAdES*, tiek *XMLDSIG* užtikrina galimybę saugoti ir tikrinti su įvairiais el. parašų tipais.

XAdES turi keletą sintaksės formų. Kiekvieną iš jų turi pradinius prieš tai sekančios *XAdES* sintaksės elementus, tačiau papildomai prideda dar vieną ar daugiau papildomų elementų užtikrinti didesnėms dokumento galimybėms dirbant su elektroniniais parašais. Bazinė forma palyginimui su *XMLDSIG* pasipildo šiais elementais (8 pav.).

```
QualifyingProperties
    SignedProperties
        SignedSignatureProperties
            SigningTime
            SigningCertificate
            SignaturePolicyIdentifier
            SignatureProductionPlace?
            SignerRole?
        SignedDataObjectProperties
            DataObjectFormat*
            CommitmentTypeIndication*
            AllDataObjectsTimeStamp*
            IndividualDataObjectsTimeStamp*
    UnsignedProperties
        UnsignedSignatureProperties
            CounterSignature*
```

8 pav. XAdES papildomi elementai

Visa tai kas pateikta 8 pav. yra talpinama *Object* elemente. Visi šie elementai turi savo paskirtį ir kai, kuriuos apžvelgsiu šiame skyrelyje.

SigningTime elementas yra pasirašymo laikas. Tai yra uždedamas laikas to momento kada buvo pasirašytas dokumentas. Elemento formatas *XAdES XML* schemose yra nustatomas kaip *dateTime* ir turi atitikti *ISO 8601* formatą.

SigningCertificate elementas naudojamas tam, kad būtų apsisaugojama nuo sertifikatų pakeitimo ir žinutė neatrodytų lyg buvo pasirašyta kažkieno kito. Jame saugoma nuorodą į patį sertifikatą, sertifikato santraukos reikšmė ir metodas naudotas santraukos reikšmės sugeneravimui.

SignerRole elementas talpina informaciją apie pasirašiusio asmens rolę kompanijoje. Šis elementas gali būti arba gali ir nebūti priskirtas prie parašo.

DataObjectFormat elementas suteikia informaciją apie pasirašytų duomenų formatą. Šis elementas yra privalomas tais atvejais kai yra reikalinga parodyti pasirašyto duomenų objektą vartotojams patikrinimui. Dokumente gali būti daugiau nei vienas toks elementas, jeigu yra pasirašoma daugiau nei vienas objektas. Šio elemento perduodama informacija gali būti apie tekstinę informaciją

susijusią su dokumento aprašymu, pasirašyto duomenų objekto(ų) *MIME* tipą, pasirašyto duomenų objekto(ų) duomenų užkodavimo (*encoding*) formatą.

CounterSignature elementas atsakingas už parašų saugojimą, galiojančių tik kartu su pagrindiniu parašu. Turi būti palaikomos kelios formos tokių parašų patenkančių į šias dvi kategorijas: nepriklausomi parašai, įtrauktieji parašai. Jame būna saugomi hierarchiškai susiję parašai, o ryšiai tarp parašų yra įgyvendinami pasinaudojant nuorodų elementais [22].

1.10. Esamų dokumentų pasirašymo ir patikrinimo sprendimų analizė

Internete galima rasti nemažai produktų siūlančių dokumentų valdymo, jų pasirašinėjimo paslaugas. Šiame skyrelyje bus apžvelgiamos 4 sistemos, kurios teikia elektroninių dokumentų pasirašymo paslaugas:

1) DocuSign;
2) HelloSign;
3) Ascertia DSS;
4) Adobe Sign.

DocuSign

Ši dokumentų pasirašymo programa siūlo galimybę susikurti savo parašą, tokį koks jis yra rašomas ranka. Gali būti pasirašomi įvairūs dokumentai. Jie turi būti įkelta į jų sistemą, kuri parodo dokumento atvaizdą ir leidžia vartotojui įvairiose vietose pridėti pageidaujamas žymes ir duomenis kaip vardas, adresas, telefono numeris ir t. t. Pasirinkus pridėti parašą leidžiama pasirinkti tarp galimybės pridėti vieną iš daugybės sukurtų parašų bendram naudojimui, parašyti norimą tekstą kursyvu, susikurti ir panaudoti savo parašą įkeliant parašo paveikslėlį arba sukuriant jį pelės vedžiojimu ekrane. Galima palikti vietos ir kitam asmeniui, kuriam bus siunčiamas dokumentas tam, kad pasirašytų. Padarytas patogus dalinimasis failu su galimybe išsiųsti pageidaujamam asmeniui [23]. Dėl kriptografinio parašo trūkumo kyla rimtų abejonių dėl šių parašų juridinės galios ir ypač – patikimumo.

HelloSign

Ši dokumentų pasirašymo platforma panašiai kaip ir *DocuSign* įgyvendina elektroninių dokumentų pasirašymą naudojant ranka sukurtus parašus. Naudotis šia sistema galima keliais būdais: prisijungiant prie jų tinklapio; naudojantis jų palaikoma aplikacijų programavimo sąsaja, leidžiančia per žiniatinklio paslaugas jungtis prie jų serverio ir valdyti dokumentus ir paskyras; naudojantis jų pateiktomis integracijomis su *Google, Microsoft* ir kitų kompanijų siūlomomis paslaugomis, pvz., *Gmail, Google Docs, OneDrive, DropBox, Slack* ir t. t. Ši sistema siūlo vartotojui galimybę pasirinkti

norimą dokumento šabloną [24]. Kaip ir *DocuSign* pasirūpinta tam tikru sistemos saugumu, pvz., naudojant SSL sertifikatus serveryje, tačiau pagrindinės problemos išlieka tos pačios – nėra panaudojami kriptografiniai elektroniniai parašai, o tai verčia abejoti ranka rašytų parašų patikimumu.

Ascertia DSS

Ascertia DSS pasirašymo serveris yra, anot jų pačių, pasaulyje lyderiaujantis dokumentų pasirašymo sprendimas. ADSS siūlo galimybę kurti kriptografinius parašus. Jis teikia sertifikavimo centro paslaugas atskirame serveryje ir leidžia susikurti, ir naudoti *RSA* (1024, 2048, 4096 bitų ilgio) bei *ECDSA* (192, 224, 256, 384, 521 bitų ilgio) asimetrinius raktus. Leidžiama pasirašyti įvairių tipų elektroninius dokumentus, pvz., *PDF, XML*, žiniatinklio formas ir kitus e-dokumentų tipus. Santraukų skaičiavimui gali būti naudojami *SHA-1, SHA-256, SHA-384, SHA-512* vienkryptės maišos funkcijos. Suteikiamos ir kitos viešųjų raktų infrastruktūros paslaugos kaip sertifikatų atšaukimo, sertifikatų būsenos tikrinimas ir kt. Pasirašant dokumentą pridedamos laiko žymos iš patikimų laiko žymėjimo centrų. Norint užtikrinti ilgalaikį dokumento parašo gyvavimą, parašai ir su jais susijusi informacija laikoma remiantis *CAdES, XAdES, PAdES* ir kt. parašų formatais. Serveris apsaugotas *SSL* sertifikatu, tačiau reikalui esant galima pasiekti naudojant tik *HTTP* [25]. Yra siūloma galimybė pasirašinėti dokumentus mobiliuoju telefonu ir juos patikrinti, pateikiant elektroninį pasirašyto dokumento failą. Galimybių patikrinti atspausdintą lapą nesuteikiama.

Adobe Sign

Adobe Sign sprendimas yra pagrįstas veikimu debesyse (*cloud-based*). Kaip ir kitos sistemos ji siūlo elektroninių dokumentų pasirašymo, tikrinimo ir valdymo paslaugas. Jie teikia galimybę susikurti savo parašą ranka arba įkeliant paveikslėlį su parašu. Pageidaujant gali būti suteikiamas kriptografinis raktas išduotas patikimo sertifikavimo centro, atitinkančio keliamus reikalavimus, pvz., *AATL* (*Adobe Approved Trust List*), *EUTL* (*European Union Trusted Lists*). Pasirašant dokumentai saugomi debesyse, kad būtų galima prie jų prieiti bet kada ir su bet kokiu įrenginiu. Jeigu pasirašoma su skaitmeniniu kriptografiniu parašu tuomet dokumentai yra saugomi naudojantis AdES reikalavimais. Dokumentų pasirašymas atitinka ES *eIDAS* (*Electronic Identification and Trust Services*) taisykles. Kadangi Adobe Sign veikia debesyse, tai yra suteikiama galimybė dirbti ir mobiliais įrenginiais parsisiuntus jų programėlę. Telefonu galima patikrinti e-dokumentus ar jie nebuvo pakeisti ar kitaip pažeisti, taip pat – patikrinti parašus [26]. Tikrinimui nuo popieriaus lapo išlieka probleminė sritis, kai vienintelis būdas yra skanuoti dokumentą ir kelti į serverį, kur bus atliekamas tikrinimas dėl sutapimų, tačiau teksto skanavimas gali neužtikrinti patikimumo dėl įvairių sąlygų.

2 lentelė Esamų sprendimų palyginimas

E-dokumentų pasirašymo sprendimai / Palyginimo kriterijus	DocuSign	HelloSign	Ascertia DSS	Adobe Sign
Kriptografinių raktų palaikymas	-	-	+	+
AdES reikalavimų laikymasis	-	-	+	+
Mobili integracija	+	+	+	+
Dokumentų šablonų įgyvendinimas	-	+	-	-
Parašo tikrinimas atspausdintame dokumente	-	-	-	+

Aprašius e-dokumentų pasirašymo sprendimus yra atliekamas jų palyginimas, pateikiamas 2 lentelėje. Žinoma, kad šios sistemos turi įgyvendinusios daug įvairių galimybių darbui su e-dokumentais, tačiau yra koncentruojamasi į šiuos kriterijus, dėl jų svarbos nagrinėjamoje probleminėje srityje:

1) kriptografinių raktų palaikymą – tikrinama, ar sistema leidžia naudoti asimetrinius kriptografinius raktus dokumentų pasirašymui, pvz., *RSA*;
2) *AdES* reikalavimų laikymąsi – tikrinama, ar pasirašius dokumentą, sistema laiko parašo duomenis pagal *AdES*(*CAdES*, *PAdES*, *XAdES*) reikalavimus;
3) mobili integracija – tikrinama, ar sistema leidžia naudotis mobiliaisiais telefonais pasirašant ir tikrinant dokumentus;
4) dokumentų šablonų įgyvendinimas – tikrinama, ar sistema teikia galimybę kurti dokumentus ir kuriant pasirinkti norimą dokumento šabloną;
5) parašo tikrinimas atspausdintame dokumente – tikrinama, ar sistema turi sugebėjimą tikrinti parašus ant atspausdintų dokumentų.

Palyginimas rodo, jog yra sprendimų, kurie gali pasiūlyti saugius kriptografinius parašus ir tų, kurie negali. Natūralu, kad neturint galimybės pasirašyti su kriptografiniu parašu negalima sukurti ir dokumento, kuris vadovautųsi *AdES* reikalavimais. Lyginant šias sistemas tik viena galėjo pasiūlyti galimybę kurtis dokumentus ir naudoti tam tikrą šabloną. Visos kitos sistemos tiesiog pasirašo jau egzistuojantį dokumentą, kurį pateikia vartotojas. Mobilią integraciją gali užtikrinti dažna parašų sistema, tačiau tai būna dažnai pritaikyta tik elektroniniams dokumentams, o galimybės tikrinti atspausdintą dokumentą beveik nerandama. Adobe Sign siūlo tokią galimybę, tačiau tam reikia skanuoti visą dokumentą ir turėti elektroninį variantą palyginimui. Čia atsiveria dvi problemos: reikia būtinai turėti elektroninį variantą palyginimui ir interneto prieigą, bei teksto skanavimo patikimumas kelia abejonių dėl patikimumo, nes esant netinkamam apšvietimui, prastai kameros kokybei ir kitoms

galimoms problemoms gali iškilti sunkumų nuskaityti dokumentą tiksliai. Teksto skaitymo problemą galima kiek sumažinti skaitant su skaneriu, tačiau tam reikia skanerio, o jie nėra tiek prieinami kiek paprasti mobilūs telefonai, todėl dirbant su įvairiais dokumentais (spausdintais ir elektroniniais) gali nukentėti darbo su jais sparta.

Kuriant projektą bus atsižvelgiama į šiuos duomenis, iškilusias problemas. Kuriamoje sistemoje bus įgyvendinami lentelėje aprašyti kriterijai, sprendžiamos šioje analizėje iškilusios problemos dėl dokumento tikrinimo mobiliuoju įrenginiu.

1.11. Analizės išvados

Atlikta analizė parodė, kad simetrinė kriptografija, nors ir greitesnė už asimetrinę kriptografiją, negali būti naudojama norint saugiai pasirašinėti dokumentus, nes ji tiesiog tam nėra pritaikyta. Viešųjų raktų kriptografijos vienas iš pagrindinių panaudojimo būdų yra būtent įvairaus turinio pasirašymas, nes ji sugeba užtikrinti svarbiausias savybes dokumentų pasirašyme: vientisumą, autentiškumą ir neišsiginamumą. Analizė parodė, kad siekiant optimizuoti parašo informacijos perdavimą naudojantis QR kodais ir minimizuoti būtinos informacijos šifravimo kiekį verčiau turėti trumpesnius raktus. Ištirti įvairūs viešųjų raktų kriptografijos parašų metodai parodė eliptinių kreivių pranašumą, nes ji gali sukurti gerokai trumpesnius raktus nei kita šiuo metu populiari *RSA* kriptosistema, tuo pačiu metu išlaikydama panašų saugumo lygį.

Dirbant su pasirašytais elektroniniais dokumentais yra patartina laikytis standartų reglamentuojančių kaip turi būti valdomi elektroniniai parašai ir kaip jie turi būti atvaizduojami kartu su dokumento informacija. Naudojamas *XAdES* standartas apibrėžia visas gaires, nurodo *XML* schemą kaip turi būti valdomi parašai e-dokumente. Laikantis šių reikalavimų galima užtikrinti, kad pasirašytas dokumentas išlaikys parašo juridinę galią, turės atsparumą nuo atakų bandančių sukompromituoti paties dokumento turinį, parašo validumą, sertifikatą ar net pasirašymo laiką.

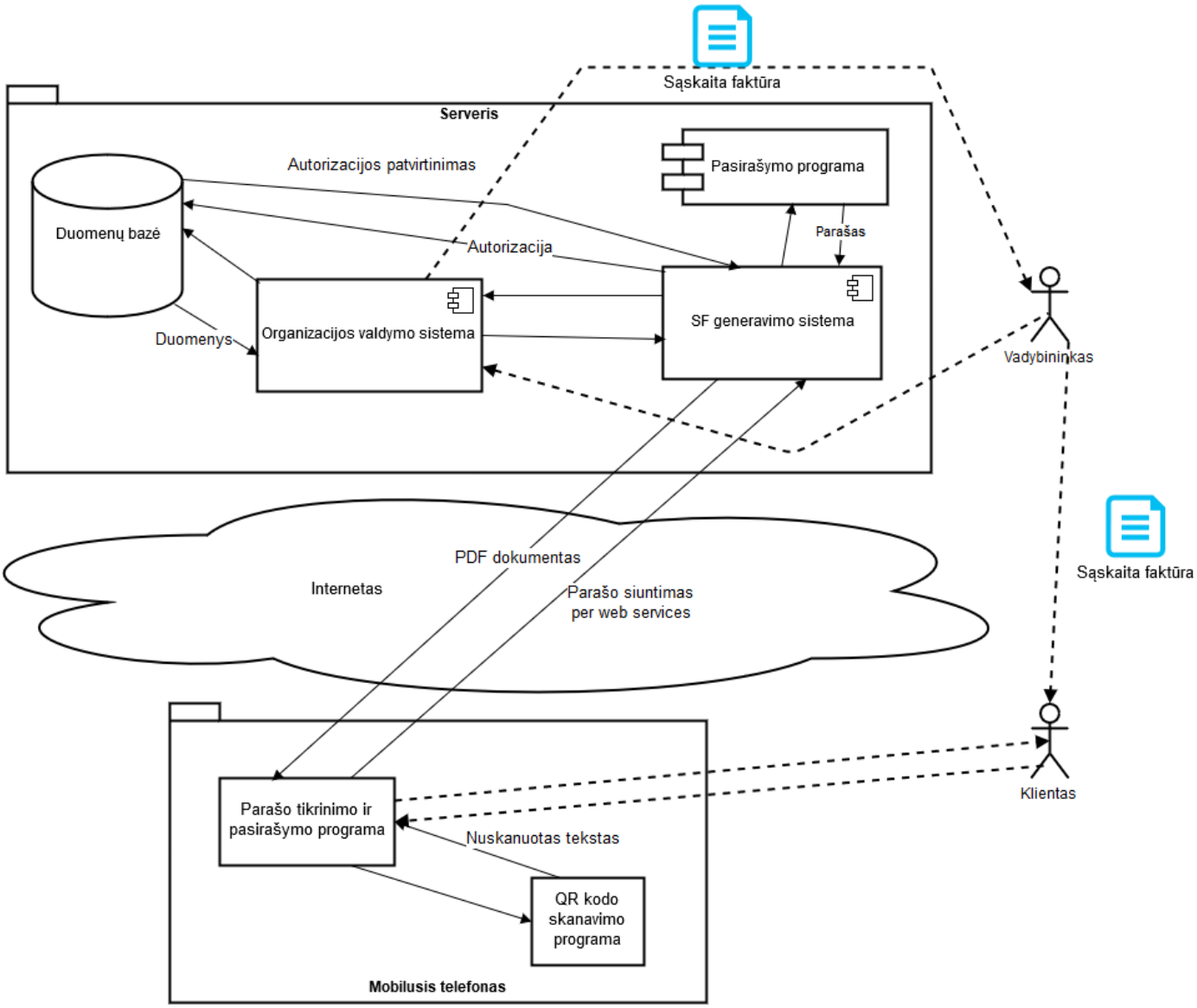
Atlikta lyginamoji analizė padėjo pamatyti kokie šiuo metu yra rinkoje esantys produktai, kuo jie geri, ko jie nėra padarę ir kur jie yra padarę nepakankamai. Išsiaiškinta, kad dauguma pigesnių sprendimų nors ir gali pasiūlyti dokumentų valdymą, pasirašymą ir parašų tikrinimą, prisijungimą per mobilius įrenginius. Kai kurie sprendimai leidžia rinktis dokumento šablonus ir juos valdyti. Tam tikrų sprendimų suteikiami parašai ne visada yra garantuoto saugumo, kadangi tai nėra kriptografiniai parašai. Sistemos, kurios gali pasiūlyti saugius sprendimus pasirašant ir tikrinant parašus, taip pat turi gausybę ir kitų papildomų funkcijų, kurios įeiną į kainą, o jos ne visiems yra reikalingos.

Atlikta analizė suteikė reikalingą informaciją pradėti sistemos projektavimą, tuo pačiu metu pagerinant jo kokybę ir užkertant kelią galimiems saugumo pažeidimams.

2. E-DOKUMENTŲ SISTEMOS PROJEKTAS

Šiame skyriuje pasinaudojant analizėje surinkta informacija aprašoma kuriamos sistemos struktūra. Bendram vaizdui sudaryti yra pateikiama visos sistemos architektūra, pateikiami funkciniai ir nefunkciniai reikalavimai. Detalesniam kiekvieno veiksmo paaiškinimui pateikiamos veiklos diagramos. *QR* kodo struktūra paaiškina kaip yra talpinami duomenys *QR* kode, kad vėliau jie būtų sėkmingai nuskaitomi. Galiausiai paaiškinami duomenys, kurie bus siunčiami per žiniatinklio paslaugas, kad visada reikalingos sistemos dalys gautų reikalingus duomenis numatytiems veiksmams atlikti.

2.1. Sistemos architektūra



9 pav. Sistemos architektūra

Sistemos architektūros paveikslėlyje (9 pav.) matome pagrindines dalis sudarančias visą sistemą. Visas bendravimas vyks tarp dviejų pagrindinių platformų – nutolusio tam tikros

organizacijos serverio ir mobiliojo telefono. Serveryje bus klientų aptarnavimo programa, tai gali būti bet kokia žiniatinklio programa, verslo valdymo sistema arba dokumentų valdymo sistema. Kai vadybininkas nusiųs nurodymą gauti organizacijos suformuotos sąskaitos faktūros, naudojantis duomenų bazės duomenimis bus kreipiamasi į sąskaitų faktūrų generavimo sistemą, jai pateikiami reikalingi duomenys, raktų informacija, o ten bus sukuriama sąskaita faktūra ir grąžinamas *PDF* dokumentas atgal vadybininkui. Tada šis asmuo nusiųs klientui sąskaitą faktūrą, kurią gavo iš serverio, o šis gavęs *SF* ir savo telefone turintis diagramoje nurodytą mobiliąją aplikaciją bei *QR* kodo skanavimo programą, galės nuskaityti dokumentą Pamatęs dokumentą ir su juo susijusią informaciją tada jis galės ją pasirašyti jeigu tai yra būtent tas žmogus iš kurio ir laukiama parašo. Klientui iniciavus šį veiksmą bus siunčiami autentifikacijos duomenys bei pati parašo informacija per žiniatinklio paslaugas į serverį. Serveris naudodamasis duomenų baze autentifikuos vartotoją ir kreipsis į SF generavimo sistemą, kuriai bus pateikiamas parašas, o ji tik sugeneruos atnaujintą sąskaitą faktūrą ir grąžins *PDF* dokumentą atgal klientui.

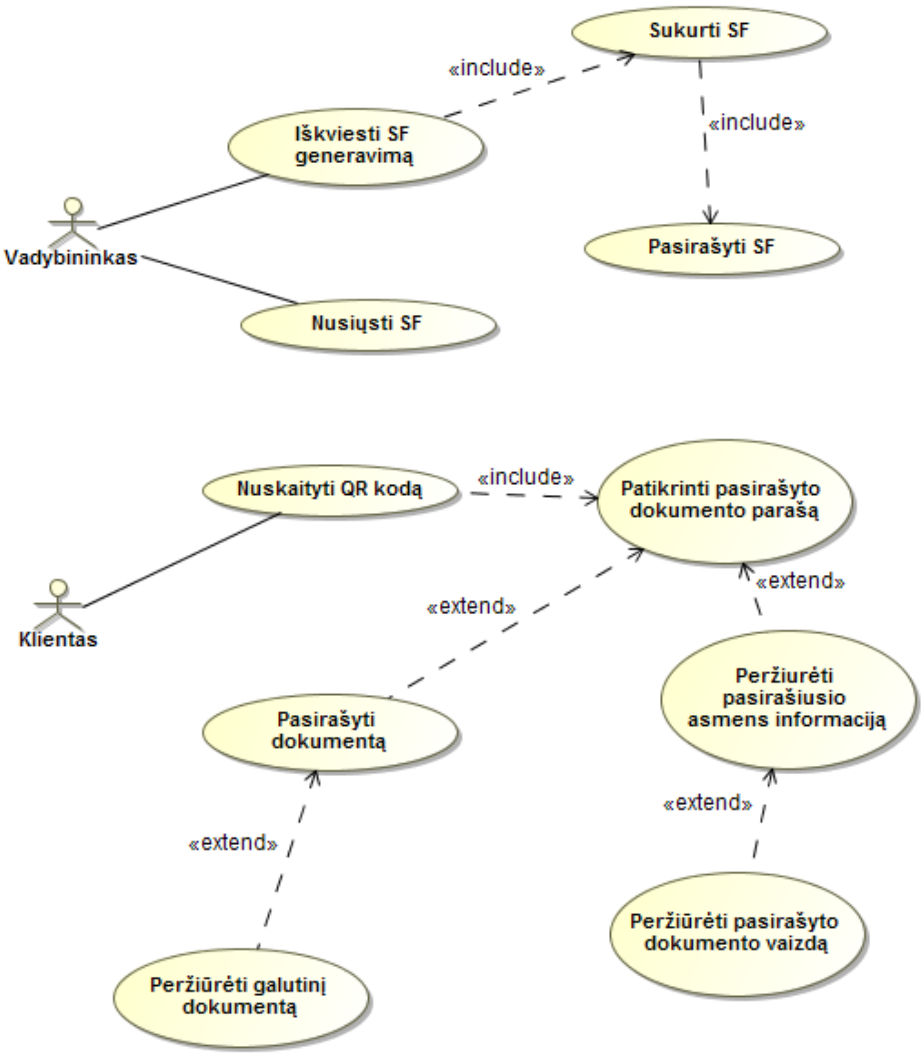
2.2. Reikalavimų specifikavimas

Šiame skyrelyje apibrėžiami funkciniai ir nefunkciniai reikalavimai. Jie padeda nustatyti ir apibūdinti sistemos funkcionalumą, galimybes, nurodo jos veikimą ir naudojamus įrankius įgyvendinimui.

2.2.1. Nefunkciniai reikalavimai

- Parašas turi būti kuriamas naudojantis *EC* (eliptinių kreivių) kriptosistema.
- Kuriamo rakto ilgis turi būti bent 256 bitai.
- Dokumentai serveryje bus laikomi *XML* formatu.
- Pasirašyti dokumentai bus laikomi *XAdES* dokumente.
- Sistemos projektas turi būti baigtas iki 2017 metų birželio mėnesio.
- Kuriama mobilioji programėlė turi būti pritaikyta Android operacinei sistemai.
- Mobiliosios programėlės veikimas turi būti užtikrinamas ne senesnėje nei *Android 7.0 OS* versijoje.

2.2.2. Funkciniai reikalavimai



10 pav. Sistemos panaudojimo atvejai

Paveikslėlyje 10 pav. pateikiami sistemos panaudojimo atvejai, Pirma organizacijoje vadybininkas pareikalauja sistemos sugeneruoti naują sąskaitą faktūrą. Tuo metu yra kuriama sąskaita faktūra, o po sukūrimo ji iškart pasirašoma. Vadybininkas tuomet turi pasirašytą dokumentą, kurį gali nusiųsti pageidaujamam asmeniui.

Klientas su turimu atspausdintu dokumentu gali perskaityti *QR* kodą esantį ant jo. Tuomet yra patikrinamas dokumento parašo tinkamumas. Jeigu parašas teisingas tuomet leidžiama klientui pasirinkti ar jis nori peržiūrėti pasirašiusiojo asmens informaciją. Po to jis dar gali pasirinkti peržiūrėti patį dokumentą, kuris buvo pirma pasirašytas. Taip pat po parašo patikrinimo klientas turi pasirinkimą pats pasirašyti dokumentą ir taip pridėti antrą parašą. Po to savo pasirašymo klientas gali peržiūrėti pasirašytą dokumentą *PDF* formatu.

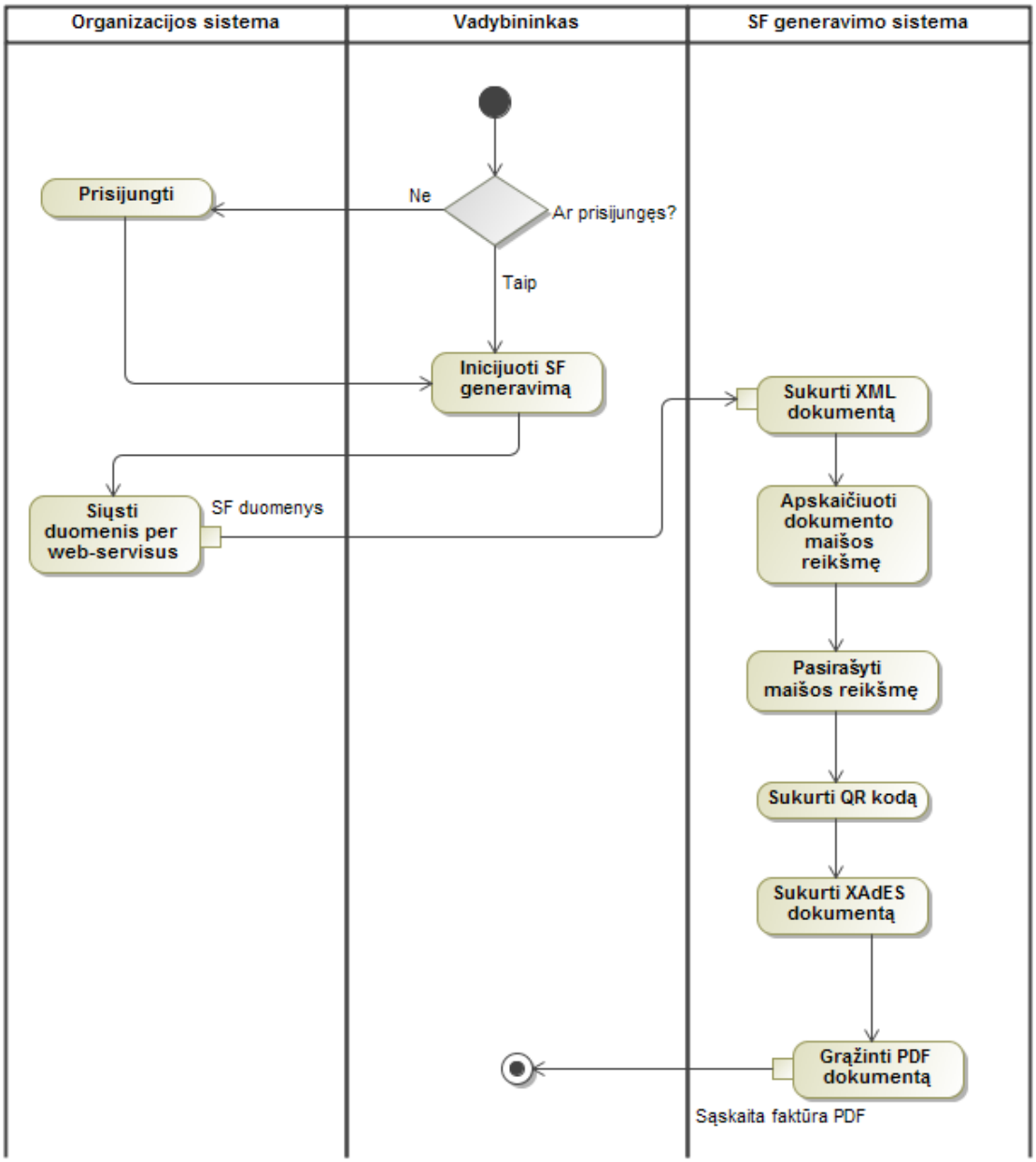
2.3. Veiklos diagramos

Šiame skyriuje yra pateikiamos veiklos diagramos apibrėžiančios kaip veikia sistema, kokius veiksmus ir kokiu eiliškumu juos atlieka.

2.3.1. Sąskaitos faktūros sukūrimas organizacijos serveryje

Organizacijai įvykdžius tam tikrą operaciją (pardavimą, pervedimą) kuriama sąskaita faktūra (*SF*). Ji būna kuriama dar esant organizacijos tinklalapyje. Kaip pateikta 11 pav. veiksmas prasideda nuo vadybininko – asmens, kuris atsakingas už tokių dokumentų išrašinėjimą. Pirmiausia jam reikia prisijungti prie savo sistemos, jeigu iki šiol dar nėra prisijungęs. Tuomet susiradęs ir išsirinkęs duomenis, kuriuos naudos sąskaitos faktūros sukūrimui jis inicijuoja SF generavimą. Sistema pasinaudodama žiniatinklio paslaugomis siųs *JSON* užklausą į *SF* generavimo sistemą. Toji užklausa neš su savimi informaciją, kuri bus naudojama sąskaitos faktūros kūrimui. Taip pat bus nešama autorizacijos žymė, kurios pagalba bus atliekama autorizacija su SF generavimo sistema tam, kad tik autorizuoti vartotojai galėtų kurti dokumentus serveryje.

Duomenis gavusi ir juos patikrinusi *SF* generavimo sistema sukurs *XML* dokumentą, kurį vėliau pasirašys. Po sukūrimo naudojantis *SHA-256* maišos funkcija sukurs dokumento santraukos reikšmę. Tada pasiims organizacijos naudojamą privatų raktą iš raktų saugyklos saugomos *JKS* (*Java KeyStore*) formato failuose. Su šiuo raktu pasirašys jau anksčiau gautą dokumento santraukos reikšmę. Tuomet bus sukurtas *XAdES XML* dokumentas, kuriame bus laikoma visa reikalinga informacija. Į jį pateks jau anksčiau sukurtas *XML* failas užkoduotas *base64* formatu, taip pat parašas užkoduotas Base64 formatu, santraukos reikšmė ir kiti atributai, kurie būna dedami į *XAdES* dokumentus. Po to pasinaudojant numatytuoju sąskaitos faktūros šablonu bus sukurtas pats dokumento vaizdas pagal turimus duomenis ir paverstas *PDF* formatu. *PDF* formato failo turinys bus paverčiamas į *base64* formatą ir kaip tekstas parsiunčiamas atgal vadybininkui per organizacijos sistemą. Po parsiuntimo *PDF* failas bus dekoduojamas iš Base64 ir sukuriamas *PDF* failas. Šis failas bus atiduotas vadybininkui ir taip bus užbaigtas *SF* generavimas.

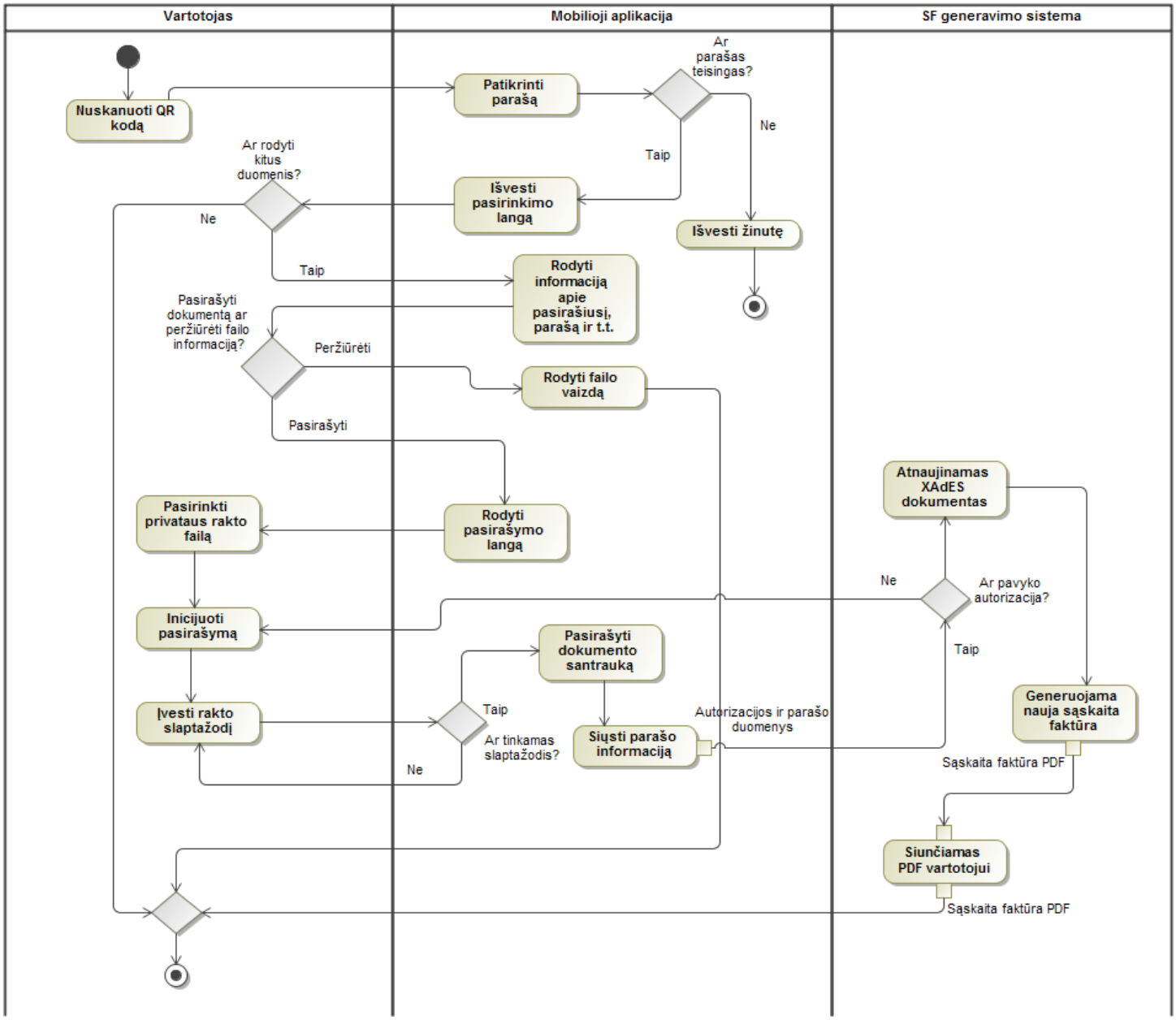


11 pav. Sąskaitos fakturos sukūrimas

Vadybininkas su gautuoju *PDF* dokumentu galės daryti, kas yra jam numatyta pagal darbo pobūdį, šiuo atveju tai būtų dokumento išsiuntimas klientui, kuriam ir buvo išrašyta ši sąskaita.

2.3.2. Sąskaitos faktūros parašo tikrinimas

Iš vadybininko gautas dokumentas(elektroniniame ar spausdintame formate) parodo klientui ar bet kokiam kitam asmeniui reikalingas operacijos detales. Tačiau, vartotojas negali būti įsitikinęs, kad šis dokumentas nėra kaip nors padirbtas ar pakeistas persiuntimo metu. Taip pat net jeigu dokumentas ir yra geras reikia įsitikinti, kad parašas esantis ant dokumento yra tikrai tos šalies kuri ir matosi dokumente. O patikrinus gali prireikti pasirašyti dokumentą ir pačiam klientui. Tokio patikrinimo ir pasirašymo schema yra atvaizduota 12 pav.



12 pav. Sąskaitos faktūros nuskaitymo, parašo tikrinimo veiklos diagrama

Pirmiausia vartotojas pasiėmęs savo mobilųjį telefoną įsijungia aplikaciją skirtą aprašytiems veiksmams atlikti. Paspaudus mygtuką „Skaityti *QR* kodą“ naudojantis telefono kamera rodomas vaizdas matomas per objektyvą. Tuomet vartotojas nukreipia kamerą į *QR* kodą matomą ant dokumento ir jį nuskaito. Programa užfiksavusi *QR* kodą jį nuskaitys. Tuomet programa iškart pradės analizuoti gautą simbolių seką. Bus pradedamas parašo tikrinimas dekoduojant *QR* kode saugomą parašą iš *base64* formato. Tuomet pasinaudojant nuskanuotu viešuoju raktu ir santraukos reikšme bus patikrintas parašas ar jis atitinka pasirašytą santraukos reikšmę. Jeigu parašas neatitinka tuomet vartotojui bus pateikiama žinutė apie nepavykusį parašo patvirtinimą ir tada bus baigtas veiksmas ir grįžtama į pradinį aplikacijos langą. Parašo patvirtinimui pavykus bus pateikiamas vartotojui pasirinkimo langas ar jis nori matyti daugiau informacijos. Pasirinkus „Ne“ bus baigtas veiksmas. Pasirinkus „Taip“ jis bus nukreipiamas į langą kur bus rodoma informacija apie pasirašiusį

asmenį(arba organizaciją), pasirašymo laiką, sertifikavimo centro informaciją, tam kad būtų patvirtinamas sertifikato autoritetas.

Atvertame lange kartu su informacija bus taip pat pateikiamos galimybės vartotojui arba peržiūrėti patį failą, kuris bus gaunamas iš *QR* kode užkoduoto *URL* adreso arba pasirašyti nuskaitytą dokumentą.

Pasirinkus pirmąjį variantą bus atveriamas naujas langas aplikacijoje ir rodomas pagal tą pati šabloną sukurtas dokumento vaizdas, toks koks būna matomas naršyklėje. Po to vartotojas gali baigti programos darbą arba grįžti atgal.

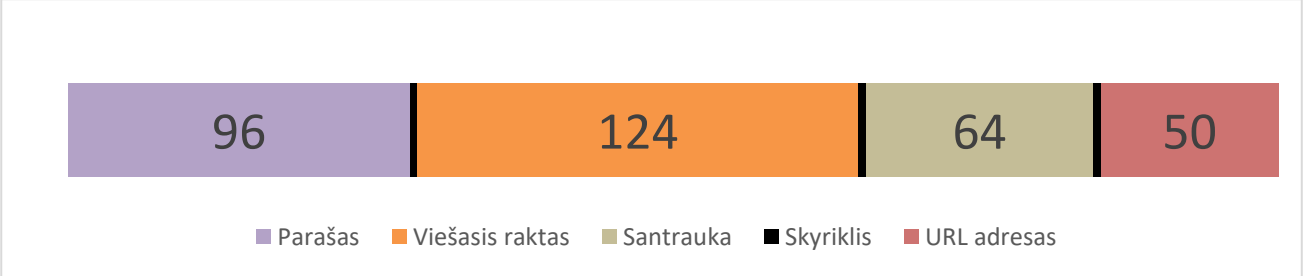
Pasirinkus antrąjį variantą bus atidaromas pasirašymo langas. Jame vartotojas turės pasirinkti savo privataus rakto failą saugomą telefone. Pasirinkus failą bus patikrinama ar failas yra tinkamo formato. Taip pat bus rodomas absoliutus failo kelias, kad būtų įsitikinta, jog pasirinktas tikrai tas failas. Po to vartotojas galės inicijuoti failo pasirašymą. Iš karto bus paprašoma vartotojo įvesti slaptažodį, kad būtų galima naudoti tą raktą panaudoti. Nepavykus įvedimui bus prašoma įvesti slaptažodį iš naujo. Po teisingo slaptažodžio įvedimo pasinaudojant raktu bus pasirašoma nuskaityto dokumento santraukos reikšmė. Iš karto po to sekantis veiksmas bus siųsti per žiniatinklio paslaugas visą parašo informaciją bei autorizacijos duomenis.

Serveris gavęs užklausą patikrins autorizacijos duomenis ir įsitikins ar siuntėjas turi teisę pasirašyti šį dokumentą t. y. ar jis yra skirtas būtent jam. Jeigu autorizacija nepavyks vartotojas bus grąžinamas į langą kur reikėjo inicijuoti dokumento pasirašymą. Jeigu autorizacija pavyks, SF generavimo sistema atnaujins turimą tos sąskaitos faktūros *XAdES* dokumentą ir pridės atsiųstą parašo informaciją.

Pasinaudojant šiuo dokumentu bus sukurtas PDF dokumentas ir grąžintas vartotojui jo pasirinkti būdu – jeigu bus prie duomenų prikabintas elektroninio pašto adresas, tada bus išsiųsta tuo elektroninio pašto adresu, o kitu atveju bus tiesiog parsiunčiamas į telefoną. Tada užsibaigs veiksmas o vartotojas su gautu *PDF* dokumentu galės atlikti norimus veiksmus.

2.4. QR kodo struktūra

Dokumente įdedamas *QR* kode bus patalpinta informacija, kuria pasinaudojant bus galima įvertinti parašo patikimumą, peržiūrėti patį failą, prireikus net pasirašyti. Į *QR* kodą turės būti užšifruota tam tikra informacija, kuri leis tai padaryti. *QR* kodo duomenų dalys matomos 13 pav.



13 pav. QR kodo struktūra

Šiame kode yra pateikiamos keturios skirtingos dalys išskirtos juodais stulpeliais, kurių kiekvienas sudaro po du simbolius:

a) parašas – tai bus su privačiuoju raktu pasirašyta dokumento santrauka, jis bus tikrinamas skanuojant dokumentą;
b) viešasis raktas – tai bus privataus rakto su kuriuo pasirašyta santrauka viešoji dalis. Pasinaudojant juo bus tikrinamas parašas;
c) santrauka – tai yra dokumento duomenų santrauka. Šiuo atveju ji taip pat bus naudojama tikrinant parašą, nes su viešuoju raktu bus gaunama santraukos reikšmė užkoduota paraše ir tada sulyginama su ta kuri užšifruota *QR* kode;
d) *URL* adresas – tai bus nuoroda į serverį, kur saugomas pilnas *XAdES* dokumentas. Mobilioji aplikacija po *QR* kodo skanavimo parsiųs šio dokumento turinį tolesniam patikrinimui ir peržiūrai.

Paveikslėlyje nurodomi pirmieji du ilgiai gali keistis priklausomai nuo naudojamų kriptografinių algoritmų ir jų raktų ilgių. Šiame paveikslėlyje rodomi skaičiai reiškia kiekvienos dalies užimamą duomenų kiekį simboliais. Šiuo atveju informacijos dalių ilgiai skaičiuoti šifruojant duomenis naudojant numatytąjį *ECDSA* 256 bitų ilgio raktą.

2.5. Žiniatinklio paslaugų duomenų perdavimas

Žiniatinklio paslaugomis bus naudojamasi tiek bendraujant tarp skirtingų serverio programų, tiek komunikuojant tarp mobiliojo telefono ir nutolusio serverio.

Generuojant sąskaitą faktūra yra atsiunčiami tam tikri duomenys iš organizacijos valdymo sistemos. Jų struktūra pavaizduota 3 lentelėje.

3 lentelė Siunčiami duomenys iš organizacijos valdymo sistemos į SF generavimo sistemą

Lauko pavadinimas	Duomenų tipas
Pilnas vardas	string
SF numeris	string
Dokumento data	date
Suma	float

Autorizacijos žymė	string

Pateikiami duomenys 3 lentelėje rodo perduodamą informaciją, kuri yra naudojama pradinės SF sugeneravimui. Antrajame stulpelyje pavaizduoti perduodamų duomenų tipai. Tipas *string* reiškia tai, kad bus perduodamas paprastas tekstas, *date* reiškia, kad tai yra datos objektas, *float* reiškia skaičiaus su kableliais formatą.

Paskutinėje eilutėje matoma autorizacijos žymė bus 64 simbolių ilgio tekstas, kuris bus atsiunčiamas kaip patvirtinimas, kad kreipiamasi iš organizacijos valdymo sistemos.

Sugeneravus sąskaitą faktūrą bus grąžinamas *PDF* dokumento turinys. Atsakymo duomenų struktūra pateikiama 4 lentelėje.

4 lentelė Iš SF generavimo sistemos grąžinamo atsakymo struktūra

Lauko pavadinimas	Duomenų tipas
PDF turinys	string
Atsakymo kodas	integer

Sugeneruotas PDF būna dvejetainio formato, todėl tam jis bus pirma paverčiamas į *base64* kodą tam, kad galėtų būti grąžintas kaip *string* formato duomenys. Gavėjas šitokį tekstą turėtų dekoduoti iš *base64* ir paversti dvejetainį turinį į PDF failą. Taip pat yra paduodamas ir atsakymo kodas. Jis perduoda informaciją apie vykdytos operacijos būseną t. y. iš atsakymo eitų suprasti ar pavyko sėkmingai generavimas ar įvyko klaidų, o jei taip tai iš kodo galėtų suprast kokios klaidos. Jis būtų *integer* formato kas reiškia, bet kokį sveikąjį skaičių.

Kreipiantis iš mobilaus telefono jau pasirašius dokumentą siunčiama įvairi su parašu ir vartotoju susijusi informacija, kurios struktūra pavaizduota 5 lentelėje.

5 lentelė Siunčiama informacija iš mob. aplikacijos į SF generavimo sistemą

Lauko pavadinimas	Duomenų tipas
Vartotojo vardas	string
Slaptažodis	string
Parašas	string
Viešasis raktas	string
X509 sertifikatas	string
Santraukos reikšmė	string

Pasirašius ir kreipiantis į serverį bus siunčiami vartotojo vardas ir slaptažodis, tam kad būtų autentifikuotas parašą siunčiantis vartotojas ir negalėtų pasirašyti tas žmogus, kuriam nėra skirtas šis

dokumentas. Kartu siunčiama ir kita informacija kaip pats parašas, viešasis raktas, pasirašiusio žmogus sertifikatas ir kita su parašu susijusi informacija. Ji bus užšifruota *base64* kodu, kad būtų galima persiųsti kaip *string* formato duomenis. Taip pat bus persiunčiama santraukos reikšmė kaip dar viena priemonė patikrinti ar pasirašytas tas pats turinys, koks yra dokumente, kurį siekiama pasirašyti.

SF sugeneravus atnaujintą sąskaitą faktūrą, vėl grąžinamas toks pats atsakymas kaip 4 lentelėje. Tada mobilioji aplikacija jau pati turi pasiversti gaunamus duomenis į *PDF* dokumentą.

3. E-SĄSKAITŲ FAKTŪRŲ PASIRAŠYMO SISTEMOS REALIZACIJA IR TYRIMAS

Šiame skyriuje aptariami kuriamos e-dokumentų pasirašymo ir nuskaitymo sistemos realizacijos įrankiai, funkcijos, bibliotekos. Aprašomas sukurtas prototipas, tyrimų metodai, pateikiami jų rezultatai.

3.1. Naudojami realizacijos įrankiai

Realizuojant prototipą sukurta mobili aplikacija, naudojamos žiniatinklio paslaugos, taip pat naudojama turinio valdymo sistema generuoti e-dokumentams ir jiems parsisiųsti.

Mobilioji aplikacija sukurta ant *Android OS* platformos. Ji parašyta naudojant Java kalbą. Programėlė parašyta naudojantis 24 Android aplikacijų programavimo sąsajos versija pritaikyta *Android 7.0.1 Nougat* (ir naujesnėms) operacinei sistemai.

Žiniatinklio paslaugų failai talpinami nutolusiame serveryje. Juose yra talpinamas PHP kodas kuris priims užklausas iš kliento, kuris jas iškviečia per sukurtąją mobiliąją aplikaciją arba prisijungiant internetu per naršyklę. Sąskaitų faktūrų sistema prisijungia prie MySQL duomenų bazės, kurioje bus saugoma informacija naudojama sąskaitų faktūrų generavimui. Organizacijos raktai dokumentų pasirašymui yra saugomi *JKS* (*Java KeyStore*) failuose, kurie yra pritaikyti raktų talpinimui. Komunikavimui tarp serverio ir mobilaus įrenginio naudojamos žiniatinklio paslaugos.

Organizacijos naudojama sistema yra įgyvendinta pasinaudojant turinio valdymo sistema *PrestaShop*. Joje yra talpinama informacija apie bankinius pavedimus, laikomi sąskaitų faktūrų generavimui reikalingi duomenys, kuriuos toje pačioje sistemoje supildo vartotojai. Ji taip pat suprogramuota *PHP* kalba, o naudojama duomenų bazė yra *MySQL*.

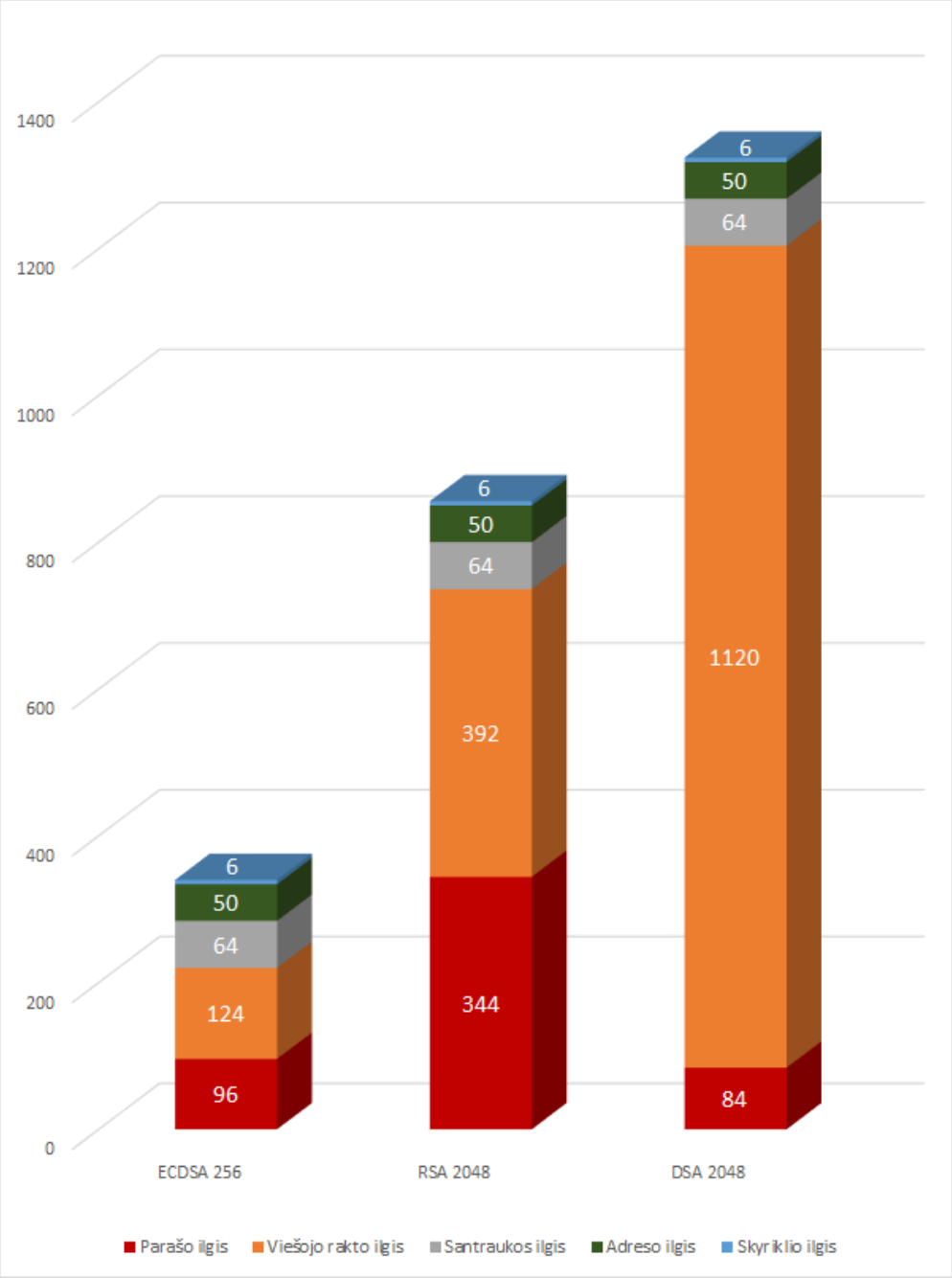
Elektroniniams parašams generuoti yra naudojama EC(eliptinių kreivių) kriptosistema. Pasinaudojant *Java Keytool* programa sukuriami *ECDSA* 256(secp256r1 kreivė) bitų ilgio privatūs raktai. Taip pat sukuriamas *X.509* sertifikatas su nurodyta informacija apie sertifikato savininką. Demonstravimo tikslais naudojantis *OpenSSL* susikurtas sertifikavimo centras, kuris turi savo paties pasirašytą sertifikatą. Tuomet siekiant imituoti galimo vartotojo parašo teisėtumą jis yra pasirašomas su sertifikavimo centro parašu. Pasirašant iš mobiliojo telefono privataus rakto ir sertifikato pora yra saugoma *PKCS#12* formato faile.

3.2. Atliekami tyrimai

3.2.1. Kriptografinio metodo įtaka šifruojamų raktų ilgiui

Norint įgalinti vartotojus nuskaityti failo parašo informaciją greitai ir patogiai, duomenys apie parašą, viešąjį raktą, santrauką, failo adresą yra užšifruojami *QR* kode. Kadangi yra siekiama padaryti, kad *QR* kodas būtų lengvai nuskaitomas įvairiomis sąlygomis(su prastesne kamera, iš didesnio atstumo

ir t. t.) siekiama, kad *QR* kode būtų užkoduota kuo mažiau informacijos, leidžiančios sužinoti reikalingą informaciją apie parašą ir turinį, tačiau užtikrinančios ir galimybę patikrinti parašą iškart neprisijungiant prie interneto. Tam, kad empiriškai būtų galima nustatyti, kuris asimetrinio šifravimo metodas geriausiai tinka šitokiam tikslui, ir yra atliekamas tyrimas. Tyrime yra naudojami trys skirtingi elektroninių parašų algoritmai: *RSA, DSA, ECDSA*. Apytiksliai vienodam saugumui užtikrinti tarp raktų buvo naudojamas *RSA* 2048 bitų raktas, *DSA* 2048 bitų raktas ir *ECDSA* 256 bitų raktas. Taip pat šifruojamas dokumento turinys yra naudojantis *SHA-256* santraukos funkcija. Tekste matomi skaičiai atspindi sugeneruotą simbolių kiekį. Patys simboliai, kurie ir yra talpinami į *QR* kodą, yra užkoduoti *base64* formatu tam, kad būtų galima turėti vientisą vienodo formato informaciją taip išvengiant skirtingų duomenų tipo kombinavimo.



14 pav. Duomenų ilgio priklausomybės nuo pasirinkto rakto grafikas

14 pav. pateikiami *DSA* 2048 bitų rakto sukuriami duomenys yra didžiausio ilgio, trumpiausios *ECDSA* 256 bitų rakto, o *RSA* 2048 bitų ilgio raktas atsidūrė viduryje. Įdomu pastebėti tai, kad pasirašant su *DSA* bus sugeneruota ilgiausias kiekis informacijos, tačiau pats parašas yra trumpiausias, užimantis tik 84 simbolius. Net ~85% viso informacijos kiekio užima viešojo rakto informacija. *ECDSA* ir *RSA* parašo ir viešojo rakto ilgiai skiriasi gerokai mažiau. Visuose trijuose stulpeliuose matomos kelios vienodos reikšmės:

a. po 64 simbolius užima failo turinys užšifruotas *SHA-256* santraukos funkcija;
b. po 50 simbolių palikta vietos *XAdES* dokumento adresui serveryje, kad būtų galima atlikti išplėstinį parašo tikrinimą. Šis ilgis gali skirtis priklausomai nuo įgyvendinimo,

tačiau tyrimo patogumo sumetimais čia buvo pasirinktas pakankamai nuosaikaus ilgio *URL* adreso ilgis;

c. po 6 simbolius yra palikta skyrikliams tarp skirtingų duomenų, kad nuskaičius *QR* kodą būtų galima atskirti, kurie duomenys prasideda, o kurie baigiasi.

Šios trys nekintančios arba mažai kintančios(priklauso nuo adreso) reikšmės sudaro atitinkamai 35 %, 14 % ir 9 % nuo viso informacijos kiekio koduojamo *QR* kode.

RSA sukurtas informacijos kiekis yra apie 2,5 karto didesnis nei *ECDSA* sukurtas informacijos kiekis. Savo ruožtu *DSA* sukurtas informacijos kiekis yra atitinkamai didesnis už kitus du metodus (*ECDSA, RSA*) ~3,9 ir 1,5 karto. Matant šitokį *ECDSA* pranašumą ilgio atžvilgiu tampa akivaizdu, kad jo sugeneruotas informacijos ilgis labiausiai priimtinas dėti į *QR* kodą naudojant projekto struktūroje numatytus raktų parametrus.

Verta atkreipti dėmesį į 15 pav. parodytus tris *QR* kodus. Juose yra talpinama sukurta informacija, kurios ilgiai yra atvaizduojami aptartame grafike. Šiame paveikslėlyje puikiai matosi, kokį efektą *QR* kodo išvaizdai turi koduojamų duomenų kiekis. *ECDSA* kodas nors ir atrodo gali pasirodyti didelio tankio, tačiau šį kriterijų *RSA* ir juo labiau *DSA* parašų informacija užšifruoti *QR* kodai lenkia *ECDSA*. Nėra abejonių, kad bandant sudėti daugiau informacijos į mažesnį plotą bus paveiktos galimybės sėkmingai nuskaityti *QR* kodą. Tokiu atveju sėkmingam *QR* kodo nuskaitymui gali prireikti labai gerų sąlygų, pvz., gero apšvietimo, geros kameros ir kantrybės laikant nutaikytą kamerą į *QR* kodą, nes bandant nuskaityti *QR* kodą su dideliu informacijos tankiu kamera turi ilgai vertinti ir skaičiuoti, kokia iš tikrųjų informacija yra užšifruota jame.

15 pav. Sugeneruoti vienodo dydžio QR kodai, panaudojant skirginsu kriptografinius metodus

3.2.2. QR kodo nuskaitomumas monitoriuje

Dokumente laikant *QR* kodą parašo atpažinimui priklausomai nuo poreikio gali būti spausdinami įvairaus dydžio *QR* kodai. Jeigu yra taupoma vieta dokumente tai būtų naudinga jei paveikslėliai ir kita papildoma informacija užimtų mažesnį plotą. Šiame tyrime yra ištirta telefono galimybė nuskaityti *QR* kodą nuo dokumento ir kokiu greičiu tai atliekama.

QR kodo skaitymai atliekami su *OnePlus 3T* galine kamera(16MP, *CMOS* sensorius su galimybe atpažinti 1.12µm dydžio pikselius). Dokumentas atvaizduotas 1920x1080 rezoliucijos monitoriaus ekrane, ekrano atnaujinimų dažnis yra 60Hz. Telefonas atitrauktas 20cm atstumu nuo monitoriaus. *QR* kodą sudaro ne tik užšifruotų duomenų dalis, tačiau ir „tylioji zona“ kuri bendrai paėmus užima apie 12% skersmens vertikaliai ir horizontaliai. *QR* kodai atvaizduoti kaip kvadratai, o tai reiškia, kad skersmuo ir vertikaliai, ir horizontaliai yra vienodas.

6 lentelė QR kodo skaitymo duomenys

QR kodo plotis	1cm	1,5cm	2cm	2,5cm	3cm	3,5cm	4cm	5cm
Vidurkis, s	n/a	n/a	0,76	20,41	2,10	1,80	0,42	1,92
Mediana, s	n/a	n/a	0,66	10,34	1,04	1,09	0,38	1,00
Sėkmės kiekis	n/a	n/a	95%	0%	90%	100%	100%	100%

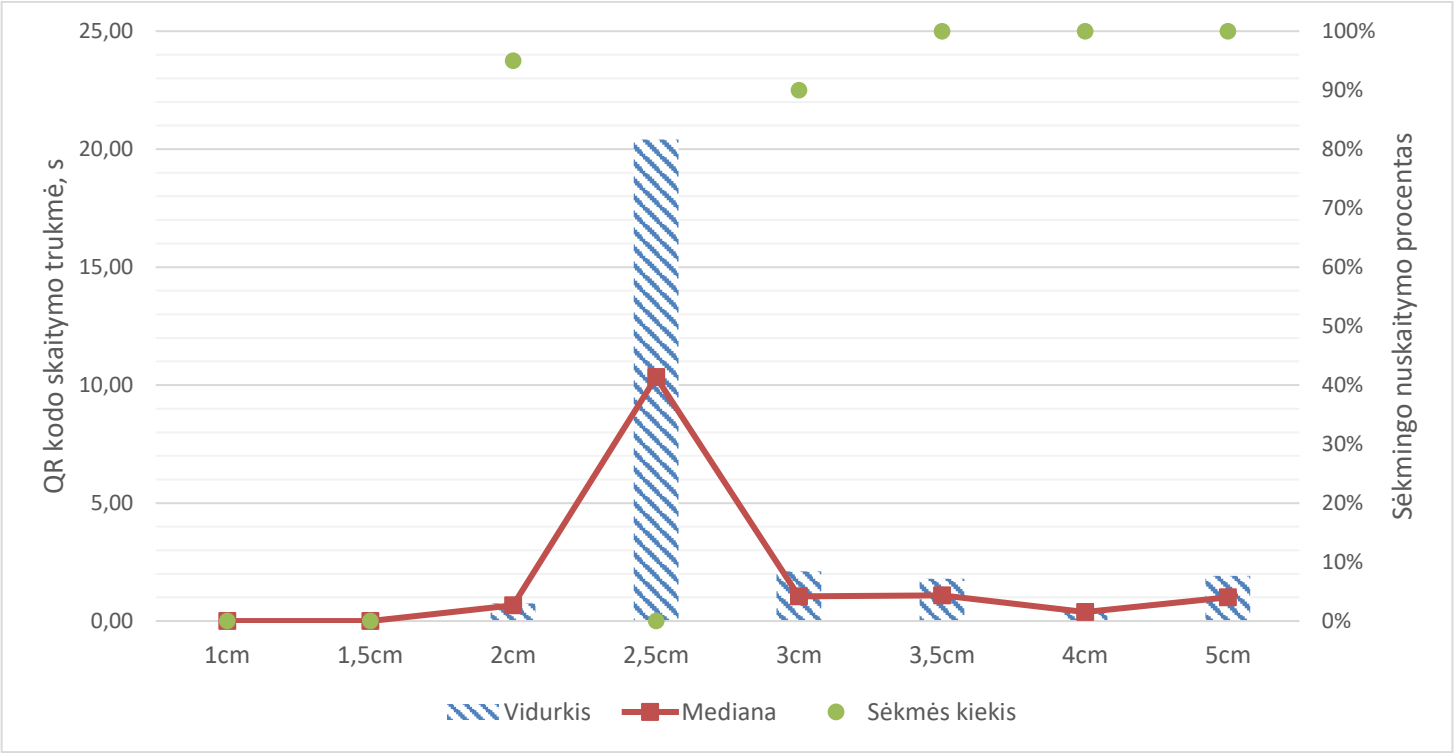
Ištyrus nuskaitymą ant monitoriaus ekrano ir gavus duomenis (6 lentelė) matoma, kad *QR* kodą jau galima nuskaityti pradedant nuo maždaug 2cm skersmens. Iki tol niekaip nepavyko telefonui iššifruoti, tai kas paslėpta šifrogramoje. Skaitant jau nuo 2cm užfiksuoti rezultatai rodo ne tik didelį greitį skaitant, bet ir gan gerą tikslumą – 95%. Procentai gali būti ir didesni, nes buvo atlikta 20 skaitymų per matuojamą *QR* kodo dydį.

Skaitant 2,5cm skersmens kodą pasireiškia anomalija. Telefonui niekaip nuskaityti *QR* kodo. Detektoriai niekaip nesugebėjo užfiksuoti net pozicionavimo kvadratų kampuose. Šis nesugebėjimas nuskaityti pasireikšdavo tuo, kad skaitytuvas ilgai fokusuodavo į skaitomą laukelį, tačiau kai galų gale pavykdavo atpažinti kažkokią duomenų seką, jis neteisingai nuskaitydavo kodą palaikydamas tai *UPC_E* tipo brūkšniniu kodu ir pateikdamas klaidingus duomenis. Tai gali būti dėl to, kad būtent atvaizduojant būtent šitokio dydžio *QR* kodą jie šiek tiek išsikreipia ir kai kurie moduliai skiriasi pagal išmatavimus todėl tai gali stipriai atsiliepti ant mažesnio dydžio *QR* kodų. Taip pat prisideda ir tai, kad *QR* kodas atvaizduojamas monitoriuje ir dėl nuolatinio ekrano vaizdo atnaujinimo ir mirksėjimo telefono kamerai iškyla sunkumų skaityti vaizdą. Šiek tiek padidinus vaizdą (10%-20%) *QR* kodas tampa nuskaitomas.

Tikrinant tolimesnius *QR* kodus neteko susidurti su didesnėmis problemomis. Skaitymo laikas svyravo, tačiau patikimumas vis didėjo. Laiko svyravimai taip pat galėjo iškilti dėl monitoriaus

galimybių atvaizduoti skirtingo dydžio *QR* kodus. Taip pat ir dėl telefono sugebėjimo greičiau sufokusuoti vaizdą nepaisant ekrano mirksėjimo.

Grafinė informacija apie nuskaitymų sėkmę pateikiama 16 pav. diagramoje.



16 pav. QR kodo skaitymų monitoriuje rezultatai grafike

Diagramoje 16 pav. pateikiami visi elementai, kuriuos pavyko nuskaityti yra apytiksliai vienodo dydžio. Laikas matuojamas sekundėmis, ir nors skirtumas tarp kai kurių matavimų atrodo skiriasi penkiagubai (0,42s ir 2,1s), tačiau palyginus su absoliučiai blogiausiu rezultatu skirtumai atrodo mažesni.

Grafike taip pat pateikiama ir išvesta mediana, kuri tiksliau parodo kiek galima tikėtis užtrukti skaitant *QR* kodą. Taip yra dėl to, kad kartais bandant nuskaityti kodą vienas ar du kartai gali pasirodyti neįprastai ilgai trunkantys ir taip gerokai padidinti bendrą skaitymo laiko vidurkį, nors realybė tokia, kad didžiąją dalį atvejų skaitymas trunka trumpesnį laiką nei gali pasirodyti matant vidurkį. Remiantis grafiko informacija galima pastebėti, kad vidurkis dažniausiai būna dvigubai didesnis nei mediana, o tai leidžia tikėtis, kad dažniausiai skaitant *QR* kodą sugaištas laikas bus trumpesnis nei rodo vidurkis.

Atvaizduoti procentai parodo, kad jeigu jau pavyksta pirmą kart nuskaityti QR kodą tai galima tikėtis ir toliau panašios sėkmės šansų nuskaitant kodą. Didinant *QR* kodą galima tikėtis didesnių šansų teisingai nuskaityti kodą. Šis skaičius teoriškai neturėtų pasiekti 100%, nes dėl pačių įvairiausių trukdžių gali kartais nepasisekti teisingai nuskaityti *QR* kodo teisingai. Dėl bandymų kiekio ir juose užfiksuotų sėkmės atvejų atvaizduota 100%, nors realybėje tai būtų tikslingiau atvaizduota kaip >99%.

3.2.3. QR kodo nuskaitomumas popieriuje

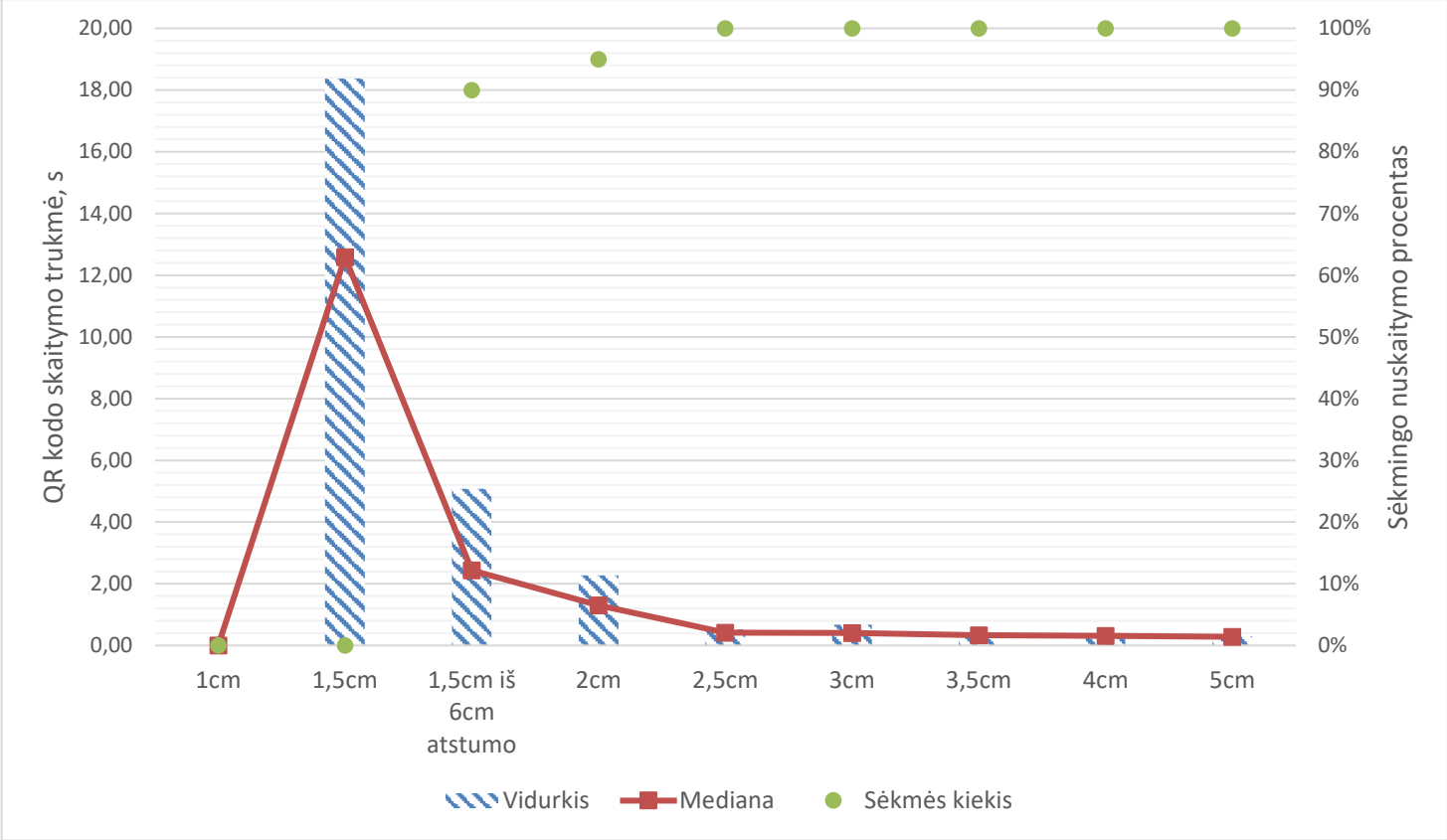
Šiame tyrime tikrinama kaip pavyksta nuskaityti *QR* kodus juos atspausdinus ant popieriaus lapo.

Tyrimas atliktas naudojant tą pačią *OnePlus 3T*, 16MP kamerą. QR kodai atspausdinti spausdintuvu, kurio raiška – 1200*dpi* (taškų per colį). Aplinkos apšvietimas atliekant tyrimą buvo apie 50 *lx* – kaip įprasto svetainės kambario. Skanuojant *QR* kodai kaip ir praėjusiame tyrime taip pat yra 20 centimetrų atstumu nuo kameros. Papildomai atliktas 1,5cm skersmens *QR* kodo skanavimas jam esant 6 cm atstumu nuo kameros.

7 lentelė QR kodų skanavimo popieriuje rezultatai

	1cm	1,5cm	1,5cm iš 5cm atstumo	2cm	2,5cm	3cm	3,5cm	4cm	5cm
Vidurkis	0,00	18,38	5,07	2,27	0,52	0,67	0,35	0,32	0,30
Mediana	0,00	12,59	2,44	1,30	0,41	0,40	0,33	0,31	0,28
Sėkmės kiekis	0%	0%	90%	95%	100%	100%	100%	100%	100%

Iš pateiktų duomenų 7 lentelėje, akivaizdu, jog yra nedidelis pranašumas skaityti iš popieriaus lyginant su skaitymu iš monitoriaus (6 lentelė). Bandant skaityti 1,5 cm skersmens *QR* kodą iš 20 cm atstumo dar nepavyko iššifruoti duomenų, nes kamera dar nesugebėjo atskirti *QR* kodo modulių. Pats nuskanavimas pavyko, tačiau gauti duomenys buvo neteisingi, todėl laikoma, kad bandymai nepavyko. Tačiau pabandžius tai padaryti iš artimesnio 6cm atstumo kodą pavyko nuskaityti gerokai greičiau ir pakankamai dideliu 90% patikimumu. Didėjant *QR* kodo modulio dydžiui patikimumas ir skaitymo greitis tolygiai vis didėjo, pažymint tik kiek didesnį šuolį tarp 2cm ir 2,5cm *QR* kodo skaitymo.



17 pav. QR kodo skaitymų ant popieriaus rezultatai grafike

Grafikas (17 pav.) puikiai iliustruoja tolygiai didėjantį *QR* kodo nuskaitomumą ant popieriaus lapo. Pirmas stulpelis kur pademonstruotas 1cm *QR* kodo nuskaitomumas sudėti nuliai. Taip yra dėl to, kad nuskaityti tokio dydžio *QR* kodo nepavyko.

Reikia pastebėti, kad priešingai nei skaitant iš monitoriaus, nuskaičius *QR* kodą su mažesniais moduliais bus nuskaitomi ir didesnių modulių *QR* kodai, nes nebėra įtakos kamerai, kaip ekrano mirgėjimas.

Žiūrint į 2cm bei 2,5cm iš 20cm bei 1,5cm skersmens iš 6cm atstumo *QR* kodo skaitymų rezultatus, taip pat kaip ir tyrime apie nuskaitymą iš monitoriaus (16 pav.) matoma, kad mediana maždaug dvigubai mažesnė už vidurkį ir tai perduoda tą pačią išvadą, kad *QR* kodo skaitymas tipiškai turėtų užtrukti mažiau nei gali pasirodyti iš vidurkio. Su likusiais kitais *QR* kodais tokio reiškinio nebėra, nes visi bandymai skaityti *QR* kodą užtrukdavo labai panašiai. Tai išduoda, kad pasiekus tam tikrą ribą *QR* kodo skaitymas toliau išlieka panašus ir tobulėjimas didinant modulio dydį gali būti neryškus.

Jeigu kodų skaitymas atliekamas žmogaus, o ne kompiuterio tada dešimtųjų sekundės dalių skirtumai yra praktiškai nereikšmingi. Turint tai omenyje ir pasitelkiant gautus duomenis galima nuspręsti koks *QR* kodo modulio dydis yra pakankamai tinkamas spausdinimui ant dokumento.

3.3. Išvados

Pasinaudojant projektavimo dalyje nubraižytomis schemomis, sukurta dalinė e-sąskaitų sistemos prototipo realizacija. Sukurta sąskaitų faktūrų generavimo sistema laikoma nutolusiame serveryje, taip pat pagalbinė Java programa, kuri bus kviečiama norint pasirašyti sąskaitos faktūros duomenų failą, naudojantis *ECDSA* asimetriniu raktu. Taip pat sukurta ir Android mobilioji programėlė, kuri gebės skanuoti *QR* kodą esantį ant dokumento, parsisiųsti *XAdES* dokumentą iš serverio, išvesti informaciją apie pasirašiusįjį ir prireikus – pačiam pasirašyti dokumentą.

Norint atlikti tyrimus buvo papildomai panaudojama *QR* kodų generavimo biblioteka galinti kurti vektorinės grafikos brūkšninius kodus, tam kad kuriant įvairaus dydžio *QR* kodus jie būtų atvaizduojami kuo tiksliau monitoriuje ir ant popieriaus lapo. Skirtingų raktų poveikiui *QR* kodo duomenų tankiui tirti buvo pridėta galimybė pasirašinėti dokumentus su *RSA* arba *DSA* raktais.

Pirmo tyrimo metu buvo tikrinama kaip pasirašymas su skirtingais raktais gali paveikti *QR* kodo duomenų tankį, kuriame yra užkoduojama visa informacija reikalinga dokumento peržiūrai, parašo patvirtinimui ir pasirašymui. Tyrimui naudojami buvo 2048 bitų ilgio *RSA* ir *DSA* raktai, 256 bitų ilgio *ECDSA* raktas. Visais atvejais buvo pasirašoma ta pati reikšmė.

Gauti rezultatai rodo, kad pagal duomenų sukūrimą rikiuojant nuo daugiausiai iki mažiausiai naudojami raktai ėjo tokia tvarka: *DSA, RSA, ECDSA*. Nors *DSA* privatusis raktas užėmė mažiausiai vietos, tačiau viešasis raktas buvo toks ilgas, kad jis nustelbė kitus du raktus pagal sukuriamą duomenų kiekį. *RSA* sugeba sutalpinti parašo informaciją mažesniame simbolių kiekyje, tačiau pagal šį kriterijų neprilygta *ECDSA* parašui. Kadangi *ECDSA* pagal dėl saugesnio raktų generavimo algoritmo gali turėti trumpesnio ilgio raktą tuo pačiu užtikrinant tokį patį arba didesni saugumo lygį nei *RSA* ir *DSA* algoritmai, todėl ir parašas bei viešasis raktas gali būti trumpesni. Užšifravus sukurtą parašų informaciją *QR* koduose, lengvai pasimato kaip padidėja duomenų tankis, o tai gali pakenkti bandant nuskaityti *QR* kodą.

Antro ir trečio tyrimų metu buvo tiriama telefono galimybė nuskaityti *QR* kodą nuo dokumento. Pirmu atveju buvo bandoma skaityti nuo monitoriaus ekrano, o kitu – nuo popieriaus lapo. Abiem atvejais telefonas buvo atitrauktas vienodu atstumu, o užkoduotų duomenų kiekis buvo toks pat – 340 simbolių, t. y. tiek kiek yra sukuriama duomenų pasirašant su *ECDSA* parašu ir naudojant *SHA-256* santrauką, pridedant 50 simbolių *URL* adresui ir 6 simbolius skyrikliams.

Ištyrus skanavimo galimybes atrasta, kad skanuoti ant popieriaus yra daug geriau, nes pasiekiamas didesnis tikslumas, galima nuskanuoti mažesnio dydžio *QR* kodus ir didesniu greičiu. Kadangi *QR* kodo nuskaitomumas priklauso nuo modulio dydžio, todėl dedant ant dokumento mažo dydžio *QR* kodus kur būtų didelis duomenų tankis reikia turėti arba didelės raiškos monitorių, arba didelės raiškos spausdintuvą. Bandant skaityti *QR* kodą su 0,2 mm pločio moduliais (1,5cm viso *QR* kodo plotis) jau iškyla sunkumų. Skaitant nuo monitoriaus ekrano nepavyko to padaryti net sumažinus

atstumą iki *QR* kodo, tačiau skaitant nuo popieriaus lapo pavyko tai padaryti 6 cm atstumu. Didinant *QR* kodą didėjo ir skaitymo patikimumas ir greitis, tačiau kaip liudija rezultatai, dar gali pasitaikyti anomalijų skaitant nuo monitoriaus dėl galimo ekrano mirgėjimo, kai tam tikro pločio *QR* kodo nuskaityti nepavyksta. Todėl, kuriant dokumentą reikia atkreipti dėmesį, kaip klientas skaitys dokumentą, ir nuspręsti dėl naudojamo *QR* kodo dydžio, nes tai gali įtakoti parašo patvirtinimo galimybes.

4. IŠVADOS

1. Atlikta probleminės srities analizė parodė darbo aktualumą, nes rinkoje esančių produktų integracija su mobiliais įrenginiais negali pasiūlyti kai kurių paslaugų, pvz., spartus parašo tikrinimas ir pasirašymas ant spausdintų ir elektroninių dokumentų.
2. Sukurta darbo struktūrinė schema įrodė sistemos pritaikomumą. Nubraižytos schemos parodo skirtingų sistemos komponentų sąveiką, tuo įrodydamos, kad sistemos veikimas nėra pernelyg komplikuotas.
3. Parašų kūrimui pasirinkta naudoti eliptinių kreivių kriptografinį metodą. Šis metodas minimizuoja sukuriamo parašo simbolių kiekį ir sumažina informacijos kiekį, kurį reikės užšifruoti *QR* kode.
4. Atlikus el. parašo informacijos minimizavimą, sukurta optimali el. sąskaitų informacijos struktūra naudojama *QR* kode, leidžianti sudėti visą reikalingą informaciją parašo patvirtinimui *QR* kode. Šią struktūrą sudaro el. parašas, viešasis raktas, dokumento santrauka, dokumento *URL* adresas ir skyrikliai padedantys išskirti duomenis.
5. Sukurtas prototipas, parodo realią veikiančią sistemą, veikiančią pagal aprašytus metodus. Kūrimui naudoti įrankiai, įrodo, kad šiai sistemai sukurti užtenka pasitelkti lengvai prieinamus, nemokamus įrankius.
6. Atlikti tyrimai parodė sukurtos sistemos panaudojimo galimybes ir pasirinktų metodų veiksmingumą. Pirmasis tyrimas pateikia duomenis kaip *ECDSA* 256 bitų raktas, pagal sukuriamos parašo informacijos kiekį lenkia *RSA* 2048 bitų ir *DSA* 2048 bitų raktus, sukurdamas atitinkamai 2,5 ir 3,9 karto mažiau informacijos. Antrajame ir trečiajame tyrimuose atrasta, kad *QR* kodo skaitymas nuo popieriaus lapo trunka mažiau laiko ir užtikrina didesnius šansus sėkmingai nuskaityti *QR* kodą, nei skaitant nuo monitoriaus.

5. LITERATŪRA

[1] D. Miles, „The Paper Free Office - dream or reality?,“ 2012. [Tinkle]. Pasiekiama: http://www.aiim.org/pdfdocuments/IW_Paper-free-Capture_2012.pdf. [Kreiptasi 19 05 2017].

[2] D. Fillingham, „A Comparison of Digital and Handwritten Signatures,“ MIT, 1997. [Tinkle]. Pasiekiama: http://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall97-papers/fillingham-sig.html. [Kreiptasi 19 05 2017].

[3] G. Samuolis, „Informacinių technologijų naudojimas įmonėse 2016 m.,“ 22 07 2016. [Tinkle]. Pasiekiama: https://osp.stat.gov.lt/informaciniai-pranesimai?articleId=4581046. [Kreiptasi 15 11 2016].

[4] „EDMS - Electronic Document Management System,“ [Tinkle]. Pasiekiama: http://www.edms.com/. [Kreiptasi 15 01 2016].

[5] Laserfiche, Laserfiche, 2007. [Tinkle]. Pasiekiama: http://www2.laserfiche.com/pdf/brochures/documentmgntover.pdf. [Kreiptasi 15 01 2016].

[6] M. J. Hancock, „Reducing Paper Consumption Will Drive Down Costs and Improve Workflows,“ 22 12 2008. [Tinkle]. Pasiekiama: https://www.gartner.com/doc/845212/reducing-paper-consumption-drive-costs. [Kreiptasi 15 01 2016].

[7] P. W. G. IT, „The Principles of Document Management,“ 2012. [Tinkle]. Pasiekiama: http://www.statetechmagazine.com/sites/default/files/the-principles-of-document-management.pdf. [Kreiptasi 15 01 2016].

[8] R. Tripathi ir S. Agrawal, „Comparative Study of Symmetric and Asymmetric Cryptography Techniques,“ *International Journal of Advance Foundation and Research in Computer,* t. 1, nr. 6, pp. 5, 8, 06 2014.

[9] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev ir P. Zimmermann, „Factorization of a 768-bit RSA modulus,“ 18 02 2010. [Tinkle]. Pasiekiama: https://eprint.iacr.org/2010/006.pdf. [Kreiptasi 19 12 2016].

[10] E. Barker ir A. Roginsky, „Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths,“ *NIST Special Publication 800-131A,* p. 5, 01 2011.

[11] N. A. Kofahi, „An Empirical Study to Compare the Performane of some Symmetric and Asymmetric Ciphers,“ *International Journal of Security and its Applications,* t. 7, nr. 5, pp. 9-13, 2013.

[12] T. Roeder, „Asymmetric-Key Cryptography,“ [Tinkle]. Pasiekiama: http://www.cs.cornell.edu/courses/cs5430/2013sp/TL04.asymmetric.html. [Kreiptasi 09 03 2017].

[13] E. Sakalauskas, T. Blažauskas ir K. Lukšys, Elektroninių dokumentų ir duomenų sauga, Kaunas: Vitae Litera, 2008, pp. 31,35-36,47,55-58.

[14] A. V. Meier, „The ElGamal Cryptosystem,“ 08 06 2005. [Tinkle]. Pasiekiama: http://wwwmayr.in.tum.de/konferenzen/Jass05/courses/1/papers/meier_paper.pdf. [Kreiptasi 03 04 2017].

[15] N. M. S. Iswari, „Key generation algorithm design combination of RSA and ElGamal algorithm,“ įtraukta *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Tangerang, 2016.

[16] E. Barker, „Recommendation for Key Management,“ *NIST Special Publication 800-57,* t. Part 1, nr. Revision 4, pp. 51-53, 01 2016.

[17] C. Kopf, *Cryptographic Hash Functions*, Hannover, 2007, p. 8.

[18] Y. Wang, J. Chen ir D. He, „A new collision attack on MD5," 2009. [Tinkle]. Pasiekiama: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4908582. [Kreiptasi 26 02 2017].

[19] M. Hassan, A. Khalid, A. Chattopadhuay, C. Rechberger, T. Güneysu ir C. Paar, „New ASIC/FPGA Cost Estimates for SHA-1 Collisions," įtraukta *2015 Euromicro Conference on Digital System Design*, 2015.

[20] *QR Code Essentials,* DENSO ADC, 2011.

[21] qrcode.com, „Information capacity and versions of the QR code," [Tinkle]. Pasiekiama: http://www.qrcode.com/en/about/version.html. [Kreiptasi 19 06 2016].

[22] J. C. Cruellas, G. Karlinger, D. Pinkas ir J. Ross, *XML Advanced Electronic Signatures (XAdES),* 2003.

[23] „How does DocuSign work?," DocuSign, [Tinkle]. Pasiekiama: https://www.docusign.com/products/electronic-signature/how-docusign-works. [Kreiptasi 29 03 2017].

[24] HelloSign, [Tinkle]. Pasiekiama: https://www.hellosign.com/. [Kreiptasi 29 03 2017].

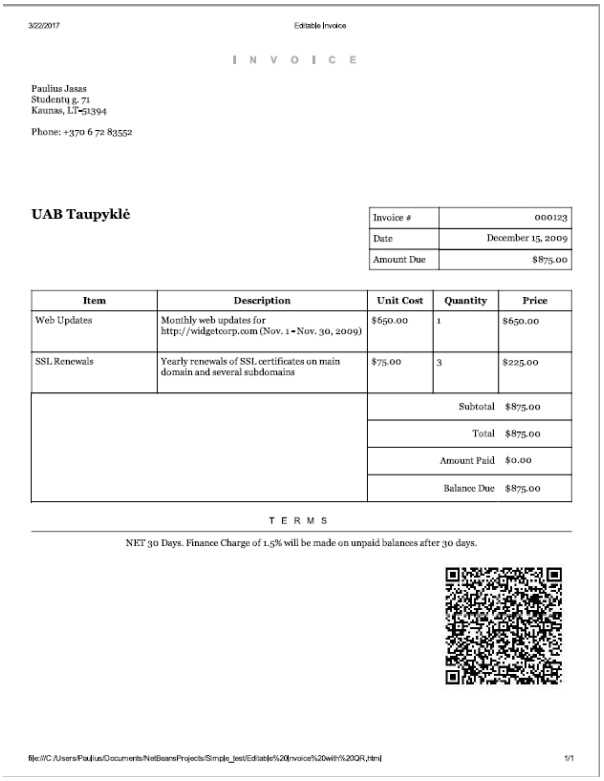
[25] „ADSS Signing Server Features," Ascertia DSS, [Tinkle]. Pasiekiama: http://www.ascertia.com/products/adss-signing-server/features. [Kreiptasi 30 03 2017].

[26] „Adobe Sign," Adobe Sign, [Tinkle]. Pasiekiama: https://helpx.adobe.com/sign/faq.html#AdobeSigncapabilities. [Kreiptasi 30 03 2017].

6. PRIEDAI

6.1. Prototipo realizacijos pavyzdys

Per organizacijos valdymo sistemą pareikalavus ir *SF* generavimo sistemoje sukurtas dokumentas spausdintinėje formoje atrodo taip:



3/22/2017 Editable Invoice

INVOICE

Paulius Jasas
Studentų g. 71
Kaunas, LT-31394

Phone: +370 6 72 83552

UAB Taupyklė

Invoice #	000123
Date	December 15, 2009
Amount Due	$875.00

Item	Description	Unit Cost	Quantity	Price
Web Updates	Monthly web updates for http://widgetcorp.com (Nov. 1 - Nov. 30, 2009)	$650.00	1	$650.00
SSL Renewals	Yearly renewals of SSL certificates on main domain and several subdomains	$75.00	3	$225.00
			Subtotal	$875.00
			Total	$875.00
			Amount Paid	$0.00
			Balance Due	$875.00

TERMS

NET 30 Days. Finance Charge of 1.5% will be made on unpaid balances after 30 days.

file:///C:/Users/Paulius/Documents/NetBeansProjects/Simple_test/Editable%20Invoice%20with%20QR.html 1/1

18 pav. Atspausdinta e-sąskaita faktūra su QR kodu

To paties dokumento duomenys laikomi *XAdES* faile serveryje, o jis gali atrodyti taip kaip pavaizduota 19 pav.

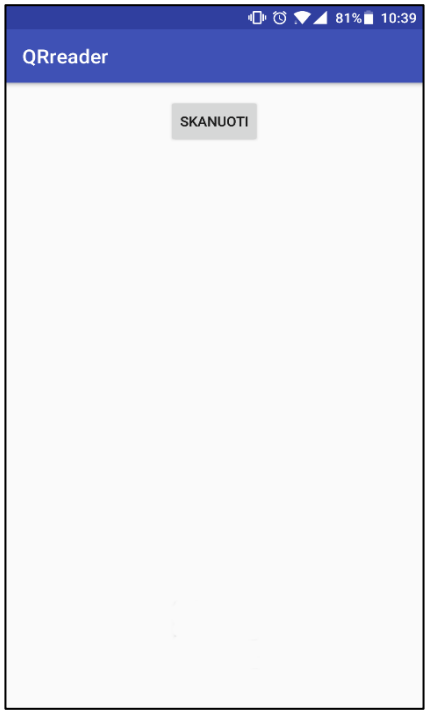
```
<SignedDoc format="XADES-XML" version="1.0">
  <DataFile Id="D0" ContentType="EMBEDDED_BASE64" Name="invoice_1.docx" Created="2017-01-19T09:15:40" Modified="2017-01-19T09:14:24" Size="48154">
        ...
  </DataFile>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#" Id="S0">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <Reference URI="#D0">...</Reference>
    </SignedInfo>
    <SignatureValue Id="S0-SIG">...</SignatureValue>
    <KeyInfo>
      <KeyValue>
        <RSAKeyValue>
          <Modulus>...</Modulus>
          <Exponent>AQABAA==</Exponent>
        </RSAKeyValue>
      </KeyValue>
      <X509Data>
        <X509Certificate>...</X509Certificate>
      </X509Data>
    </KeyInfo>
    <Object>
      <QualifyingProperties>
        <SignedProperties Id="S0-SignedProperties" Target="#S0">
          <SignedSignatureProperties>
            <SigningTime>2017-01-19T09:16:07</SigningTime>
            <SigningCertificate>
              <Cert Id="S0-CERTINFO">
                <CertDigest>
                  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <DigestValue>3Qpek5fygVry6qcYFxdmP/Yg4II=</DigestValue>
                </CertDigest>
                <IssuerSerial>0</IssuerSerial>
              </Cert>
            </SigningCertificate>
            <SignaturePolicyIdentifier>
              <SignaturePolicyImplied/>
            </SignaturePolicyIdentifier>
            <Signer>
              <Name>Jasas Paulius, Kauno technologijos universitetas</Name>
            </Signer>
          </SignedSignatureProperties>
          <SignedDataObjectProperties/>
        </SignedProperties>
        <UnsignedProperties>
          <UnsignedSignatureProperties/>
        </UnsignedProperties>
      </QualifyingProperties>
    </Object>
  </Signature>
</SignedDoc>
```

19 pav. Pasirašytas dokumentas

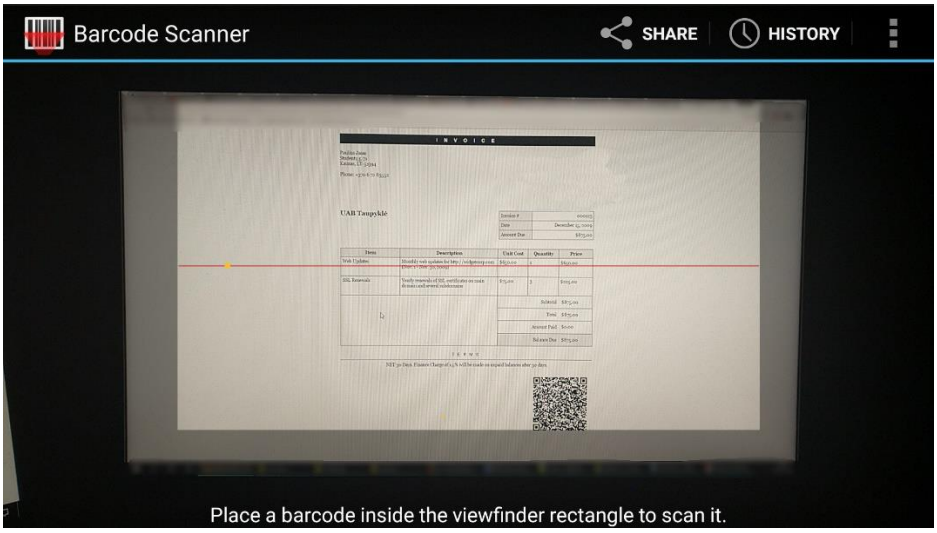
Pasirašytas dokumentas talpina visa reikiamą informaciją, apie pasirašiusįjį asmenį ir dokumento turinį. Tada galima patikrinti jį nuskaičius *QR* kodą kuriame talpinama informacija leidžia tai padaryti.

Norint spausdintą arba atidarytą kompiuteryje dokumentą perskaityti ir patikrinti parašą arba pasirašyti reikia turėti telefone programėlę kuri galės atlikti pageidaujamus veiksmus.

Atsidarius programėlę iš karto matosi mygtukas, kurį paspaudus atveriama brūkšniniu kodu skanavimo programėlė skanuosianti *QR* kodą (21 pav.).



20 pav. Pagrindinis aplikacijos langas

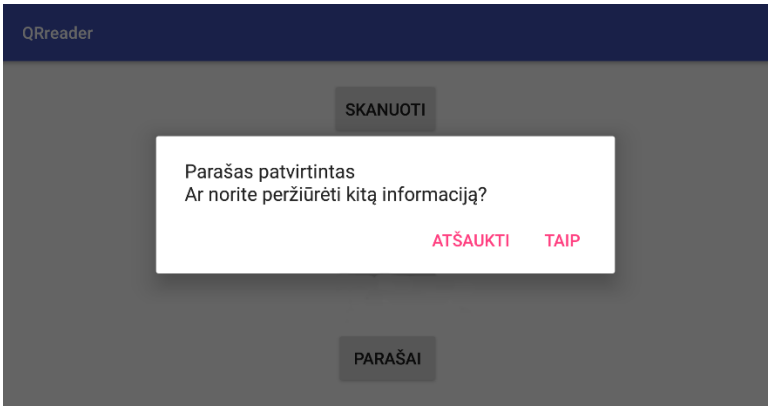


21 pav. QR kodo skanavimas

Atspausdinus jau pasirašytą dokumentą su *QR* kodu matomą 18 pav. galime atsidarę aplikaciją paspausti mygtuką Skanuoti matomą 20 pav. Paspaudus atsidaro *QR* kodo skanavimo langas. Jis veikia taip pat kaip, bet kuri kita *QR* kodų skanavimo programėlė t. y. tereikia tik nutaikyti kamerą į *QR* kodą ir palaukti kol bus užfiksuotas *QR* kodas. Jis suprogramuotas naudojantis *zxing* biblioteka, ir jeigu nebus jau telefone instaliuotos brūkšninių kodų skanavimo programėlės tai pirma prašys tokią parsisiųsti.

Nuskanavus iškart nuskaitomas *QR* kode užkoduotas tekstas. Jis yra išskaidomas į atskiras dalis: parašą, viešąjį raktą, dokumento santrauką ir *URL* adresą. Pasinaudojant pirmosiomis trejomis

nuskaityto teksto dalimis yra patikrinamas parašas ar jis yra teisingas ir nesugadintas ir tada išvedamas atsakymas vartotojui į ekraną (22 pav.).



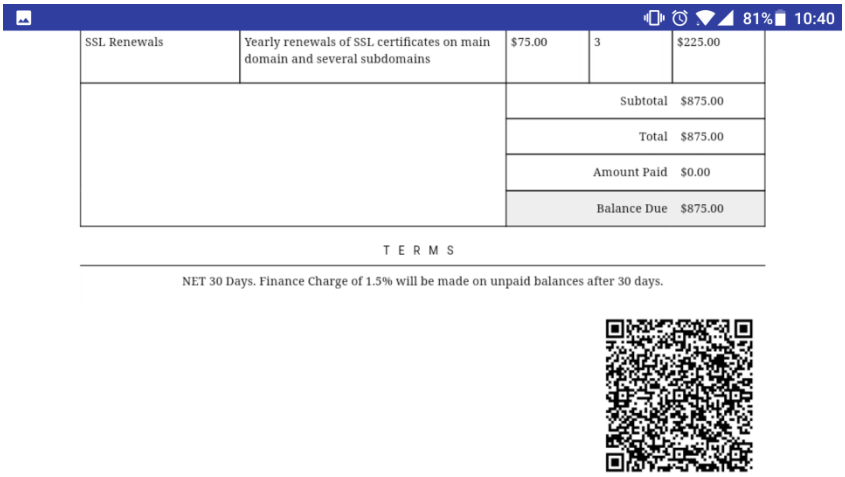
22 pav. Atsakymas apie parašą vartotojui

Paspaudus mygtuką *Taip* pasinaudojant iš *QR* kodo gautu *URL* adresu parsiunčiamas *XAdES* dokumento turinys. Iš jo paimama sertifikato informacija ir išvedama į ekraną vartotojui (23 pav.).

23 pav. Pasirašiusiojo peržiūros langas

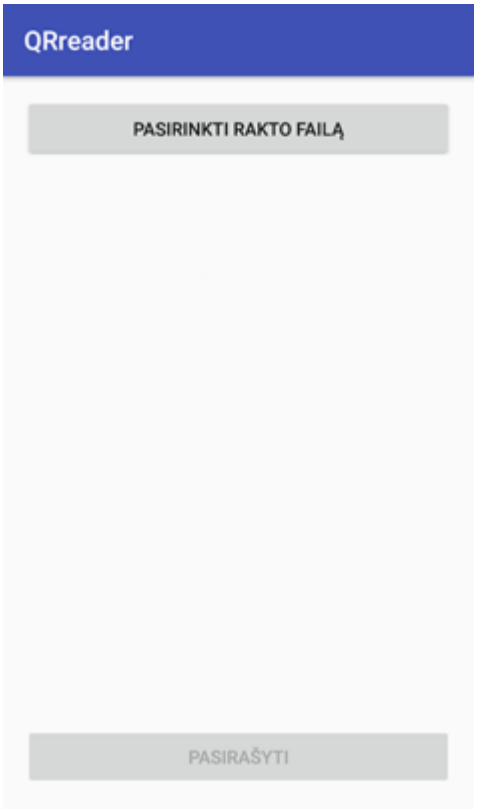
Tuomet vartotojas turi du pasirinkimus – peržiūrėti nuskaitytą dokumentą arba jį pats pasirašyti.

Pasirinkus failo peržiūra yra paimami duomenys iš *XAdES* dokumento *DataFile* elemento(19 pav.), dešifruojami iš *base64* formato ir naudojantis tuo pačiu šablonu kaip ir generuojant sąskaitą faktūrą serveryje, sukuriama to dokumento peržiūra pačiam telefone naudojant *Android* aplikacijų programavimo sąsajoje prieinamu *Webview* lango komponentu, kuris leidžia atvaizduoti pateiktą *HTML* kodą taip lyg tai būtų internetinis puslapis naršyklėje (24 pav.).



24 pav. SF peržiūra

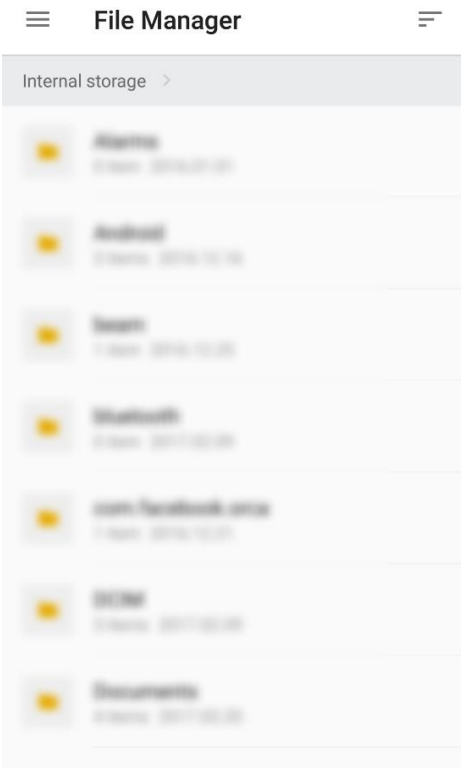
Pasirinkus dokumento pasirašymą (23 pav.) atveriamas failo pasirašymo langas (25 pav.)



25 pav. Failo pasirašymo langas

Šiame lange vartotojas gali vykdyti dokumento pasirašymą. Paspaudus „Pasirinkti rakto failą“ mygtuką iššoka pasirinkimo langas (26 pav.) kur vartotojas pasirenka failą kur yra saugomas jo privatus raktas. Pasirinkus vartotojui nurodomas kelias iki to failo, tam kad matytų, kad failas tikrai parinktas ir galėtų pamatyti kelią iki jo. Tada atsirakina mygtukas Pasirašyti, kuris iki šiol buvo užrakintas nuo paspaudimų. Iškart po paspaudimo programėlė prašo vartotojo įvesti privataus rakto slaptažodį. Tuomet yra paimama dokumento santraukos reikšmė ir ji pasirašoma. Parašas, sertifikatas,

autorizacijos duomenys ir kita informacija yra siunčiama per žiniatinklio paslaugas į *SF* generavimo sistemą nutolusiame serveryje. Tenai yra patikrinami autorizacijos duomenys. Po duomenų patikrinimo yra atnaujinamas tos sąskaitos faktūros *XAdES* failas. Sukuriamas ir pridedamas dar vienas *QR* kodas rodantis į antrą parašą. Sugeneruojamas *PDF* dokumentas ir atsiunčiamas atgal į telefoną.



26 pav. Failo pasirinkimo langas

Taip yra užbaigiamas visas dokumento sukūrimo, nuskaitymo, pasirašymo ciklas. Nuo šio momento toks dokumentas yra laikomas tik peržiūros tikslais. Jis gali būti archyvuojamas ir saugomas sistemoje. Jeigu dokumentui buvo sukurtas apribojimas turėti tik du parašus jis jau daugiau nebegali būti pasirašomas.