KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Lina Šukauskaitė

ELEKTRONINIŲ PINIGŲ SISTEMOS MOBILIESIEMS ĮRENGINIAMS SUKŪRIMAS IR TYRIMAS

Baigiamasis magistro projektas

Vadovas
Prof. dr. Eligijus Sakalauskas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

ELEKTRONINIŲ PINIGŲ SISTEMOS MOBILIESIEMS ĮRENGINIAMS SUKŪRIMAS IR TYRIMAS

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

Vadovas
(parašas) Prof. dr. Eligijus Sakalauskas
(data)

Recenzentas
(parašas) dr. Kęstutis Lukšys
(data)

Projektą atliko
(parašas) Lina Šukauskaitė
(data)

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

(Fakultetas)

Lina Šukauskaitė

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Elektroninių pinigų sistemos mobiliesiems įrenginiams sukūrimas ir tyrimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 17 m. gegužės 22 d.

Kaunas

Patvirtinu, kad mano, **Linos Šukauskaitės,** baigiamasis projektas tema „Elektroninių pinigų sistemos mobiliesiems įrenginiams sukūrimas ir tyrimas“ 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)*

Šukauskaitė, Lina. Elektroninių pinigų sistemos mobiliesiems įrenginiams sukūrimas ir tyrimas. *Magistro* baigiamasis projektas / vadovas prof. dr. Eligijus Sakalauskas; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: kriptografija

Reikšminiai žodžiai: *e. pinigai, e. mokėjimai, e. piniginė*

Kaunas, 2017. 43 p.

SANTRAUKA

Naudojimasis internetine bankininkyste ir sparčiai didėjančios mobiliųjų mokėjimų apimtys rodo, kad patogesnių ir lankstesnių atsiskaitymo būdų poreikis taip pat auga. Esamų mokėjimo sistemų pagrindinis trūkumas yra tas, kad jos saugumą užtikrina vartotojo anonimiškumo sąskaita. Šią problemą galėtų išspręsti elektroniniai pinigai, kurie savo savybėmis artimiausi gryniesiems. Jau yra sukurta keletas elektroninių pinigų sistemų modelių, tačiau jie sunkiai pritaikomi mobiliesiems įrenginiams dėl pinigų duomenų kiekio augimo po kiekvienos transakcijos. Taip pat, tos sistemos, kurios užtikrina visišką anonimiškumą, neturi dalumo savybės ir atvirkščiai.

Šiame darbe pateikta ir detaliai aprašyta banko kontroliuojama elektroninių pinigų sistema, kuri suteikia vartotojui anonimiškumą pardavėjo atžvilgiu, išpildo pinigų dalumo savybę ir yra apsaugota nuo duomenų kiekio augimo. Ištirtas tokios sistemos efektyvumas lyginant su Brandso schema ir pateikta mokėjimo protokolų realizacija *Octave* programavimo kalba.

Šukauskaitė, Lina. *Creation and Analysis of Electronic Money System for Mobile Devices*: *Master's* thesis in Informatics. / supervisor assoc. prof. dr. Eligijus Sakalauskas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Cryptography

Key words: *e-money, payment system, e-wallet, e-coin*

Kaunas, 2017. 43 p.

SUMMARY

Fast growing amount of mobile payments shows that there is a demand for more effective and advanced payment methods. The biggest withdraw of our current electronic payment systems is that customers have to loose their anonimity if they want to be secure. This problem could be solved with electronic money (e-money) system.

There are several models already created, but they are not adapted to mobile devices which have limited calculation recourses. More over, if a system provide full anonimity, it doesn't have property of divisibility and vice versa. In this work there is an e-money system presented that provides partial anonimity of purchaser, but keeps e-money divisibility property and is protected from e-money growing in size during transactions. There is also provided detailed description of payment protocols, system effectiveness evaluation and comparison with Brands model. The system realisation was done with Octave language.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

ĮVADAS

Tobulėjant technologijoms ir populiarėjant e. komercijai, vis labiau auga patogesnių, greitesnių ir lankstesnių atsiskaitymo metodų poreikis. Nemažai ekspertų kalba apie visišką grynųjų pinigų netolimoje ateityje atsisakymą, todėl dėmesio centre vis dažniau atsiduria elektroniniai pinigai, kurie savo savybėmis artimiausi gryniesiems. Siekiant paskatinti elektroninių pinigų sistemų kūrimą ir vystymąsi Europos Sąjungoje buvo priimta jas reglamentuojanti direktyva. Šiuo metu viena ryškiausių tendencijų yra sparčiai augantis išmaniųjų telefonų populiarumas bei su juo susijusios didėjančios mobiliųjų mokėjimų apimtys. Prognozuojama, kad iki 2020 metų mobilioji komercija bus aktualiausias segmentas, todėl šioje infrastruktūroje elektroniniai pinigai ir jų realizavimas mobiliosiose e. piniginėse turės didelį potencialą.

Yra sukurta nemažai teorinių modelių, tačiau praktinis jų panaudojimas kol kas dar labai ribotas. Kadangi dabartinių elektroninių mokėjimų didžiausias trūkumas yra anonimiškumo nebuvimas, dauguma sistemų kūrėjų seka elektroninių pinigų išradėjo Davido Chaumo privatumo samprata ir kuria atjungties režimu (angl. *Off-line*) veikiančias sistemas, kurios užtikrina visišką anonimiškumą ir mokėjimo neatsekamumą, tačiau tokios sistemos laikomos mažiau saugiomis ir didinančiomis nusikalstamų veikų riziką. Taip pat dažnai pamirštama gryniesiems pinigams būdinga dalumo savybė, kuri vartotojui suteikia galimybę iš banko gautą e. monetą išleisti dalimis.

Šio darbo analizės dalyje pateikiami esamų elektroninių mokėjimo sistemų tipai, reikalavimai bei trūkumai, detalizuota elektroninių pinigų sistema, teoriniai modeliai ir jų palyginimas. Darbe bus pateiktas banko kontroliuojamos mokėjimo sistemos modelis, užtikrinantis pirkėjo anonimiškumą pardavėjo atžvilgiu ir pinigų dalumą. Darbo rezultatuose bus pateikta sistemos realizacija, įvertinti reikalingi skaičiavimo resursai ir pateikiamos išvados apie efektyvumą.

1. ELEKTRONINIŲ MOKĖJIMO SISTEMŲ ANALIZĖ

1.1. Analizės tikslas

Auganti e. komercija sąlygoja didėjantį pažangesnių mokėjimo sistemų poreikį. Šiuo metu vis dar populiariausia ir plačiausiai taikoma mokėjimo sistema yra paremta mokėjimo kortelėmis. Didžiausias trūkumas yra tas, kad tokia sistema neišpildo ir saugumo, ir vartotojo anonimiškumo reikalavimų vienu metu, t. y. sistema saugi vartotojo anonimiškumo sąskaita. Šią problemą išsprendžia elektroniniai pinigai. Šiame skyriuje pateikiamos analizės tikslas yra išskirti elektroninių mokėjimo sistemų tipus, jų trūkumus ir privalumus.

1.2. Elektroninių mokėjimo sistemų tipai

Įvairiuose literatūros šaltiniuose elektroninių mokėjimų sistemos klasifikuojamos skirtingai. Berry‘is Schoenmakersas tai padarė ko gero tiksliausiai, išskirdamas du pagrindinius tipus [1]:

- mokėjimo taisyklėmis paremtos sistemos (angl. *Payment by instruction*);
- išankstinio mokėjimo elektroniniai pinigai (angl. *Prepaid electronic cash*).

Mokėjimo taisyklėmis paremtos sistemos – tai pinigų pervedimai, kai bankas, pagal mokėtojo duotas instrukcijas, perveda pinigus iš jo sąskaitos į gavėjo sąskaitą. Šios kategorijos pavyzdžiai yra kredito ir debeto kortelės, elektroniniai čekiai. Transakcijas gali atlikti tik teisėti banko sąskaitų valdytojai. Siekiant apsaugoti duomenų perdavimą nesaugiu interneto tinklu naudojama elektroninių parašų sistema ir viešųjų raktų infrastruktūra (angl. *Public Key Infrastructure, PKI*).

Išankstinio mokėjimo sistemos yra konceptualiai artimos gryniesiems pinigams. Į šią kategoriją patenka telefono kortelės, lustinės kortelės (angl. *smart cards*) ir elektroniniai pinigai (angl. *e-cash, e-money*). Vartotojo sąskaita debetuojama iškart, kai tik pinigai patenka į elektroninę piniginę (kortelę ar mobilųjį įrenginį).

1.3. Elektroninių mokėjimo sistemų reikalavimai

Elektroninė komercija didžiulį populiarumą pelnė dėl atvirumo, greičio, anonimiškumo ir globalaus prieinamumo. Šios interneto savybės supaprastino labai daug procesų, reikalingų verslui realiu laiku. Kadangi visos finansinės operacijos atliekamos nesaugiu interneto tinklu, tikslumas, saugumas ir patikimumas mokėjimo sistemose tapo itin svarbus.

Profesorius Hsiao-Cheng‘as Yu savo analizėje išskyrė keturis aspektus, pagal kuriuos turėtų būti vertinama mokėjimo sistema [2]:

- **technologinis aspektas** – sistema turi būti lanksti, efektyvi, lengvai integruojama, saugi ir patikima. Finansinės operacijos sistemoje turi būti autorizuotos, išlaikyti privatumo, vientisumo, neišsiginamumo (angl. *non-repudiation*) savybes;
- **ekonominis aspektas** – operacijų kaštai (tai ypač svarbu renkantis sistemą mažos vertės mokėjimams), prieinamumas (sistema turi būti prieinama kiek įmanoma didesniam

vartotojui skaičiui), vertės mobilumas (vartotojas gali atsiskaityti skirtingose vietose, taip pat konvertuoti turimą vertę į kitą valiutą), finansinė rizika;

- **socialinis aspektas** – sistema turi užtikrinti vartotojo anonimiškumą, ja turi būti paprasta ir patogu naudotis visur, nepririšant vartotojo prie vieno konkretaus įrenginio;
- **institucinis aspektas** – be technologinių, ekonominių ir socialinių poreikių užtikrinimo, mokėjimo sistema taip pat privalo laikytis Vyriausybės nustatytų reglamentų ir įstatymų.

1.4. Elektroninių mokėjimo sistemų trūkumai

Kiekvienos mokėjimo sistemos sėkmė priklauso nuo to, kaip plačiai ji pritaikyta ir kiek vartotojų pasiekia. Žvelgiant iš šios perspektyvos, mokėjimo kortelėmis pagrįstos sistemos turi neginčijamą pranašumą, tačiau tobulėjant technologijoms, vartotojų poreikiai keičiasi ir mokėjimo kortelių industrijai darosi vis sunkiau juos patenkinti. Esminiai šios mokėjimo sistemos trūkumai [3]:

- **anonimiškumo nebuvimas**. Atlikta mokėjimo operacija atskleidžia nemažai prekybininkams naudingos informacijos;
- **mažas saugumo lygis**. Sąskaitos ir tapatybės duomenų perdavimas internetu nėra saugus;
- **dideli operacijų kaštai**. Kiekviena operacija kainuoja, ypač nuostolingi yra mažos vertės mokėjimai;
- **mažas operacijų greitis**. Tame pačiame banke mokėjimai atliekami per 5 – 30 min, tačiau pavedimai į kitus bankus užtrunka dar ilgiau arba tenka sumokėti papildomą mokestį;
- **sudėtingumas**. Būtina atsidaryti sąskaitą, taip pat, siekiant apsaugoti vartotoją, įvedami įvairūs limitai, apribojantys galimų operacijų skaičių, maksimalią pervedamų pinigų sumą ir pan.

1.5. Elektroninių pinigų sistemos analizė

Elektroniniai pinigai – tai kriptografinėmis savybėmis pasižyminti mokėjimo sistema, sukurta siekiant pateikti alternatyvą gryniesiems pinigams, išlaikant pagrindines jų savybes. Pirmasis anonimiškus elektroninius pinigus 1998 metais pristatė Davidas Chaumas.

1.5.1. Teisiniai aktai

Naujausia ir šiuo metu galiojanti Europos Parlemento ir Tarybos direktyva 2009/110/EB buvo išleista 2009 m. rugsėjo 16 d. Joje elektroniniai pinigai apibrėžiami taip: „*elektroniniai pinigai – išleidėjui pateikiamu reikalavimu išreikšta, elektroninėse, įskaitant, magnetines, laikmenose saugoma piniginė vertė, kuri išleidžiama gavus lėšas, skirta mokėjimo operacijoms, kaip apibrėžta Direktyvos 2007/64/EB 4 straipsnio 5 punkte, atlikti ir priimama fizinių arba juridinių asmenų, neskaitant elektroninių pinigų išleidėjo.“*

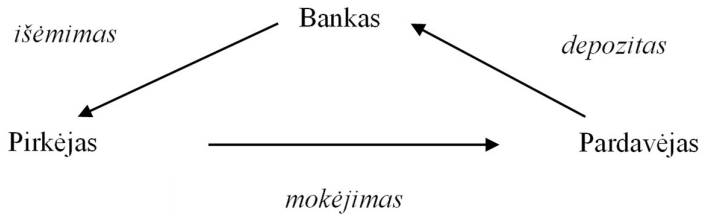
Taip pat šios direktyvos preambulės 8 punkte pažymėta, kad „Elektroninių pinigų apibrėžtis turėtų apimti elektroninius pinigus, kurie laikomi elektroninių pinigų turėtojui priklausančioje mokėjimo priemonėje arba nuotoliniu būdu laikomi serveryje ir elektroninių pinigų turėtojo valdomi per specialią elektroninių pinigų sąskaitą. Ta apibrėžtis turėtų būti pakankamai plati, kad netrukdytų technologinėms inovacijoms ir apimtų ne tik visus šiandien rinkoje egzistuojančius elektroninių pinigų produktus, bet ir produktus, kurie galėtų būti sukurti ateityje.“ Toks elektroninių pinigų sąvokos patikslinimas, kurio trūko ankstesnėje ES direktyvoje, įvedė aiškumo mokėjimų sistemų kūrėjams.

LR elektroninių pinigų ir elektroninių pinigų įstaigų įstatymas buvo priimtas 2011 m. gruodžio 22 d., o įsigaliojo 2012 m. sausio 1 d. Šiuo metu Lietuvoje yra trys įmonės, kurioms suteikta elektroninių pinigų įmonės licencija. Viena iš jų, gerai žinoma mokėjimų sistema „Paysera“ (UAB „EVP International“).

1.5.2. Protokolai

Elektroninių pinigų mokėjimo sistemose įprastai naudojami trys pagrindiniai protokolai, kurie modifikuojami atsižvelgiant į norimas e. pinigams suteikti savybes:

- pinigų išėmimo (angl. *Withdrawal*);
- mokėjimo (angl. *Payment*);
- depozito (angl. *Deposit*).



1.1 pav. E. monetos ciklas

E. pinigų protokolai gali būti vykdomi prijungties režimu veikiančiose sistemose (angl. *On-line*) arba atjungties (angl. *Off-line*). Prijungties sistemos pavyzdys – kredito / debito kortelių transakcijos. Kiekvieno mokėjimo metu pardavėjas susijungia su banku ir prašo patvirtini, kad e. moneta galiojanti ir ja nebandoma atsiskaityti antrą kartą. Atjungties sistemose toks susijungimas nebūtinas. Pardavėjas pats verifikuoja pirkėjo e. monetą, o depozito protokolas įvykdomas vėliau. Nors autonominė sistema praktiniu požiūriu yra patrauklesnė, vis dėl to ji siejama su mažesniu saugumu, didesne dvigubo išleidimo rizika.

1.5.3. Savybės

Saugumas (angl. *Security*). Saugumas reiškia, kad e. monetos apsaugotos nuo klastojimo ir dvigubo išleidimo (angl. *Double-spending*). Nuo klastojimo apsaugo vartotojo autentikavimo funkcijos, apimančios vartotojo identifikavimo, duomenų vientisumo ir transakcijos neišsiginamumo (angl. *Non-repudiation*) aspektus. Autentikavimo sistema paremta viešųjų raktų infrastruktūra.

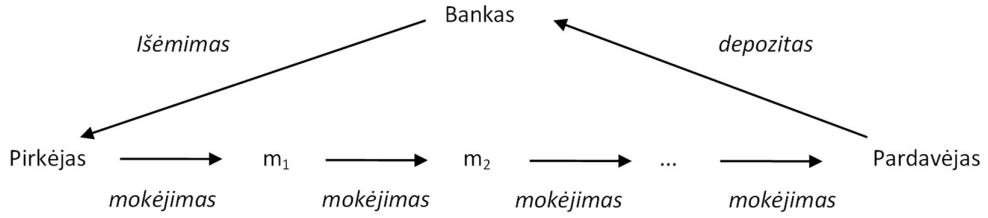
Tam, kad būtų išvengta dvigubo išleidimo rizikos, bankas duomenų bazėje kaupia informaciją apie jau panaudotas e. monetas ir kiekvienos transakcijos metu patikrina tokios e. monetos tinkamumą. Jei mokėjimas vyksta atjungties režimu, norint apsaugoti pardavėją, aptikus dvigubą išleidimą bankas turi turėti galimybę atskleisti mokėtojo tapatybę. Tokiu atveju pirkėjo anonimiškumas panaikinamas.

Anonimiškumas (angl. *Anonymity*). D. Chaumas šią savybę apibūdina kaip visišką mokėtojo anonimiškumą ir transakcijos neatsekamumą, t. y. bankas neturi galimybės nustatyti, kieno pinigai dalyvauja mokėjime. Ši savybė yra labai artima gryniesiems pinigams, tačiau tokį besąlyginį anonimiškumą suteikti elektroniniams pinigams yra pavojinga. Tai smarkiai padidintų pinigų plovimo, nelegalaus pirkimo, šantažo ir klastojimo riziką. Anonimiškumas labai glaudžiai susijęs su saugumu – kuo daugiau anonimiškumo, tuo mažiau saugumo ir atvirkščiai.

Dalumas (angl. *Divisibility*). Dalumo savybė suteikia galimybę iš banko gautą e. monetą išleisti dalimis, įsigyjant prekių, kurių bendra suma neviršija jos vertės. E. monetos dalijimo procese neturi dalyvauti patikima trečioji šalis.

Mobilumas (angl. *Portability*). Elektroniniai pinigai, kaip ir grynieji, nėra susieti su konkrečia fizine vieta. Jie gali būti lengvai perduodami kompiuterių tinklais į įrenginį arba atvirkščiai.

Perleidžiamumas (angl. *Transferability*). Perleidžiamumo savybė leidžia vartotojui išleisti gautą e. monetą be banko įsikišimo. Mokėjimas vadinamas perleidimu, kai e. monetos gavėjas jį panaudoja kitam mokėjimui. Jei mokėjimo sistema kiekvienai e. monetai gali įvykdyti nors vieną tokį perleidimą, tuomet laikoma, kad ši sistema išpildo perleidžiamumo savybę.



1.2 pav. Perleidžiamos e. monetos ciklas

Kiekvieno perleidimo metu e. monetoje paauga duomenų kiekis (angl. *grow in size*), nes į ją įrašoma vykdomos transakcijos informacija. Dėl šios priežasties sistema turi apriboti galimų perleidimų skaičių. Taip pat tokiose sistemose palankesnės sąlygos pinigų plovimui, nes kiekvienas

perleidimas atideda dvigubo išleidimo tikrinimo mechanizmą. Jis bus aptiktas tik tuomet, kai dvi vienodos e. monetos grįš depozito metu į banką.

1.5.4. Teorinių modelių analizė ir palyginimas

Šioje darbo dalyje pateikiami pagrindinių literatūroje randamų modelių apžvalga. Kiekvienas iš jų skirtingu lygmeniu išpildo elektroninių pinigų savybes. Jų palyginimas pateikiamas skyriaus pabaigoje esančioje lentelėje.

- **Chaumo, Fiato ir Naorio modelis**

Pati svarbiausia elektroninių pinigų savybė yra anonimiškumas. Chaumo, Fiato ir Naorio modelyje jis užtikrinamas bankui naudojant RSA akląjį parašą. Tokio parašo esmė yra ta, kad bankui neatskleidžiama ant ko jis pasirašo. Tai reiškia, kad bankas neturi galimybės susieti e. monetos išėmimo protokolo su depozito protokolu.

Aklojo parašo schema [4 p. 23-24]:

1. pirkėjas pasirenka maskavimo faktorių r, kuris tenkina sąlygą $DBD(r,n) = 1$, kur (e,n) yra banko viešasis raktas. Tuomet savo užklausą m e. monetai gauti užmaskuoja: $m' = mr^e modn$;
2. bankas pasirašo savo privačiuoju raktu d: $s' = (m')^d modn = (mr^e)^d modn$;
3. pirkėjas nuima maskavimo faktorių: $s = s'/r(modn)$;
4. pirkėjas naudoja gautą banko parašą $s = m^d$ apmokėjimui atlikti.

Kadangi r yra atsitiktinis, bankas negali nustatyti m, o taip pat ir pirkėjo tapatybės. Taip užtikrinamas visiškas anonimiškumas, tačiau atsiranda galimybė sukčiavimui. Pavyzdžiui, bankas galės nustatyti pakartotiną e. monetos panaudojimą, tačiau negalės nustatyti sukčiaujančiojo tapatybės. Siekiant to išvengti, Chaumo Fiato ir Naorio modelis siūlo įvesti papildomus parametrus. Bankas aklai pasirašys ant užklausos tik tuomet, kai įsitikins, kad yra įtraukta dalis pirkėją identifikuojančios informacijos.

Dvigubo išleidimo prevencijai naudojamas atskyrimo – parinkimo metodas (angl. *Cut-and-Choose*) [5]. Šis metodas pagrįstas sąžiningu dalijimusi:

1. Alisa padalina kažką per pusę;
2. Bobas pasiima vieną pusę sau;
3. Alisa pasiima sau likusią pusę. Pirmą žingsnį Alisa turi atlikti sąžiningai, nes ji nežino, ką pasirinks Bobas antrame žingsnyje.

- **Fergusono modelis**

Fergusono modelis, kaip ir prieš tai minėtoji sistema, paremtas RSA. Skirtumas tas, kad naudojamas ne įprastas aklasis RSA parašas, bet atsitiktinių imčių aklasis parašas (angl. *Randomized*

blind signature). Kadangi šiuo atveju tiek pirkėjas, tiek bankas į pranešimą įtraukia atsitiktinius duomenis, bankas, nors ir vis dar nematydamas ant ko pasirašo, žino, kad pasirašomas pranešimas nebuvo suformuotas kenkėjiškai. Atlikdamas mokėjimą, pirkėjas privalo atskleisti dalį eilutės, susietos su jo tapatybe. Jei pirkėjas bandys jau panaudotą e. monetą išleisti dar kartą, jis atskleis likusią eilutės dalį ir bus identifikuotas banko.

Atsitiktinių imčių aklojo parašo schema [6]:

1. bankas paskelbia savo viešąjį raktą (e, n), vienkryptę funkciją f ir atsitiktinį skaičių $g \in Z_n^*$;
2. pirkėjas pasirenka atsitiktinį skaičių $a_1 \in Z_n^*$, du maskavimo faktorius σ ir γ ir apskaičiuoja $\gamma^e a_1 g^\sigma$. Gautą vertę nusiunčia bankui;
3. bankas pasirenka savo atsitiktinį a_2 ir nusiunčia pirkėjui;
4. pirkėjas atsako su $f(a_1 a_2) - \sigma$;
5. bankas sudaugina $\gamma^e a_1 g^\sigma$ su a_2 ir $g^{f(a_1 a_2)^{-\sigma}}$;
6. pirkėjas nuima maskavimo faktorių γ, kad gautų porą $(a, (a g^{f(a)})/e)$. Skaičius $a = a_1 a_2$ vadinamas baziniu parašo skaičiumi.

- **Brandso modelis**

Brandso sistema naudoja Schnorro autentifikacijos schemą, susidedančią iš sistemos inicijavimo, vartotojo registracijos, vartotojų autentifikacijos bei elektroninio parašo protokolų [7]. Tai pirmasis elektroninių pinigų modelis, paremtas reprezentacijos uždaviniu, kuris leido atsisakyti „*cut-and-choose"* metodo*,* naudojamo kitose sistemose.

Reprezentacijos uždavinys

Tarkime, kad $k > 1$ ir $1 \leq a_i \leq q$, kiekvienam $i = 1 \ldots k$. Reprezentacijos uždavinys yra turint generatorių vektorių $\{g_1, \ldots, g_k\} \in G_q$ rasti indeksų vektorių $\{a_1, \ldots, a_k\}$, kad būtų tenkinama lygybė: $g_1^{a_1} g_2^{a_2} \ldots g_k^{a_k} \equiv h (\mathrm{mod}\ p)$.

Vektorius $\{a_1, \ldots, a_k\}$ yra vadinamas reprezentacija. Jis suvedamas į uždavinį: $a_1 \log q_1 + a_2 \log q_2 + \cdots + a_k \log q_k \equiv \log h (\mathrm{mod}\ p)$.

Jei turime k elementų reprezentacijoje $\{a_1, \ldots, a_k\}$, tai tuomet egzistuoja q^{k-1} skaičiaus h reprezentacijos.

Mokėjimo procesas:

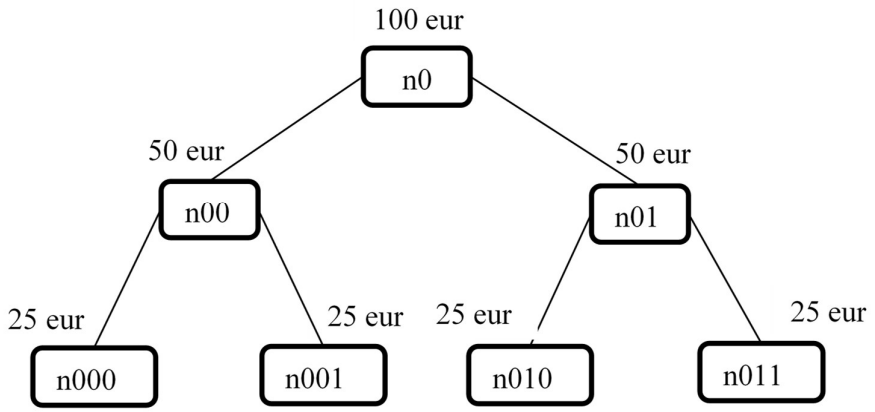
1. pirkėjas įvykdo sąskaitos atidarymo protokolą, t. y. sugeneruoja ir apskaičiuoja vertę *I*, kuri bus saugoma banke kartu su identifikuojančia informacija bei sąskaitos numeriu;
2. pirkėjas išsprendžia reprezentacijos uždavinį. Jei bankas jį patvirtina, pereinama prie išėmimo protokolo;
3. išėmimo protokolo metu, sukuriama e. moneta, turinti banko parašą, kuris autentifikuoja pirkėją ir dalį jo užmaskuotos identifikuojančios eilutės. Šią eilutę pirkėjas naudos mokėjimo

protokole. Jei jau panaudota e. moneta bandoma atsiskaityti dar kartą, atskleidžiama ir likusi identifikuojančios eilutės dalis, t. y. tapatybė;

4. mokėjimo protokolo metu, pirkėjas nusiunčia e. monetą pardavėjui. Jis patikrina banko parašą. Jei viskas gerai, išsiunčia pirkėjui iššūkį, kuris kiekviename mokėjime turi būti unikalus. Pirkėjas atsiunčia apskaičiuotą atsakymą. Pardavėjas jį patikrina ir priima mokėjimą;
5. depozito protokolo metu, pardavėjas siunčia bankui pasirašytą informaciją apie gautą e. monetą, iššūkį ir atsakymą. Bankas tuos duomenis patikrina ir jei viskas gerai, kredituoja pardavėjo sąskaitą.

- **Okomoto modelis**

Visi trys minėtieji modeliai užtikrina pilną anonimiškumą, tačiau neišpildo pinigų dalumo savybės. Sprendimą pasiūlė Okomoto, sukurdamas dvejetainiu medžiu pagrįstą sistemą, kuri leidžia e. monetą suskaidyti į smulkesnes dalis [8]. Svarbiausia, kad bendra dalių suma neviršytų e. monetos vertės. 1.3 pav. pavaizduotas 100 eur dvejetainis medis.



1.3 pav. Okomoto dvejetainis medis

Okomoto savo sistemoje siūlo laikytis dviejų taisyklių, kurios garantuoja, kad bus užtikrinta apsauga nuo permokos, t. y. vartotojas negalės išleisti daugiau, nei bendra e. monetos vertė:

1. jei vienas medžio mazgas jau išleistas, tai nei aukščiau esantis, nei žemiau esantys mazgai jau negali būti panaudoti. Pvz., jei mokėjime panaudota e. monetos dalis n00, tai iš medžio panaikinami n000, n001 ir n0;
2. kiekvienas medžio mazgas gali būti panaudotas tik vieną kartą.

Mokėjimo procesas

1. Pirkėjas atsidaro sąskaitą, su iš banko gautais duomenimis susigeneruodamas licenciją.

2. Išėmimo protokolo metu uždedamas aklasis RSA parašas ant pirkėjo licencijos dalies, sujungtos su sugeneruotu atsitiktiniu dydžiu, kuris užtikrina e. monetos unikalumą.
3. Mokėjimo protokolo metu autentifikuojama e. moneta. Taip pat pirkėjas atskleidžia šiek tiek informacijos apie tai, kurie mazgai iš e. monetos medžio jau išleisti ir pasirenka mazgą, kuris neprieštarauja antrajai Okomoto sistemos taisyklei.
4. Depozito protokolo metu, pardavėjas siunčia bankui mokėjimo stenogramą.

- **Pateiktų modelių palyginimas**

1.1 lentelė. E. pinigų modelių palyginimas

	Chaumo, Fiato ir Naoro	**Fergusono**	**Brandso**	**Okomoto**
Saugumo pagrindas	Faktorizavimo problema	Faktorizavimo problema	Diskrečiojo logaritmo problema	Faktorizavimo problema
Anonimiškumo lygis	aukštas	aukštas	aukštas	žemas
Mobilumo lygis	aukštas	vidutinis	aukštas	žemas
Perleidžiamumo savybė	nėra	yra	nėra	nėra
Dalumo savybė	nėra	nėra	nėra	yra
Komunikacijos sudėtingumas (kiek žingsnių reikia vienam mokėjimui atlikti)	8 (4 - išėmimas, 3 - mokėjimas, 1 - depozitas)	8 (4 - išėmimas, 3 - mokėjimas, 1 - depozitas)	10 (6 - išėmimas, 3 - mokėjimas, 1 - depozitas)	6 (2 - išėmimas, 3 - mokėjimas, 1 - depozitas)

Chaumo sistemos didžiausias privalumas tas, kad ji užtikrina pirkėjo anonimiškumą ir yra konceptualiai paprasčiausia, tačiau dėl naudojamo „*cut and choose*“ metodo nėra efektyvi, taip pat suteikia sąlygas lengviau apeiti apsaugos nuo dvigubo išleidimo mechanizmą. Šio modelio efektyvumas ir saugumas priklauso nuo to, kiek pirkėjas bankui siųs paketų K e. monetos išėmimo protokolo metu, nes tuos paketus reikia sugeneruoti, užmaskuoti akluoju parašu ir vėl atskleisti. Skaičiavimo resursus įtakoja ir tai, kad mokėjimo protokolo metu pirkėjas turi atsakyti į banko siunčiamą iššūkį generuodamas $K/2$ skirtingų atsakymų.

Fergusono schema išsprendė Chaumo saugumo spragą naudodama sudėtingesnį atsitiktinių imčių akląjį parašą, tačiau būtent dėl jo nukenčia sistemos efektyvumas. Galiojanti perleidžiamumo savybė smarkiai išaugina informacijos kiekį e. monetoje, tuo padidindama ir sukčiavimo riziką, nes e. monetai keliaujant, kiekvieną kartą sukuriama jos kopija.

Brandso sistema daugelio yra laikoma geriausia pirmiausia dėl to, kad tai pirmoji schema, leidusi atsisakyti „*cut and choose*“ metodo. Antra, priešingai nuo kitų metodų, dėl naudojamo Schnorro autentifikavimo protokolo ji gali būti pritaikyta ir elipsinių kreivių algoritmuose. Diskrečiojo logaritmo problema ir duomenims apsaugoti naudojama maišos funkcija užtikrina aukštesnį sistemos

saugumo lygį. Nepaisant visų privalumų, tai matematiškai gana sudėtinga sistema, todėl nėra tokia populiari kaip Chaumo.

Okomoto pasiūlė dalius elektroninius pinigus, tačiau jie sunkiai pritaikomi mobiliosioms e. piniginėms – norint sumokėti sumą x iš turimos e. monetos, kaskart reikia sugeneruoti dvejetainio medžio mazgus ir atsakyti į pardavėjo iššūkį. Taip pat vartotojas turi prisiminti, kurie medžio mazgai jau išleisti. Nepaisant to, didžiausias šios sistemos trūkumas yra tas, kad tinkamai neapsaugoma vartotojo tapatybė. Kiekviena e. moneta susiejama su vartotojui išduodama licencija, o visi e. monetos mazgai medyje susiję vienas su kitu. Šios sistemos savybės leidžia gana nesudėtingai susekti mokėjimus ir identifikuoti pirkėją.

1.6. Kitų autorių realizuoti modeliai

Paulius Palevičius savo baigiamajame magistro darbe „Elektroninių pinigų modelio realizacija standartinėse ir ribotų asimetrinių funkcijų sistemose“ [5] pateikė praktinę minėto Brandso modelio realizaciją mobiliajame telefone *Nokia 6212 Classic*. Žemiau pateikta keletas šio eksperimento išvadų:

- kompiuteryje tie patys mokėjimo protokolai vykdomi 100 kartų sparčiau nei mikroprocesorinėje kortelėje;
- bendras vartotojo e. monetos paėmimas ir išleidimas mikroprocesorinėje kortelėje užima 52 s, todėl autorius siūlo kortelėje atlikti tik tą protokolo dalį, kurioje naudojamas privatusis raktas. Likusi dalis vykdoma Java ME platformoje. Laikas šiuo atveju sutrumpėja iki 15 – 20 s.

Šio atlikto tyrimo rezultatai naudingi atliekant palyginimą ir įvertinant skirtumus, sąlygojamus skirtingų telefonų techninių parametrų.

1.7. Darbo tikslas ir uždaviniai

Išanalizuotose elektroninių pinigų sistemose matyti, kad jei užtikrinamas pilnas vartotojo anonimiškumas, tai neišpildyta pinigų dalumo savybė arba atvirkščiai – ten, kur ji išpildyta, vartotojo tapatybė nėra tinkamai apsaugota.

Darbo tikslas yra sukurti tokį elektroninių pinigų modelį, kuris išsaugotų pirkėjo anonimiškumą bent pardavėjo atžvilgiu, išpildytų dalumo savybę, o saugumą ir tinkamą priežiūrą užtikrintų bankas. Siekiama įvertinti, koks tokios sistemos protokolų vykdymo laikas, lyginant su jau sukurtais modeliais. Žemiau pateiktas detalus uždavinių sąrašas:

- naudojamų kriptografinių bei matematinių metodų aprašymas;
- detalus siūlomos sistemos mokėjimo protokolų aprašymas;
- protokolų skaičiavimo resursų įvertinimas;
- efektyvumo rezultatų palyginimas su kitomis sistemomis;
- išvadų pateikimas.

1.8. Analizės išvados

Atsižvelgiant į mokėjimo sistemoms keliamus reikalavimus, esamų sistemų trūkumus bei rinkos tendencijas, galima teigti, kad elektroniniai pinigai yra viena geriausių ir perspektyviausių alternatyvų. Kol kas tiek aptartieji metodai, tiek dauguma kitų literatūroje pateiktų schemų, yra tik teoriniai modeliai, sunkiai pritaikomi mobiliosioms piniginėms. Didžiausias dėmesys juose yra skiriamas visiškam mokėjimo anonimiškumui ir neatsekamumui užtikrinti, tačiau tokiose sistemose realizuoti pinigų dalumą yra gana sudėtinga ir dažniausiai neefektyvu. Darbe siekiama pateikti paprastesnį, lengviau realizuojamą modelį, maksimaliai atsižvelgiant į vartotojų poreikius, teisines normas bei saugumo reikalavimus.

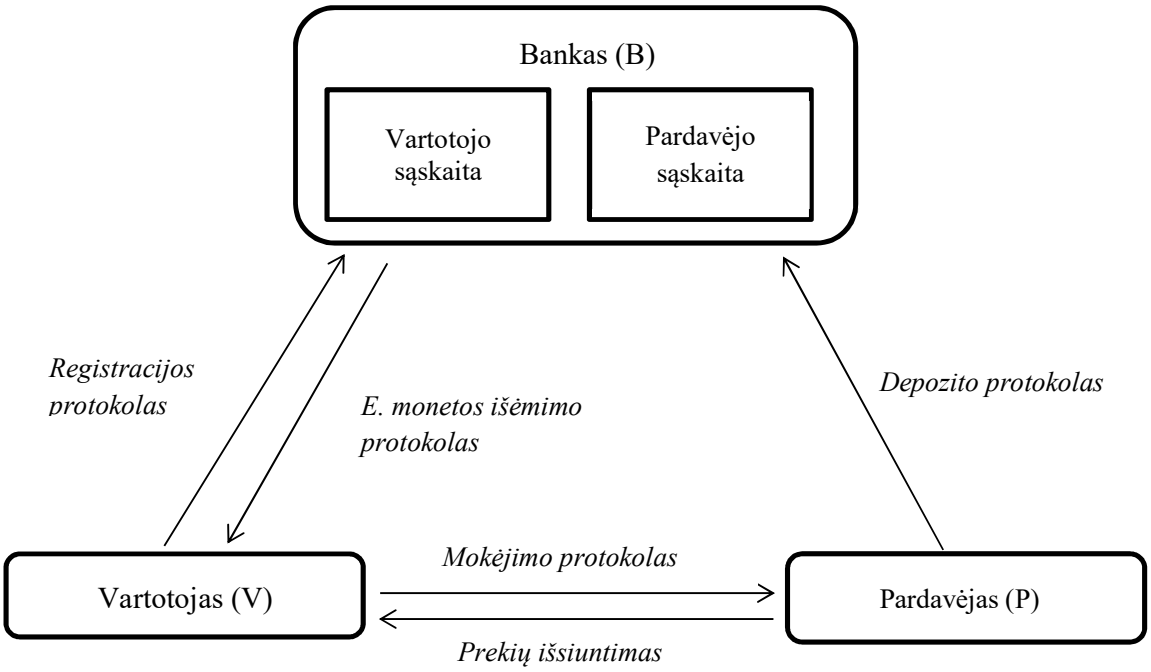
2. ELEKTRONINIŲ PINIGŲ SISTEMOS MOBILIESIEMS ĮRENGINIAMS MODELIS

2.1. Modelio architektūra

Kuriamą mokėjimo sistemą sudaro trys pagrindinės šalys: vartotojas (V), bankas (B), pardavėjas (P). Daroma prielaida, kad bankas valdo ir vartotojo, ir pardavėjo sąskaitas. Sistemos pinigų cirkuliaciją apibrėžia šie protokolai:

- registracijos;
- išėmimo;
- mokėjimo;
- depozito.

Šie protokolai bus detaliai aprašyti tolesniuose skyreliuose.



2.1 pav. Mokėjimo sistemos schema

2.2. Modelio savybės

Daugelis esamų elektroninių pinigų mokėjimo sistemų stengiasi išpildyti visiško anonimiškumo ir neatsekamumo, perleidžiamumo bei pilno veikimo atjungties režimu savybes. Tokios sistemos labai artimos gryniesiems pinigams, tačiau esminis jų trūkumas – e. monetos duomenų kiekio augimas. Taip pat jose pasigendama pinigų dalumo savybės, reikalingos ir naudingos realizuojant mobiliąsias e. pinigines, tačiau sunkiai realizuojamos tokiose sistemose. Siekiant to išvengti, modeliuojama sistema, kuri tik iš dalies atsisako kelių savybių, tačiau užtikrina ir anonimiškumą, ir pinigų dalumą.

Siūlomos sistemos savybės:

- **dalinis atjungties režimas**. Pilnas veikimas atjungties režimu užtikrina visišką anonimiškumą ir neatsekamumą, tačiau lieka minėta duomenų augimo problema bei saugumo spragos, didelė dvigubo išleidimo tikimybė. Kuriamoje sistemoje atjungties režimu bus vykdomi tik mokėjimo protokolai. Depozitas atliekamas dalyvaujant bankui, t. y. prisijungus. Tai leis bankui kontroliuoti mokėjimus bei tikrinti saugumo parametrus;
- **dalinis anonimiškumas**. Pilnas anonimiškumas reiškia, kad bankas žino tik tai, kiek vartotojas išsiima pinigų, tačiau nežino kur ir kaip juos išleidžia rinkoje. Kuriamame modelyje, siūlomas dalinis anonimiškumas – vykdant mokėjimo protokolą, pardavėjas negali identifikuoti pirkėjo ir sekti jo pirkinių istorijos;
- **pilna banko kontrolė**. Bankas yra atsakingas už vartotojų registravimą, autentifikavimą, sąskaitų valdymą bei saugią e. monetų cirkuliaciją. Daloma prielaida, kad banko skaičiavimo resursai yra pakankamai dideli tam, kad sektų atsiskaitymų istoriją, apsaugotų nuo dvigubo išleidimo bei identifikuotų nesąžiningus vartotojus;
- **dalumas**. Jei e. pinigai nėra dalūs, pirkėjui kiekvieną kartą juos išleidus reikia vėl kreiptis į banką arba e. piniginėje laikyti skirtingų nominalų e. pinigus, kaip ir atsiskaitant grynaisiais pinigais. Dalumas reiškia, kad išėmus iš banko tam tikros vertės e. pinigus, galima juos dalinti į tiek dalių, kiek reikia, ir atsiskaityti su pardavėju be grąžos ar papildomo banko įsikišimo;
- **perleidžiamumas.** Ši savybė reiškia, vartotojas išsiimtus e. pinigus gali išleisti atsiskaitydamas su keliais pardavėjais, t. y. be banko įsikišimo atlikti keletą skirtingų mokėjimo protokolų.

2.3. Kriptografinės funkcijos e. pinigų sistemoje

Siekiant realizuoti kuriamos sistemos dalinio anonimiškumo, e. monetos dalumo bei apmokėjimo atjungties režimu savybes, bus naudojami du kriptografiniai algoritmai – Paillier‘io schema duomenų šifravimui ir RSA e. parašo schema pasirašymui.

Šios schemos pasirinktos todėl, kad jomis galima modeliuoti homomorfinę sistemą, kuri leidžia atlikti saugius kriptografinius skaičiavimus nesaugioje aplinkoje. Homomorfinio skaičiavimo idėja:

užšifravus keletą skaičių, su jų šifrogramomis atlikus algebrines operacijas (tokias kaip sudėtis arba daugyba) ir iššifravus rezultatą, jis gaunamas toks pats, kaip ir veiksmus atlikus su tekstogramomis (nešifruotais skaičiais).

Elektroninių pinigų pranešimą žymėsime m, jo šifrogramą c, o šifrogramos parašą s. Žemiau išvardintos elektroninių pinigų sistemoje naudojamos funkcijos:

$Enc_{Pai}(m)$ – Paillier'io šifravimo funkcija tekstogramai m;

$\mathrm{Dec_{Pai}(c)}$ – Paillier'io iššifravimo funkcija šifrogramai c;

$Sig_{RSA}(c)$ – RSA parašo funkcija šifrogramai c, kuri lygi reikšmei s;

$Ver_{RSA}(s)$ – RSA parašo s tikrinimo funkcija ant šifrogramos c.

2.3.1. Paillier'io šifravimo schema

Paillier'io schema – tai asimetrinio šifravimo algoritmas, įprastai naudojamas elektroniniame balsavime, nes jo savybės leidžia sudauginus atskirus užšifruotus balsus gauti bendrą balsų sumą. Tai galima išnaudoti ir elektroninių pinigų sistemose. Žemiau pateikiama detalesnė informacija, kaip veikia ši schema [9].

Raktų poros generavimas

1. Atsitiktinai sugeneruojami du dideli pirminiai skaičiai p ir q, tenkinantys sąlygą: $\mathrm{DBD}(pq, (p-1)(q-1)) = 1$. Abu pirminiai skaičiai turi būti vienodo ilgio.
2. Apskaičiuojamas $n = p \cdot q$.
3. Apskaičiuojamas skaičius $g = n + 1$, kur $g \in \mathrm{Z}_{n^2}^*$.
4. Apskaičiuojamas $\lambda = \phi(n)$ ir $\mu = \phi(n)^{-1} \bmod n$, kur $\phi(n) = (p-1)(q-1)$.
5. Viešasis raktas bus $Pu_K = (n, g)$.
6. Privatusis raktas $Pr_K = (\lambda, \mu)$.

Užšifravimo funkcija

1. Tekstogramą žymėsime m. Čia $m \in \mathrm{Z}_n$.
2. Atsitiktinai pasirenkame r, kai $r \in \mathrm{Z}_n^*$.
3. Apskaičiuojama šifrograma: $c = (n+1)^m \cdot r^n \bmod n^2$.

Iššifravimo funkcija

1. Imama šifrograma c, kuri $c \in \mathrm{Z}_{n^2}^*$.
2. Apskaičiuojama tekstograma $m = L\left(c^{\lambda} \bmod n^2\right) \cdot \mu \bmod n$, kur funkcija L apibrėžiama kaip $L(u) = \frac{u-1}{n}$.

Homomorfizmo savybės

- Šifrogramų sandauga lygi tekstogramų sumai: $Enc(m_1) \cdot Enc(m_2) = Enc(m_1 + m_2) \bmod n$.
- Multiplikatyvumas – iššifruojant vienos tekstogramos šifrogramą pakeltą kita tekstograma, gaunama tekstogramų sandauga:

$$Dec(Enc(m_1, r_1)^{m_2} \bmod n^2) = m_1 m_2 \bmod n;$$
$$Dec(Enc(m_2, r_2)^{m_1} \bmod n^2) = m_1 m_2 \bmod n.$$

2.3.2. RSA algoritmas

RSA e. parašo algoritmas bus naudojamas Paillier‘io sistema užšifruotų duomenų pasirašymui. RSA taip pat turi homomorfizmo savybę, kuri reiškia, kad sudauginus šifrogramų parašus, gaunamas parašas ant šifrogramų sandaugos [10], t. y. $Sig_{RSA}(c_1) \cdot Sig_{RSA}(c_2) = Sig_{RSA}\big((c_1 \cdot c_2) \bmod (n^2)\big) = Sig_{RSA}(c)$, kai $c = (c_1 \cdot c_2) \bmod (n^2)$, visiems $c, c_1, c_2 \in \mathbb{Z}_{n^2}^*$, kur $Sig_{RSA}(c)$ reiškia RSA parašą ant šifrogramos $c \in \mathbb{Z}_{n^2}^*$.

Tai įrodoma naudojantis šiomis ypatybėmis: $Sig_{RSA}(c_1) \cdot Sig_{RSA}(c_2) = {c_1}^d {c_2}^d \bmod n = (c_1 \cdot c_2)^d \bmod n = Sig_{RSA}(c_1 \cdot c_2)$.

Raktų generavimo algoritmas

1. Generuojami du dideli vienodo ilgio atsitiktiniai pirminiai skaičiai P ir Q.
2. Apskaičiuojamas $N = PQ$ ir $\phi(N) = (P-1)(Q-1)$.
3. Pasirenkamas atsitiktinis sveikasis skaičius e, $1 < e < \phi(N)$ toks, kad būtų tenkinama lygybė $\mathrm{DBD}\big(e, \phi(N)\big) = 1$.
4. Naudojantis išplėstiniu Euklido algoritmu, apskaičiuojamas unikalus sveikasis skaičius d, $1 < d < \phi(N)$ toks, kad $ed \equiv 1 \bmod(\phi(N))$.
5. Viešasis raktas $\mathrm{Pu_K}$ yra (N, e) ir privatusis raktas $\mathrm{Pr_K}$ yra (d).

Parašo generavimas

1. Tegul $c \in \mathbb{Z}_{n^2}^*$, $c < N$ yra šifrograma gauta užšifravus Paillier‘io algoritmu. Ji tenkina tokią lygybę $\mathrm{DBD}(c, N) = 1$.
2. Formuojamas parašas $s = c^d \bmod(N)$.

Parašo tikrinimas

1. $s \in \mathbb{Z}_N^*$ yra parašas, kurį reikia patikrinti.
2. Apskaičiuojamas $c' = s^e \bmod(N)$.
3. Patikrinama, ar $c' = c$. Jei ne, tuomet parašas atmetamas (funkcijos $Ver_{RSA}(s)$ reikšmė bus “Ne”). Priešingu atveju, reikšmė bus “Taip”.

2.3.3. Ryšys tarp viešųjų parametrų

Modulis $n = pq$, naudojamas Paillier'io šifravimo schemoje turėtų būti parenkamas toks, kad $n^2 < P$ ir $n^2 < Q$. RSA parašo schemoje bus naudojamas modulis $N = PQ$. Tai reiškia, kad jei p ir q yra 2^{1024} eilės, tai P ir Q turi būti 2^{2048} eilės.

2.4. Modelio protokolai

2.4.1. Registracijos protokolas

Registracijos protokolas kiekvienam vartotojui (V) vykdomas vieną kartą ir įprastai tuomet, kai vartotojas atsidaro sąskaitą, naudojant saugią ir autentifikuotą komunikaciją tarp jo ir banko (B). Registracijos protokolo metu vartotojas iš banko gauna elektroninę licenciją.

Naudojami simboliai:

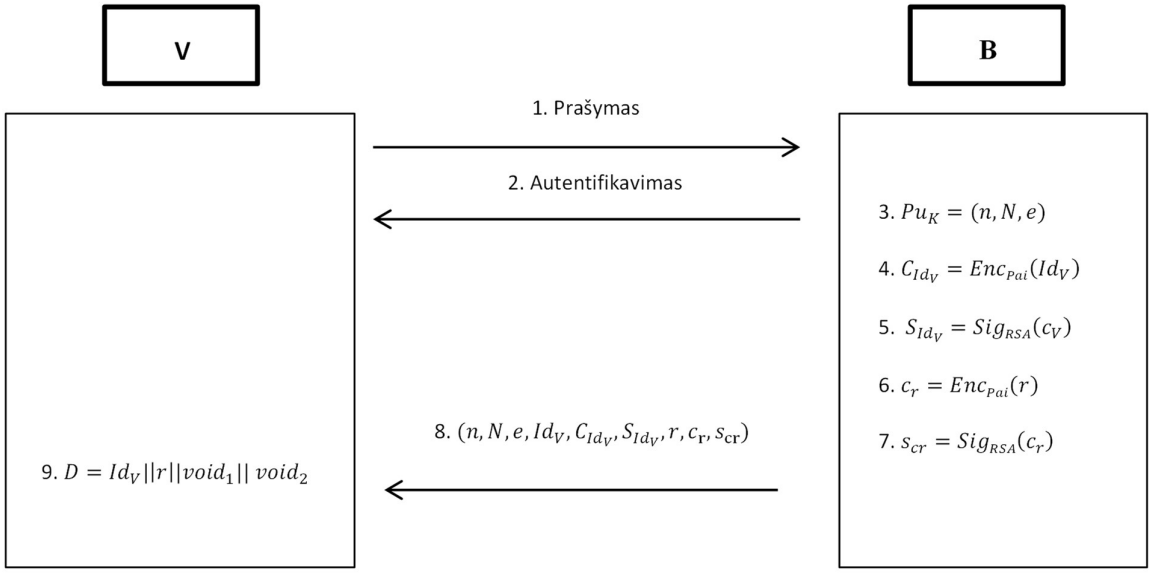
Id_V – identifikacinis vartotojo (V) numeris;

Pu_K – banko viešasis raktas;

Enc_{Pai} – viešojo rakto šifravimo funkcija, paremta Paillier'io schema;

Sig_{RSA} – e. parašo funkcija, paremta RSA algoritmu;

r – atsitiktinis pirminis banko sugeneruotas skaičius.



2.2 pav. Vartotojo registracijos protokolas

Protokolo procesas:

1. **V** kreipiasi į **B**, norėdamas atsidaryti elektroninių pinigų sąskaitą ir susikurti e. piniginę;
2. **B** autentifikuoja **V;**

3. jei **V** savo banko sąskaitoje turi depozito, tuomet **B** atsiunčia jam savo viešuosius parametrus $Pu_K = (n, N, e)$;

4-5. **B** priskiria **V** identifikacijos numerį Id_V, jį užšifruoja naudodamas Paillier'io funkciją ir uždeda RSA parašą;

6-7. **B** sugeneruoja atsitiktinį skaičių r, jį užšifruoja naudodamas Paillier'io funkciją ir uždeda RSA parašą;

8. **B** saugiu šifruotu kanalu nusiunčia **V** parametrus $(n, N, e, Id_V, C_{Id_V}, S_{Id_V}, r, c_{\mathrm{r}}, s_{\mathrm{cr}})$;
9. **V** iš gautų duomenų suformuoja e. piniginės duomenų struktūrą D, susidedančią iš: $D = Id_V||r||void_1||\ void_2$. Dydis r atsitiktinai generuojamas kiekvienai transakcijai. $void_1$ – tai tuščia pozicija, skirta didžiausiai sumai M, kurią **B** leidžia išleisti **V**. $void_2$ – tuščia pozicija, skirta dešimtainiam skaičiui, reprezentuojančiam e. monetų kiekiui, pervedamam mokėjimo protokolo metu. Struktūra D apibrėžia pirminį e. piniginės turinį sujungus su iš **B** gautais duomenimis.

2.4.2. Elektroninių pinigų formatas

Kuriamoje mokėjimo sistemoje bus naudojamas toks e. pinigų formatas:

2.1 lentelė. E. pinigų formatas

i	1	2	5	10	20	50	100	200	500
m_i	m_{001}	m_{002}	m_{005}	m_{010}	m_{020}	m_{050}	m_{100}	m_{200}	m_{500}
c_i	c_{001}	c_{002}	c_{005}	c_{010}	c_{020}	c_{050}	c_{100}	c_{200}	c_{500}
s_i	s_{001}	s_{002}	s_{005}	s_{010}	s_{020}	s_{050}	s_{100}	s_{200}	s_{500}

m_i – yra e. monetos nominalas (1 – lygu 1 centui, 2 – 2 centams,…. 500 – 5€);

c_i – yra m_i šifrograma, gauta panaudojus Paillier'io šifravimo schemą;

s_i – banko RSA parašas, uždėtas ant šifrogramos c_i.

Tinkamam e. piniginės duomenų struktūros $D = (Id_p\ ||\ \alpha \cdot r\ ||\ M\ ||\ M_1)$ suformavimui, kiekvienam dydžiui pranešime reikia priskirti poziciją, t. y. atitinkamą jai skirtų dešimtainių skaičių kiekį. Pranešimo duomenų pozicijos parodytos lentelėje žemiau.

2.2 lentelė. Pranešimo duomenų pozicijos

	Pozicija pranešime	**Daugiklis**
$\boldsymbol{Id_V}$	11	10^{53}
$\boldsymbol{\alpha \cdot r}$ **arba** $\boldsymbol{r}$	18	10^{42}
$\boldsymbol{M_{dep}}$	11	10^{12}
$\boldsymbol{M_1}$	11	1

Atsitiktiniams dydžiams r ir α skiriame 9 dešimtainius skaičius, o jų sandaugai 18 pozicijų. Pavyzdžiui, jei modeliuojame pranešimą: $Id_V + \alpha \cdot r + M_{dep} + M_1$, tuomet:

$$38905319999 \cdot 10^{53} + 123456789012345678 \cdot 10^{42} + 100000001 \cdot 10^{12} + 95620 \cdot 1$$
$$= 3890531999912345678901234567800100000001000000095620$$

2.4.3. Išėmimo protokolas

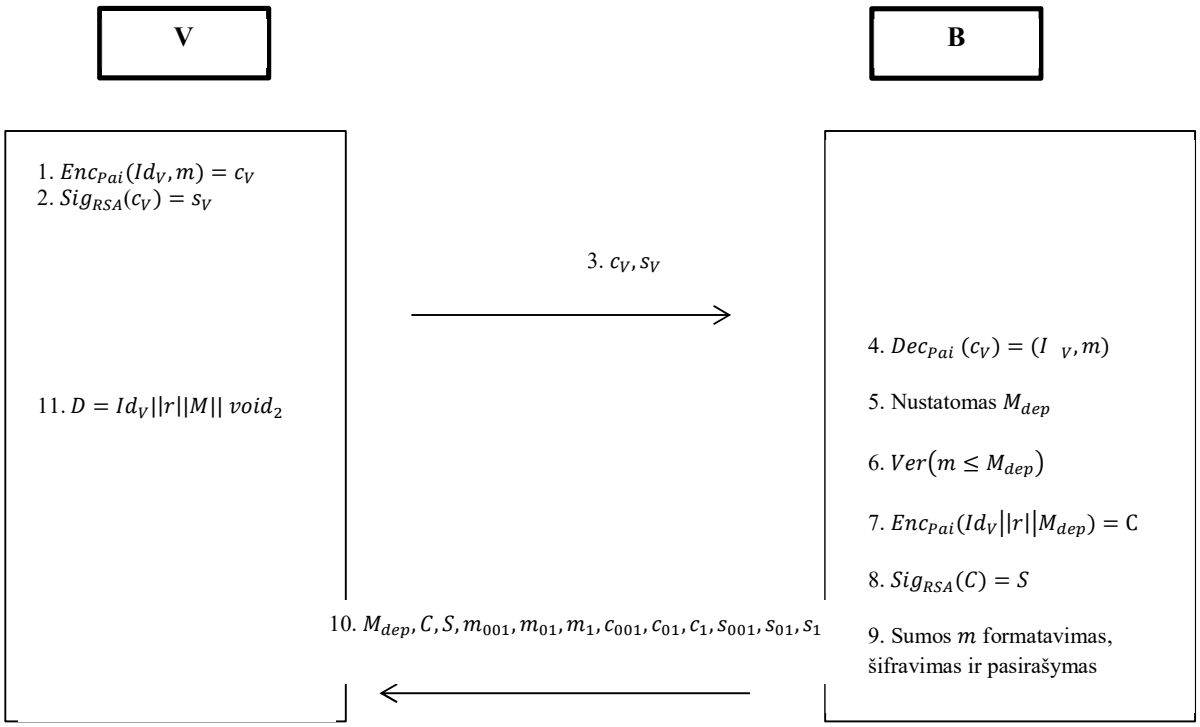
Šioje skyriaus dalyje detaliai aprašytas e. monetos išėmimo protokolas, t. y. kaip vartotojas $\boldsymbol{V}$ gali papildyti savo mobiliąją e. piniginę elektroniniais pinigais. Schemoje naudojami simboliai ir jų reikšmės:

Dec_{Pai} – viešojo rakto iššifravimo funkcija, paremta Paillier'io schema;

m – vartotojo norima gauti pinigų suma;

Ver – patikrinimo funkcija;

M_{dep} – vartotojo turimas balansas banko sąskaitoje.



2.3 pav. E. pinigo išėmimo protokolas

Išėmimo protokolo procesas:

1-3. **V** nori savo e. piniginę papildyti suma m. Užšifravęs savo Id_V ir sumą m ir ant šifrogramos c_V uždėjęs parašą s_V, siunčia **B**;

4-6. **B** iššifruoja **V** atsiųstus duomenis. Matydamas **V** Id, **B** jį autentifikuoja, patikrinęs turimą balansą banko sąskaitoje nustato maksimalią leistiną išleisti sumą M_{dep} ir patikrina ar prašoma suma nėra už ją didesnė;

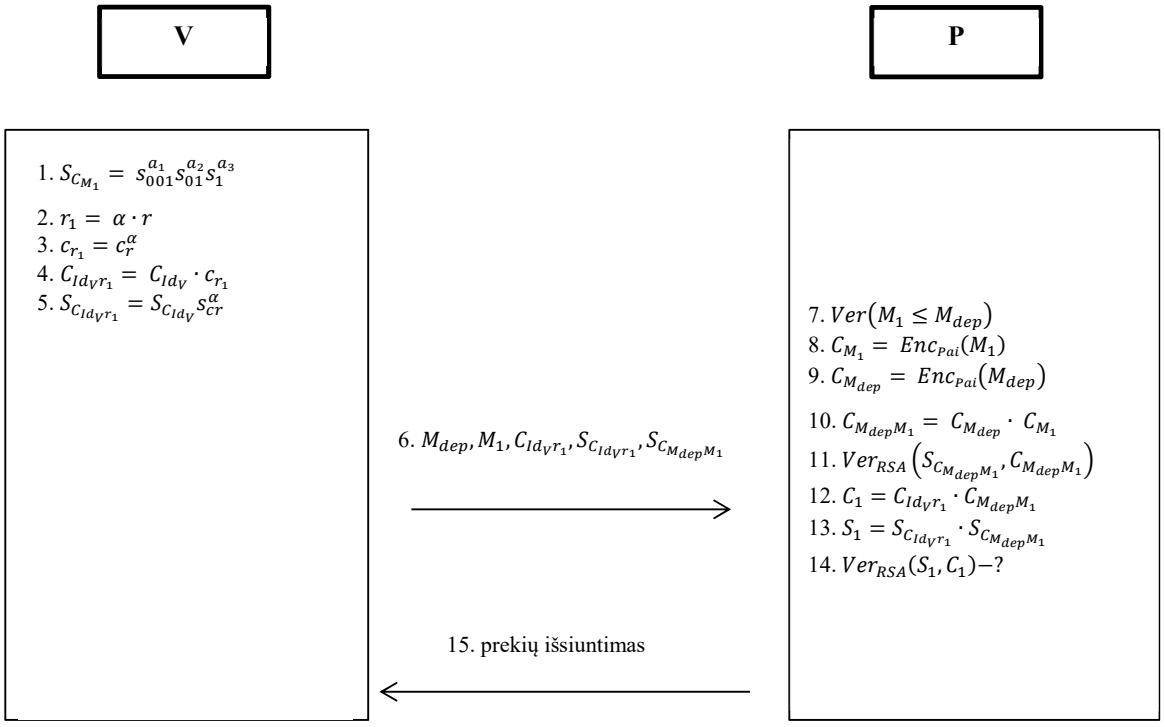
7-9. **B** sugeneruoja atsitiktinį dydį r ir apskaičiuoja $(Id_V||r||M_{dep})$ šifrogramą, tada ją pasirašo. Toliau suformatuoja sumą m, naudodamas skirtingus nominalus. Pavyzdžiui, $m_{001} = 0{,}01€$, $m_{01} = 0{,}1€$, $m_1 = 1€$. **B** užšifruoja juos Paillier'io funkcija, gaudamas $c_{001} = Enc_{Pai}(m_{001})$, $c_{01} = Enc_{Pai}(m_{01})$, $c_1 = Enc_{Pai}(m_1)$. Pinigų integralumui užtikrinti, ant gautų šifrogramų dedamas RSA parašas: $s_{001} = Sig_{RSA}(c_{001})$, $s_{01} = Sig_{RSA}(c_{01})$, $s_1 = Sig_{RSA}(c_1)$;

10. **B** nusiunčia **V** gautus parametrus $(M_{dep}, C, S, m_{001}, m_{01}, m_1, c_{001}, c_{01}, c_1, s_{001}, s_{01}, s_1)$;

7-10. **V** papildo e. piniginę gautais duomenimis: $D = Id_V||r||M_{dep}||\, void_2$. Iš **B** gautus užšifruotus duomenis **V** naudos kartu su dydžiais Id_V, r ir M_{dep}.

2.4.4. Mokėjimo protokolas

Tarkime, kad vartotojas **V** nori pervesti sumą $M_1 \leq M_{dep}$ pardavėjui **P**. **V**, naudodamasis iš **B** gautais e. banknotais m_{001}, m_{01}, m_1 suformuoja reikiamą sumą: $M_1 = a_1 m_{001} + a_2 m_{01} + a_3 m_1$, kur a_1, a_2, a_3 yra e. banknotų kiekis.



2.4 pav. E. monetos mokėjimo protokolas

Mokėjimo protokolo procesas:

1. **V** sudauginęs iš banko gautus parašus s_{001}, s_{01} ir s_1, apskaičiuoja parašą $S_{C_{M_1}}$;

2-3. **V** šiam mokėjimui sugeneruoja atsitiktinį skaičių α ir apskaičiuoja r_1 bei c_{r_1};

4-5. **V** apskaičiuoja šifrogramą ir parašą ant $Id_V + \alpha r$, t. y. dydžius $C_{Id_V r_1}$ ir $S_{C_{Id_V r_1}}$;

6. **V** siunčia **P** tokius duomenis: $M_{dep}, M_1, C_{Id_V r_1}, S_{C_{Id_V r_1}}, S_{C_{M_{dep} M_1}}$, kur pagal Paillier'io schemoje galiojančią savybę: $S_{C_{M_{dep} M_1}} = Sig_{RSA}\left(Enc_{Pai}(M_{dep} + M_1)\right) = Sig_{RSA}\left(C_{M_{dep}}\right) \cdot Sig_{RSA}(C_{M_1}) = S_{C_{M_{dep}}} \cdot S_{C_{M_1}}$;

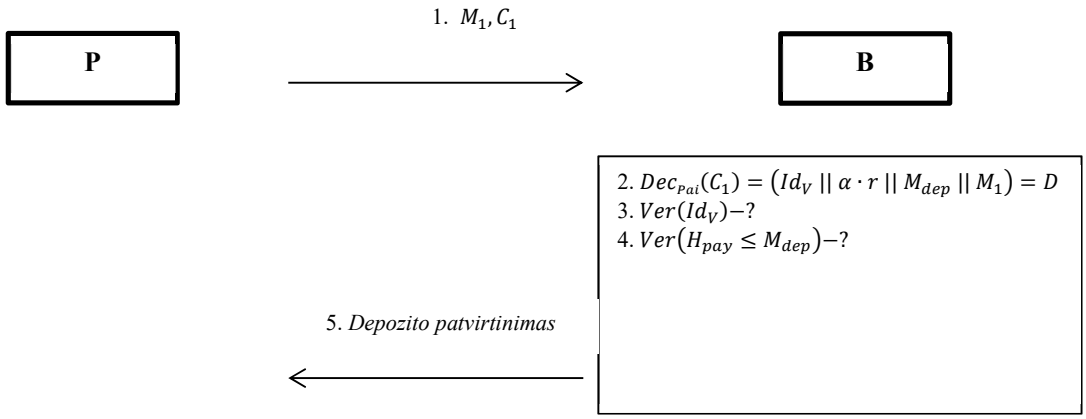
7-10. **P** patikrina ar $M_1 < M_{dep}$ ir jei taip, su Paillier'io šifravimo funkcija apskaičiuoja šifrogramas C_{M_1}ir $C_{M_{dep}}$, taip pat ir $C_{M_{dep} M_1}$;

11-13. **P** patikrina parašą $S_{C_{M_{dep} M_1}}$ant $C_{M_{dep} M_1}$. Tegul $Ver_{RSA}\left(S_{C_{M_{dep} M_1}}, C_{M_{dep} M_1}\right) = True$. Tada **P** apskaičiuoja šifrogramą C_1 ir parašą S_1 ant jos;

14-15. Paskutiniame etape **P** patikrina S_1 ant C_1 ir jei $Ver_{RSA}(S_1, C_1) = True$, tada **P** priima e. monetą M_1 ir išsiunčia **V** prekes.

2.4.5. Depozito protokolas

Po mokėjimo protokolo įvykdymo, pardavėjas **P** siunčia iš vartotojo **V** gautą sumą bankui **B**, kad atliktų depozitą. Depozitas visada vyksta prijungtiems režimu. Tai svarbu todėl, kad būtent šio protokolo metu **B** tikrina e. monetos galiojimą ir kitus saugumo parametrus.



2.5 pav. E. monetos depozito protokolas

Depozito protokolo procesas:

1. **P** siunčia B iš V gautus duomenis – pervedamą sumą M_1 ir jos šifrogramą C_1;
2. **B** iššifruoja šifrogramą $Dec_{Pai}(C_1) = (Id_V \,||\, \alpha \cdot r \,||\, M_{dep} \,||\, M_1) = D$;
3. **B** pirmiausia patikrina **V** statusą pagal gautą Id_V. Šioje protokolo stadijoje **B** pagal turimą Id_V gali atsekti visą **V** mokėjimo istoriją. Jei bendra atliktų mokėjimų suma H_{pay} yra didesnė nei **V** turimas depozitas banke M_{dep}, tuomet fiksuojama permoka.

2.4. Sistemos saugumo vertinimas

Sistemos saugumas paremtas Paillier‘io šifravimo schemos bei RSA parašo algoritmo saugumu. Saugumą padidina tai, kad šiuo atveju, RSA parašas dedamas ne ant atviro teksto, bet ant jau šifruotų duomenų. Be to, RSA algoritmas remiantis apytiksliais įverčiais bei bandymų rezultatais laikomas saugiu.

2.5.1. Anonimiškumas pardavėjo atžvilgiu

Vykdant mokėjimo protokolą, pirkėjas (P) keičia savo Id_V pridėdamas sandaugą $\alpha \cdot r$, kur α yra vartotojo V parinktas atsitiktinis skaičius, o r – iš banko (B) gautas atsitiktinis skaičius. Vartotojas (V) maskuoja savo Id_V kiekvieno atsiskaitymo metu pasirinkdamas skirtingą α.

2.5.2. Permokos prevencija

Kadangi kuriama mokėjimo sistema yra pilnai kontroliuojama banko, jis atsakingas ir už apsaugą nuo permokėjimo. Ši apsauga realizuojama depozito protokolo metu. Bankas (B) iššifravęs mokėjimo duomenis (D), gauna pirkėjo (P) identifikacinį numerį ($\mathrm{Id_V}$), pagal kurį gali atsekti visus prieš tai jo įvykdytus mokėjimus, taigi bandymas įvykdyti permoką bus iš karto aptiktas depozito metu.

2.5.3. E. monetos galiojimo užtikrinimas naudojant RSA parašą

Yra įrodyta, jog RSA parašas, uždėtas ant atviro teksto, gali būti pažeidžiamas įvairių atakų. Kuriamame modelyje parašas dedamas ant šifrogramos, gautos pritaikius Paillier‘io šifravimo algoritmą. Toks RSA parašo taikymas yra ekvivalentus maišos RSA, kuris laikomas saugiu. Šiuo atveju, maišos funkciją atitinka Paillier‘io šifravimo funkcija, todėl RSA parašą galima laikyti saugiu ir patikimu e. pinigų galiojimo užtikrinimu.

2.5. Tolimesnė darbo eiga

Tolesnės projektinės darbo dalies tikslas yra apskaičiuoti resursus, reikalingus kiekvienam mokėjimo protokolui atlikti ir įvertinti tokios sistemos efektyvumą, lyginant su kitomis e. pinigų schemomis.

3. ELEKTRONINIŲ PINIGŲ SISTEMOS EFEKTYVUMO TYRIMAS

3.1. Kompiuterio aritmetikos algoritmai

Elektroninių pinigų sistemos modelis kuriamas ribotus išteklius turintiems įrenginiams, tokiems kaip mobilieji telefonai ar planšetės. Siekiant įvertinti siūlomos sistemos efektyvumą ir tinkamumą jos realizavimui, tikslinga bent teoriškai įvertinti mokėjimo protokolams reikalingus skaičiavimo resursus.

Įrenginio procesoriaus dažnis f parodo, kiek taktų per 1 sekundę jis pajėgus įvykdyti. Priklausomai nuo procesoriaus specifikacijų, kiekvieno takto metu jis gali įvykdyti vieną ar daugiau instrukcijų. Kriptografinių funkcijų ir elektroninių pinigų sistemos protokolų efektyvumui įvertinti, t. y. taktų kiekiui apskaičiuoti, bus naudojami Donaldo Knuto kompiuterio aritmetikos algoritmai [11].

Darome prielaidą, kad elektroninių pinigų sistema planuojama realizuoti įrenginyje, turinčiame 1.6 GHz taktinį dažnį. Vieno takto atlikimo laikas yra:

$$T = \frac{1}{f} = \frac{1}{1.6\ GHz} = \frac{10^{12}}{1.6 \cdot 10^{9}} = 625\ ps = 625 \times 10^{-9}\ sec. \qquad (3.1)$$

Kompiuterio aritmetikos algoritmai:

$$M(w) = 3M(w/2) + 5A(w) + 2S; \qquad (3.2)$$

$$A(w) = w/32; \qquad (3.3)$$

$$\mathrm{Mod}(w) = \mathrm{Mod}(w/2) + 4M(w/2) + 1{,}5A(w) + 3S; \qquad (3.4)$$

$$\mathrm{MOD}_E(y, z) = 1{,}5 \cdot l(y)[M(l(z)) + 2\mathrm{Mod}(l(z)) + 1]; \qquad (3.5)$$

čia:

- $\mathbf{MOD}_{\boldsymbol{E}}(\boldsymbol{y}, \boldsymbol{z})$ - kėlimo laipsniu modulinė operacija ($x^y mod\ z$);
- $\boldsymbol{M(w), A(w), Mod(w)}$ – daugybos, sudėties ir modulio operacijos, kur w yra operando ilgis bitais;
- $\boldsymbol{l(w)}$ – dydžio w ilgis bitais;
- $\boldsymbol{S}$ – postūmio operatorius.

Naudosime prielaidą, kad operacijos $M(32)$, $A(32)$, $\mathrm{Mod}(32)$ ir S užims vieną procesoriaus taktą.

3.2. Kriptografinių funkcijų efektyvumo įvertinimas

- **Paillier'io užšifravimas**

Paillier'io sistemoje pranešimo m šifravimas vyksta naudojant dvi kėlimo laipsniu operacijas moduliu n^2: $c = g^m \cdot r^n \mathrm{mod}\ n^2$.

Bendras procesoriaus taktų skaičius, reikalingas šifrogramai c gauti priklauso nuo pranešimo m ilgio. Parametrą g rekomenduojama rinktis lygų $n + 1$, tačiau įvertinę, kad naudosime saugų 2048

bitų parametrą n ir siekiant paspartinti g^m apskaičiavimą, renkamės $g = 2$. Net ir priskyrę tokią mažą reikšmę, mes vis dar tenkiname reikalavimą, jog $g \in Z_{n^2}^*$. Skaičiavimą skaidome į dvi dalis: $r^n \text{mod}\ n^2$ ir $g^m \text{mod}\ n^2$.

Apskaičiuojame, kiek reikės procesoriaus taktų daugybos operacijai atlikti:

$$M\big(l(n^2)\big) = M(4096) = 3M(2048) + 5A(4096) + 2S = 3M(2048) + 642$$
$$= 3[3M(1024) + 5A(2048) + 2S] + 642 = \cdots = 24963.$$

Apskaičiuojame, kiek reikės modulio operacijų:

$$\text{Mod}\big(l(n^2)\big) = \text{Mod}(4096) = \text{Mod}(2048) + 4M(2048) + 1{,}5A(4096) + 3S$$
$$= [\text{Mod}(1024) + 4M(1024) + 1{,}5A(2048) + 3S] + 32623 = \cdots = 47759.$$

Bendras taktų skaičius, reikalingas apskaičiuoti $r^n \text{mod}\ n^2$:

$$\text{MOD}_{\text{E}}(n, n^2) = 1{,}5 \cdot l(n)\big[M\big(l(n^2)\big) + 2\text{Mod}\big(l(n^2)\big) + 1\big] = 1{,}5 \cdot 2048[M(4096) +$$
$$2\text{Mod}(4096) + 1] = 370120704.$$

Tarkime, kad pranešimas m yra 32 bitų ilgio. Vadinasi, taktų skaičius, norint gauti $g^m \text{mod}\ n^2$ yra:

$$\text{MOD}_{E}(m, n^2) = 1{,}5 \cdot l(m)\big[M\big(l(n^2)\big) + 2Mod\big(l(n^2)\big) + 1\big] = 1{,}5 \cdot 32[M(4096) +$$
$$2\text{Mod}(4096) + 1] = 5783136.$$

Bendras užšifravimui reikalingas taktų skaičius yra $375903840 = 376{\times}10^6$.

- **Paillier‘io iššifravimas**

Paillier‘io iššifravimo funkcijai $m = \frac{c^{\lambda} mod n^2 - 1}{n}$ atlikti reikalinga tik viena kėlimo laipsniu moduliu n^2 operacija. Privačiojo rakto parametras λ yra 2048 bitų ilgio.

$$\text{MOD}_{\text{E}}(\lambda, n^2) = 1.5 \cdot l(\lambda)\big[M\big(l(n^2)\big) + 2\text{Mod}\big(l(n^2)\big) + 1\big]$$
$$= 1.5 \cdot 2048[M(4096) + 2\text{Mod}(4096) + 1] = 421533696$$

Pranešimo iššifravimui reikia $422{\times}10^6$ taktų. Analogišku būdu tirsime ir RSA skaičiavimo efektyvumą.

- **RSA parašo generavimas**

Siekiant išvengti atviro teksto atakų (angl. *plaintext attack*), siūlomoje e. pinigų sistemoje, RSA parašas s dedamas ant Paillier‘io algoritmu užšifruoto pranešimo: $s = c^d \text{mod}(N)$.

Parašo uždėjimui reikalinga viena kėlimo laipsniu operacija moduliu N. Parametrai d ir N yra 2048 bitų ilgio.

$M\big(l(N)\big) = M(4096) = 24963.$

$\mathrm{Mod}\big(l(N)\big) = \mathrm{Mod}(4096) = 47759.$

$\mathrm{MOD_E}(d, N) = 1.5 \cdot l(d)\big[M\big(l(N)\big) + 2\mathrm{Mod}\big(l(N)\big) + 1\big] = 1.5 \cdot 2048[M(4096) + 2\mathrm{Mod}(4096) + 1] = 370120704.$

RSA parašui uždėti reikia $370{\times}10^6$ taktų.

- **RSA parašo tikrinimas**

Parašo tikrinimui apskaičiuojamas $c' = s^e mod(N)$. Šiam apskaičiavimui taip pat reikalinga tik viena kėlimo laipsniu operacija moduliu N.

$$\mathrm{MOD}_E(e, N) = 1.5 \cdot l(e)\big[M\big(l(N)\big) + 2\mathrm{Mod}\big(l(N)\big) + 1\big] = 1.5 \cdot 40[M(4096) + 2\mathrm{Mod}(4096) + 1] = 7228920 = 7{,}2{\times}10^6$$

- **Rezultatai**

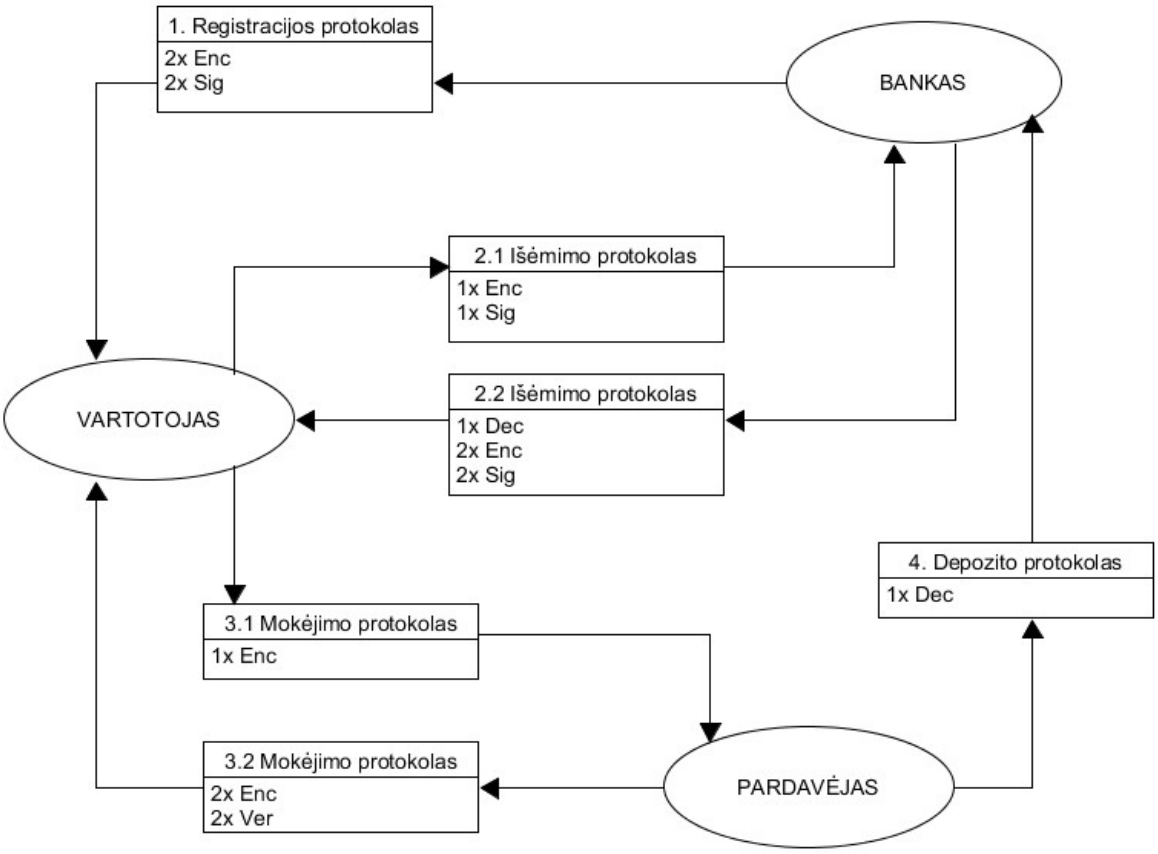
3.1 lent. pateiktas kiekvienos kriptografinės funkcijos vykdymo laikas. Paillier‘io duomenų šifravimo algoritmas užtrunka ilgiausiai (0,46 s), nes jam įvykdyti reikalingos dvi modulinės eksponentės operacijos. Trumpiausiai užtrunka RSA parašo tikrinimas (0,005 s).

3.1 lentelė. Kriptografinių funkcijų efektyvumo rezultatai

	Procesoriaus taktų skaičius	**Įvykdymo laikas, sekundėmis**
Paillier‘io šifravimas	$376{\times}10^6$	0,463
Paillier‘io iššifravimas	$422{\times}10^6$	0,264
RSA parašas	$370{\times}10^6$	0,231
RSA parašo tikrinimas	$7{,}2{\times}10^6$	0,005

3.3. Protokolų vykdymo laiko įvertinimas

Sistemoje dirbama su saugiais parametrais, todėl naudojamos kriptografinės funkcijos labiausiai įtakoja protokolų vykdymo laiką. Žemiau esančioje schemoje pateikiama, kiek ir kokių funkcijų reikia atlikti kiekviename sistemos žingsnyje.



3.1 pav. Kontekstinė modelio schema

Protokolų vykdymo laikas skaičiuojamas darant prielaidą, kad parametrai yra tokio dydžio:

- pranešimas m – 32 bitai;
- dydis a (e. banknotų kiekis) – 8 bitai;
- eksponentė e – 40 bitų;
- modulis N – 4096 bitai;
- visi kiti dydžiai – 2048 bitai.

Žemiau esančiose lentelėse pateiktas kiekvienam protokolo žingsniui, detaliai aprašytam metodologinėje projekto dalyje, atlikti reikalingas laikas.

3.2 lentelė. Registracijos protokolo vykdymo laikas

Protokolo žingsnis	Procesoriaus taktų skaičius	Atlikimo laikas, s
$C_{Id_V} = Enc_{Pai}(Id_V)$	740241408	0,462651
$S_{Id_V} = Sig_{RSA}(c_V)$	370120704	0,231325
$c_r = Enc_{Pai}(r)$	740241408	0,462651

$s_{cr} = Sig_{RSA}(c_r)$	370120704	0,231325
VISO	**2220724224**	**1,387953**

3.3 lentelė. E. pinigo išėmimo protokolo vykdymo laikas

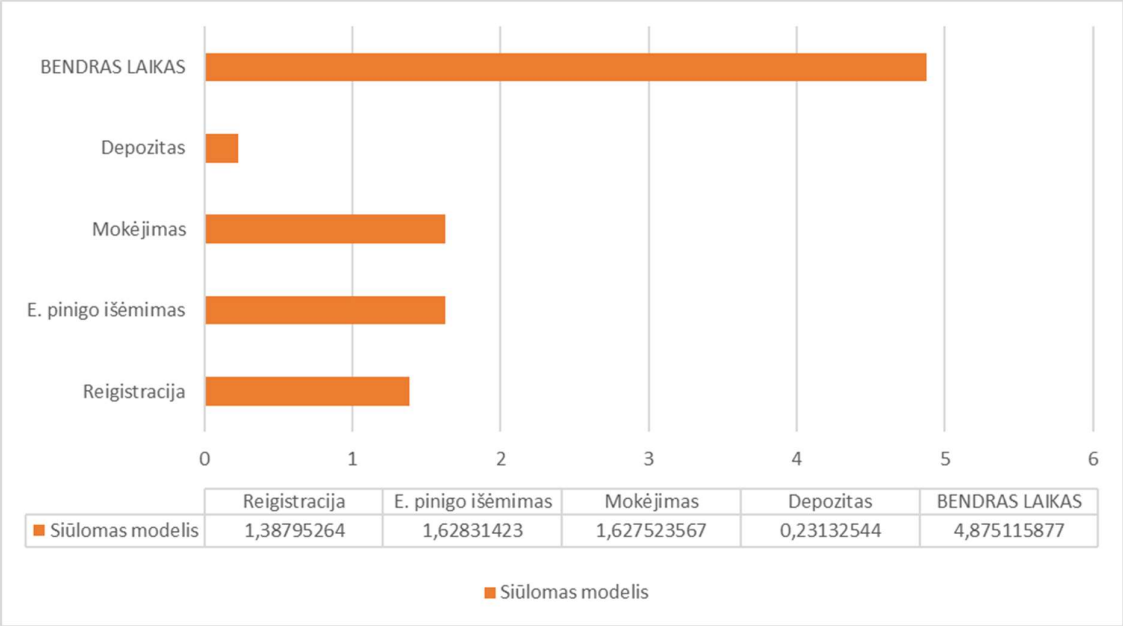
Protokolo žingsnis	Procesoriaus taktų skaičius	Atlikimo laikas, s
$Enc_{Pai}(Id_V, m) = c_v$	747470328	0,467169
$Sig_{RSA}(c_V) = s_v$	370120704	0,231325
$Dec_{Pai}(c_V) = (Id_V, m)$	370120704	0,231325
$Sig_{RSA}(C) = S$	370120704	0,231325
Nominalų m šifravimas	377349624	0,235844
Nominalų m pasirašymas	370120704	0,231325
VISO	**2605302768**	**1,628314**

3.4 lentelė. Mokėjimo protokolo vykdymo laikas

Protokolo žingsnis	Procesoriaus taktų skaičius	Atlikimo laikas, s
$S_{C_{M_1}} = s_{001}^{a_1} s_{01}^{a_2} s_1^{a_3}$	4337352	0,002711
$c_{r_1} = c_r^{\alpha}$	370120704	0,231325
$C_{Id_V r_1} = C_{Id_V} \cdot c_{r_1}$	361446	0,000226
$S_{C_{Id_V r_1}} = S_{C_{Id_V}} \cdot s_{cr}^{\alpha}$	370301427	0,231438
$C_{M_1} = Enc_{Pai}(M_1)$	740241408	0,462651
$C_{M_{dep}} = Enc_{Pai}(M_{dep})$	740241408	0,462651
$C_{M_{dep}M_1} = C_{M_{dep}} \cdot C_{M_1}$	361446	0,000226
$Ver_{RSA}\left(S_{C_{M_{dep}M_1}}, C_{M_{dep}M_1}\right)$	7228920	0,004518
$C_1 = C_{Id_V r_1} \cdot C_{M_{dep}M_1}$	361446	0,000226
$S_1 = S_{C_{Id_V r_1}} \cdot S_{C_{M_{dep}M_1}}$	361446	0,000226
$Ver_{RSA}(S_1, C_1)$	370120704	0,231325
VISO	**3344279115**	**1,627523**

3.5 lentelė. Depozito protokolo vykdymo laikas

Protokolo žingsnis	Procesoriaus taktų skaičius	Atlikimo laikas, s
$Dec_{Pai}(C_1) = (Id_V \| \| \alpha \cdot r \| \| M_{Dep} \| \| M_1) = D$	370120704	0,231325
VISO	**370120704**	**0,231325**



3.2 pav. Siūlomo modelio protokolų vykdymo laikas

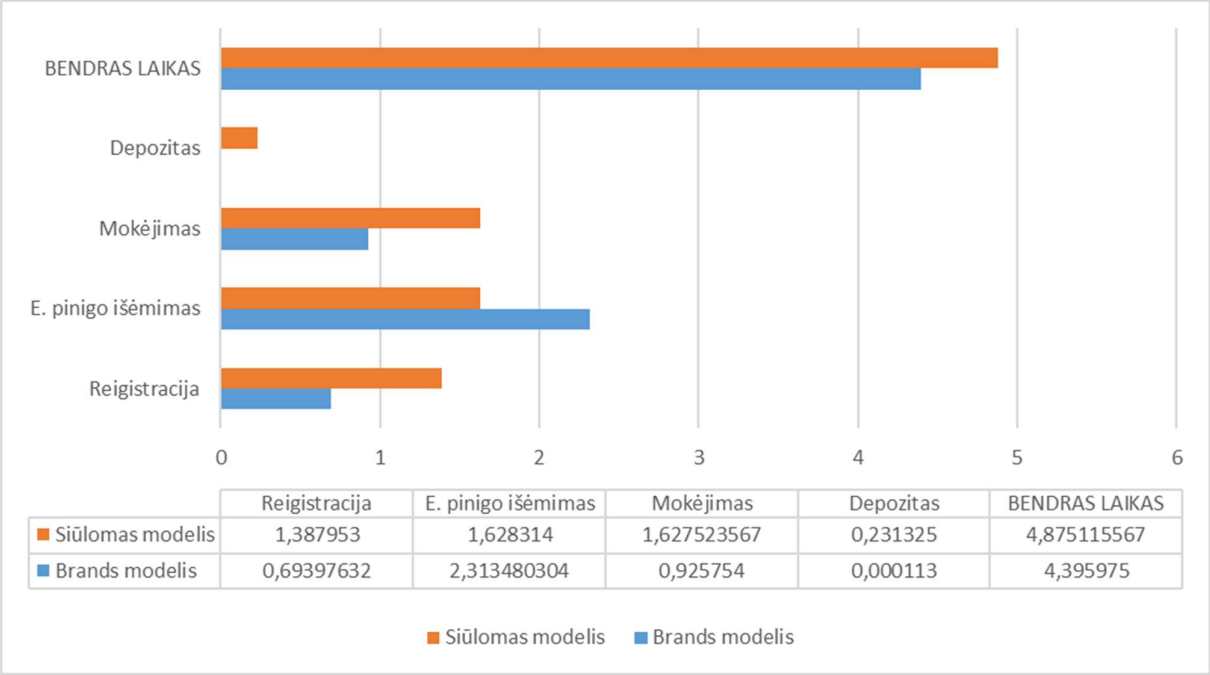
Siūlomoje sistemoje vienas pilnas mokėjimo ciklas užima 4.88 s. Dėl panašaus kriptografinių funkcijų skaičiaus, mokėjimo ir e. pinigo išėmimo protokolai užima tiek pat laiko (po 1,63 s). Depozito protokolas trunka trumpiausiai (0,23 s), nes reikalauja vos vienos iššifravimo funkcijos.

3.4. Palyginimas su Brandso modeliu

Iš analizės dalyje aptartų ir literatūroje dažniausiai sutinkamų modelių – Chaumo, Fergusono, Okomoto ir Brandso - pastarasis daugelio vertintojų laikomas geriausiu, todėl tikslinga siūlomą sistemą palyginti būtent su juo.

P. Palevičius savo magistro projekte detaliai aprašė visus Brandso sistemoje naudojamus protokolus [7 p. 19-22]. Šiame tyrime Brandso sistemos iniciavimo ir pirkėjo sąskaitos atidarymo protokolai bus laikomi atitikmeniu kuriamos sistemos registracijos protokolui. Vertinimas atliekamas analogiškai naudojantis D. Knuto metodika, parenkant tokio pat dydžio parametrus, kaip ir šiame darbe pasiūlytoje elektroninių pinigų sistemoje.

Mokėjimo ciklo laiką labiausiai įtakoja modulinės eksponentės operacijos. Brandso sistemoje, registracijos protokole tokių operacijų yra 3, e. pinigo išėmimo – 12, mokėjimo – 6, o depozito protokole nei vienos. Palyginimo rezultatai pateikiami 3.3 pav.



	Reigistracija	E. pinigo išėmimas	Mokėjimas	Depozitas	BENDRAS LAIKAS
Siūlomas modelis	1,387953	1,628314	1,627523567	0,231325	4,875115567
Brands modelis	0,69397632	2,313480304	0,925754	0,000113	4,395975

3.3 pav. Palyginimas su Brandso modeliu

Brandso sistemoje visų protokolų vykdymo laikas nesiekia 1 s, išskyrus e. pinigo išėmimo, kuris trunka ilgiausiai – 2,31 s. Nors kuriamame modelyje beveik visi protokolai vyksta šiek tiek ilgiau, bendras mokėjimo ciklo laikas yra nepilnai 0,5 s ilgesnis nei Brandso.

4. EKSPERIMENTINIO MOKĖJIMO REALIZACIJA SU OCTAVE

Siūlomo elektroninių pinigų sistemos modelio realizavimui ir pinigų dalumo savybės išpildymo demonstravimui pasirinkta moksliniams skaičiavimams pritaikyta programavimo kalba *Octave*. Mokėjimo procese dalyvaujančių šalių komunikacija realizuota naudojant *Symfony 2.8* karkasą. Vartotojas su banku ir pardavėju bendrauja per interneto svetainę.

Banko naudojami Paillier'io ir RSA parametrai:

$PuK_{Pai} = (n_B = 35, n_{1B} = 36)$ $\quad PuK_{RSA} = (n_B = 12138031, e_B = 827)$

$PrK_{Pai} = (\Phi(n)_B = 24, \Phi(n)_B^{-1} = 19)$ $\quad PrK_{RSA} = (d_B = 4694003).$

Realizacijoje naudojamas pinigų formatas:

i	1 eur	2 eur	5 eur
m_i	m_1	m_2	m_3
c_i	c_1	c_2	c_3
s_i	s_1	s_2	s_3

Siekiant pagreitinti kai kuriuos skaičiavimus, sukurtos papildomos *Octave* funkcijos:

- gen_RSA – automatiškai sugeneruojami RSA parametrai;
- genprime(x) – generuojamas x bitų ilgio pirminis skaičius;
- mod_exp(x,y,z) – apskaičiuojama $x^y \bmod z$;
- get_inv(x,y) – grąžina $x \bmod y$ atvirkšinį elementą z, kuris tenkina lygybę $z \cdot x = 1 \bmod y$.

Paillier'io šifravimas $c = (n+1)^m \cdot r^n \bmod n^2$. Octave bus atliekamas per 3 žingsnius:

1. $a = (n+1)^m \bmod n^2$;
2. $b = r^n \bmod n^2$;
3. $c = a \cdot b \; mod \; n^2$.

Pastaba. Dėl ribotų Octave skaičiavimo galimybių, eksperimentui pasirinkti mažesni parametrai nei siūloma metodologinėje darbo dalyje.

4.1. Registracijos protokolo vykdymas

V:	B:

1. Siunčiama registracijos užklausa ----------→	**2. Vartotojui priskiriamas $ID_V = 22$:**
	`>> ID=22`
	`ID = 22`

6. Vartotojui siunčiami duomenys

V ←----------

$(n_B, e, n_{1B}), ID_V, C_{ID}, S_{ID}, c_r, s_r, r_T)$

3. Sugeneruojamas atsitiktinis dydis r_T:

```
>> rT=gennumber(4)
rT = 5
```

4. Užšifruoja ir pasirašo vartotojo $ID_V = 22$ naudodamas savo parametrus.

```
>> nB=35
nB = 35
>> n1B=nB+1
nB1 = 36
```

4.1. Šifravimas:

```
>> r=gennumber(5)
r = 8
>> a=mod_exp(n1B,ID,nB*nB)
a = 771
>> b=mod_exp(r,nB,nB*nB)
b = 932
>> CID=mod(a*b,nB*nB)
CID = 722
```

4.2. Pasirašymas:

```
>> dB= 4694003
dB = 4694003
>> NB= 12138031
NB = 12138031
>> SID=mod_exp(cID,dB,NB)
SID = 1673591
```

5. Užšifruoja ir pasirašo sugeneruotą $r_T = 5$.
5.1. Šifravimas:

```
>> a=mod_exp(n1B,rT,nB*nB)
a = 176
>> b=mod_exp(r,nB,nB*nB)
b = 932
>> cr=mod(a*b,nB*nB)
cr = 1107
```

5.2. Pasirašymas:

```
>> sr=mod_exp(cr,dB,NB)
sr = 205375
```

4.2. Išėmimo protokolo vykdymas

Prielaida. Vartotojas savo banko sąskaitoje turi 20 eur. Į savo mobiliąją piniginę jis nori išsiimti 10 eur vertės e. monetą.

V:	B:

1. Užšifruoja savo ID_V , naudodamas gautus iš banko parametrus ir pasirašo savo RSA raktu.
 	
`>> dV=1913`
`dV = 1913`
`>> nV=33127`
`nV = 33127`
`>> r=gennumber(5)`
`r = 11`
 	
1.1. Šifravimas:
 	
`>> a=mod_exp(n1B,ID,nB*nB)`
`a = 771`
`>> b=mod_exp(r,nB,nB*nB)`
`b = 226`
`>> cID=mod(a*b,nB*nB)`
`cID = 296`
 	
1.2. Pasirašymas:
 	
`>> sr=mod_exp(cID,dV,nV)`
`sr = 11055269`
 	
2. Užšifruoja norimą išsiimti sumą $m = 10$ ir pasirašo savo RSA raktu.
 	
`>> m=10`
`m = 10`
 	
2.1. Šifravimas:
 	
`>> a=mod_exp(n1B,m,nB*nB)`
`a = 351`
`>> b=mod_exp(r,nB,nB*nB)`
`b = 226`
`>> cm=mod(a*b,nB*nB)`
`cm = 926`
 	
2.2. Pasirašymas:
 	
`>> sr=mod_exp(cm,dV,nV)`
`sr = 3497875`
 	
3. Bankui siunčiami duomenys: $c_{ID_V}, s_{ID_V}, c_m, s_m$
----------→
	4. Gautų duomenų iššifravimas.
	
	`>> Phi=24`
	`Phi = 24`
	`>> Phi_inv=19`
	`Phi_inv = 19`
	`>> nB=35`
	`nB = 35`
	
	4.1. $Dec_{Pail}(c_{ID_V})$:
	
	`>>L=((mod_exp(cID,Phi,nB*nB))-`
	`1)/nB`
	`L = 3`
	`>> mID=mod(L*Phi_inv,nB)`
	`mID = 22`
	
	4.2. $Dec_{Pail}(c_m)$:
	
	`>> L=((mod_exp(cm,Phi,nB*nB))-1)/nB`
	`L = 30`
	`>> mm=mod(L*Phi_inv,nB)`
	`mm = 10`
	
	5. Patikrina ar $(m \leq M_{dep})$. Jei ne, operacija atmetama. Jei taip, vykdomi sekantys žingsniai.
	
	`>> m5=5`
	`m5 = 5`
	`>> m2=2`
	`m2 = 2`
	`>> m1=1`
	`m1 = 1`
	`>> r=gennumber(5)`
	`r = 9`
	
	5.1. 5 eur e. monetos šifravimas:

	`>> a=mod_exp(n1B,m5,nB*nB)`
	`a = 176`
	`>> b=mod_exp(r,nB,nB*nB)`
	`b = 949`
	`>> c5=mod(a*b,nB*nB)`
	`c5 = 424`
	
	5.2. 2 eur e. monetos šifravimas:
	
	`>> a=mod_exp(n1B,m2,nB*nB)`
	`a = 71`
	`>> c2=mod(a*b,nB*nB)`
	`c2 = 4`
	
	5.3 1 eur e. monetos šifravimas:
	
	`>> a=mod_exp(n1B,m1,nB*nB)`
	`a = 36`
	`>> c1=mod(a*b,nB*nB)`
	`c1 = 1089`
	
	5.4. E. monetų pasirašymas
	
	`>> s5=mod_exp(c5,dB,NB)`
	`s5 = 11711302`
	`>> s2=mod_exp(c2,dB,NB)`
	`s2 = 11513548`
	`>> s1=mod_exp(c1,dB,NB)`
	`s1 = 7282809`

6. Vartotojui siunčiami duomenys: $M_{Dep}, m_5, m_2, m_1, c_5, c_2, c_1, s_5, s_2, s_1$ V ←----------	

4.3. Mokėjimo protokolo vykdymas

Prielaida. Vartotojas iš savo turimos 10 eur vertės e. monetos nori išleisti 6 eur ($m1 = 6$).

***V*:**	***P*:**

1. Naudojantis RSA homomorfizmo savybe, iš banko atsiųstų duomenų apskaičiuojamas parašas ant norimos vertės e. monetos, t. y. 6 eur:
 	
`>> NB=12138031 //Banko viešasis RSA parametras`
`NB = 12138031`
`>> s2 = 11513548// buvo gautas iš banko`
`S2 = 11513548`
`>> Sm1=mod_exp(s2,3,NB)`
`Sm1 = 8575089`
`>> Smdep=mod_exp(s2,5,NB)`
`Smdep = 3509888`
	6. Patikrina, ar $M_1 \leq M_{Dep}$. Jei ne, operacija atmetama. Jei taip, vykdomi sekantys žingsniai.
	
	`>> r=genprime(5)`
	`r = 9`

2. Apskaičiuoja naują transakcijos numerį r_1 ir jos šifrogramą $c_{r_1} = c_r^{\alpha}$:

```
>> alpha=gennumber(4)
alpha = 4
>>rT = 5
rT=5
>>r1 = alpha*rT
r1 = 20
```

3. Maskuoja savo ID su nauju transakcijos numeriu ir pasirašo: r_1, t. y. $C_{IDr_1} = Enc_{Pai}(ID||r_1)$:

```
>> x =strcat("22","20")
x = 2220
>> r = gennumber(5)
r = 11
>> a=mod_exp(n1B,x,nB*nB)
a = 526
>> b=mod_exp(r,nB,nB*nB)
b = 226
>> cIDr1=mod(a*b,nB*nB)
cIDr1 = 51
>> sIDr1=mod_exp(cMdep,dV,nV)
sIDr1 = 1725063
```

4. Apskaičiuojamas parašas ant $S_{M_{Dep}m_1}$, t. y. $S_{M_{dep}m_1} = S_{M_{Dep}} \cdot S_{m_1}$:

```
>> SMdepM1=mod(sMdep*Sm1,NB)
SMdepM1 = 6794091
```

5. Pardavėjui siunčiami duomenys:

$M_{Dep}, m_1, c_{IDr_1}, S_{IDr_1}, S_{M_{Dep}m_1}$

----------→

12. Vartotojui siunčiamos prekės.

V ←----------

7. Apskaičiuoja $C_{m_1} = Enc_{Pai}(m_1)$ naudodamas banko viešuosius Paillier'io parametrus:

```
>> m1=6
m1 = 6
>> a=mod_exp(n1B,m1,nB*nB)
a = 211
>> b=mod_exp(r,nB,nB*nB)
b = 949
>> cm1=mod(a*b,nB*nB)
cm1 = 564
```

8. Apskaičiuoja $C_{M_{Dep}} = Enc_{Pai}(M_{Dep})$:

```
>> Mdep=10
Mdep = 10
>> a=mod_exp(n1B,Mdep,nB*nB)
a = 351
>> b=mod_exp(r,nB,nB*nB)
b = 949
>> cMdep=mod(a*b,nB*nB)
cMdep = 1124
```

9. Apskaičiuoja $C_{M_{Dep}M_1} = C_{M_{Dep}} \cdot C_{M_1}$:

```
>> cMdepM1=mod(cm1*cMdep,nB*nB)
cMdepM1 = 611
```

10. $Ver(S_{M_{Dep}m_1}, C_{M_{Dep}M_1})$:

```
>> eB = 827
eB = 827
>> NB = 12138031
NB = 12138031
>> c_1=mod_exp(SMdepM1,eB,NB)
c_1=611
```

11. $Ver(S_{IDr_1}, c_{IDr_1})$:

```
>> eV = 139
eV = 139
>> nV = 12796697
NV = 12796697
>> c_1=mod_exp(sIDr1,eV,nV)
c_1= 51
```

4.4. Depozito protokolo vykdymas

$\boldsymbol{P}$ ----------M_1, c_{IDr_1}----------→
 	
2. Depozito patvirtinimas
←----------
	$\boldsymbol{B}$:
	
	1. $\boldsymbol{Dec_{Pai}(c_{IDr_1})}$:
	
	`>>`
	`L=((mod_exp(cIDr1,PhiB,nB*nB))-1)/nB`
	`L = 1314`
	`>> m=mod(L*PhiB_inv,nB)`
	`m = 2220`

5. IŠVADOS

1. Pasiūlyta homomorfinė mokėjimo sistema, kurioje realizuota e. pinigo dalumo savybė ir užtikrintas dalinis vartotojo anonimiškumas (pardavėjo atžvilgiu).
2. Sistemos saugumą užtikrina tai, kad visi siunčiami duomenys yra ne tik užšifruojami Paillier‘io algoritmu, kuris atlieka maišos funkciją, bet ir pasirašomi RSA parašu, turinčiu homomorfizmo savybę. Tokiu būdu pašalinamos homomorfinio RSA parašo saugumo spragos.
3. Sistemos realizacijos efektyvumas įvertintas teoriškai, panaudojant Knut'o metodiką. Gauti rezultatai rodo, kad Brandso modelyje mokėjimo ciklas įvykdomas 0,48 s greičiau nei kuriamoje sistemoje, tačiau jis nėra tinkamas realizacijai mobiliajame įrenginyje dėl e. pinigų duomenų kiekio augimo ir neišpildytos dalumo savybės.
4. Remiantis tyrimo rezultatais galima teigti, kad siūlomas modelis yra ne tik pažangesnis ir labiau tinkamas realizacijai mobiliojoje e. pinginėje nei Brandso modelis, bet ir pakankamai efektyvus vykdymo laiko prasme.
5. Sudarytas sistemos imitacinis modelis, kuris patvirtina sprendimų teisingumą.

6. LITERATŪRA

[1] B. Schoenmakers, „Basic Security of the ecash Payment System," *State of the Art in Applied Cryptography*, nr. 6, p. 338-352, 1997.

[2] H. Yu, K. Hsi, P. Kuo, „Electronic payment systems: an analysis and comparison of types," įtraukta *Management of Engineering and Technology,* Taivanas, 2001.

[3] M. Laurinaitis, „Elektroninių pinigų teisinis reguliavimas", *Daktaro disertacija*, Vilnius, 2015, p. 18-19.

[4] M. Jahanian Farsi, „Digital cash," *Magistro baigiamasis darbas*, Švedija, 1997, p. 23-24.

[5] B. Schneier, „Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth)," Willey, 1996, p. 159

[6] N. Ferguson, „Single Term Off-Line Coins," įtraukta *Advances in Cryptology EUROCRYPT '93*, Norvegija, 1994

[7] P. Palevičius, „Elektroninių pinigų modelio realizacija standartinėse ir ribotų aritmetinių funkcijų sistemose," *Magistro baigiamasis darbas,* Kaunas*,* 2011, p. 17.

[8] T. Okamoto, „An Efficient Divisible Electronic Cash Scheme", įtraukta *Advances in Cryptology CRYPTO '95*, Berlynas, 1995

[9] J. Katz ir Y. Lindell, Introduction to Modern Cryptography, USA, 2008, p. 411-416.

[10] E. Sakalauskas ir kt., Kriptografinės sistemos, KTU: e. knyga, 2012, p. 88

[11] R. Hwang ir F. Su, „An Efficient Decryption Method for RSA Cryptosystem," įtraukta *AINA'15*, Taivanas, 2015.

7. PRIEDAI

7.1. 1 Priedas. D. Knuto algoritmų realizacija Java programavimo kalba

```
package sistema;
public class Aritmetika {

static long temp,temp2;
	public static double A(int w){
			int sum=w/32;
			return sum;
	}
	public static double M(int w){
			int S=1;
			int w0=w;
			temp=0;
			int k=(int) (Math.log(w)/Math.log(2)-Math.log(32)/Math.log(2)+1);

			for (int i=1; i<=k; i++){
				int d=(int) Math.pow(3, i-1);
				if (i!=k){
				temp = (int) (temp+(5*A(w0)+2*S)*d);
				w0=w0/2;
				} else temp=temp+d;
			}
			return temp;
	}
	public static double Modw(int w){
			int S=1;
			int w0=w;
			int k=(int) (Math.log(w)/Math.log(2)-Math.log(32)/Math.log(2)+1);

			for (int i=1; i<=k; i++){
				if (i!=k){
				temp2 = (long) (temp2 + (4*M(w0/2)+1.5*A(w0)+3*S));
				w0=w0/2;
				}
			}
			return temp2+1;
	}
	public static double Modyz(int y, int z){

			temp = (long) (1.5 * y * (M(z) + 2 * Modw(z) + 1));
			return temp;
	}
	public static double op_time(double f){

			double temp = 1 / (f * Math.pow(10, 9));
			return temp;
	}
}
```