KAUNO TECHNOLOGIJOS UNIVERSITETAS

ALEKSEJUS MICHALKOVIČ

ASIMETRINIO ŠIFRAVIMO SISTEMOS, PAREMTOS NEKOMUTATYVIAIS KRIPTOGRAFIJOS METODAIS, SUKŪRIMAS IR SAVYBIŲ TYRIMAS

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07 T)

2015, Kaunas

Disertacija rengta 2010–2014 metais Kauno technologijos universitete, Matematikos ir gamtos mokslų fakultete, Taikomosios matematikos katedroje. Mokslinius tyrimus rėmė Lietuvos valstybinis mokslo ir studijų fondas.

Mokslinis vadovas:
Prof. dr. **Eligijus Sakalauskas** (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

TURINYS

SĄVOKOS IR SUTRUMPINIMAI

Agentas - programa, kuri gali aptarnauti sistemos klientus, t.y. leidžia klientams saugiai užšifruoti ir iššifruoti duomenis.

Algebrinė kriptoanalizė – kriptoanalizės metodas, kuriuo tikslas yra gauti informaciją apie vartotojo slaptąjį raktą sprendžiant algebrines lygtis, siejančias slaptąjį ir viešąjį raktus.

Asimetrinio šifravimo protokolas – matematinis algoritmas, kuris leidžia užšifruoti pranešimą ir iššifruoti šifrogramą naudojant matematiškai susietų raktų porą.

DLA – diskretinio logaritmo ataka.

DLU – diskretinio logaritmo uždavinys.

ECC – elipsinių kreivių asimetrinis šifravimas.

Jungtinumo lygtis – matricinė lygtis $X^{-1}AX = B$, kai matricos A ir B yra žinomos.

JPU – jungtinio elemento paieškos uždavinys.

Komutatyvumo lygtis – matricinė lygtis $AX = XB$, kai matricos A ir B yra žinomos.

Kriptografinis primityvas – rakto apsikeitimo, šifravimo arba elektroninio parašo protokolas.

Laipsninė matrica = matricinis laipsnis.

Laipsninis žiedas – kvadratinių matricų žiedas, kuriam priklauso MLF argumentai (matricos X ir Y).

Matricinis laipsnis – MLF argumentas (matrica X arba Y).

ML eksponentė – matricinio laipsnio funkcijos reikšmė.

MLAŠ – matricinio laipsnio asimetrinis šifravimas.

MLF – matricinio laipsnio funkcija.

MQ uždavinys – daugelio kintamųjų kvadratinių lygčių sistemos sprendimo uždavinys.

MMQ uždavinys – matricinės lygties $XAY = B$, kai A ir B – žinomos matricos, o X ir Y – nežinomos matricos, sprendimo uždavinys.

NP-pilnasis uždavinys – NP klasės uždavinys, į kurį galima transformuoti bet kokį kitą šios klasės uždavinį per polinominį laiką.

NP uždavinių klasė – sprendžiamumo uždavinių, kurių sprendinio teisingumą galima patikrinti per polinominį laiką, aibė.

Paieškos lentelė – lentelė, kurioje saugojamos visos galimos algebrinės operacijos reikšmės.

Platforminis žiedas – kvadratinių matricų žiedas, kuriam priklauso MLF pagrindas (matrica Q) ir šios funkcijos rezultatas (matrica E).

Polinominis laikas – skaičiavimo laiko priklausomybė nuo sisteminių parametrų yra polinominė.

Priimtinas laikas = polinominis laikas.

Pusgrupės idealas – pogrupis, kuris yra uždaras daugybos atžvilgiu visiems pusgrupės elementams.

RSA – asimetrinio šifravimo protokolas, pavadintas pagal jį sukūrusių autorių pavardes (Rivest, Shamir, Adleman).

Simetrinio šifravimo protokolas – matematinis algoritmas, kuris leidžia užšifruoti pranešimą ir iššifruoti šifrogramą naudojant tą patį šifravimo raktą, kuris turi būti laikomas paslaptyje.

STR – raktų apsikeitimo protokolas, pavadintas pagal jį sukūrusių autorių pavardes (Sakalauskas, Tvarijonas, Raulynaitis).

Statistinė kriptoanalizė – kriptoanalizės metodai, kurių tikslas yra pseudoatsitiktinių skaičių generatorių, sukurtų vienkryptės funkcijos pagrindu, analizė siekiant prognozuoti funkcijos reikšmes remiantis turimais doumenimis.

Vienkryptė funkcija - funkcija, kurios reikšmė yra apskaičiuojama per priimtiną laiką, tačiau rasti funkcijos argumentą pagal jos reikšmę per priimtiną laiką yra neįmanoma.

PAŽYMĖJIMAI

$\oplus$ – XOR operacija (sumos moduliu 2 pabičiui operacija).

$|A|$ – aibės A galia.

${}^{X}Q$ – matrica Q keliama matriciniu laipsniu X iš kairės.

Q^{Y} – matrica Q keliama matriciniu laipsniu Y iš dešinės.

$\varphi(n)$ – Oilerio funkcija nuo n.

$\lambda(n)$ – Karmaiklo funkcija nuo n.

$\gcd(a, b)$ – dviejų skaičių didžiausias bendras daliklis

$Id(\boldsymbol{S})$ – multiplikacinės pusgrupės $\boldsymbol{S}$ idealas.

L – saugumo lygis.

m – kvadratinių matricų eilė.

n – multiplikacinės grupės $\boldsymbol{Z_n}^*$ arba pusgrupės $\boldsymbol{Z_n}^{\#}$ dydžio parametras.

p – pirminis skaičius, apibrėžiantis multiplikacinę pusgrupę.

r – skaitinio žiedo dydžio parametras.

$\boldsymbol{Z_n}$ – baigtinis žiedas, kurį sudaro sveikieji skaičiai nuo 0 iki n – 1. Sudėties ir daugybos operacijos šiame žiede atliekamos moduliu n.

$\boldsymbol{Z_n}^*$ – multiplikacinė grupė, kurią sudaro sveikieji skaičiai tarp 0 ir n – 1, kurie yra tarpusavyje pirminiai su n.

$\boldsymbol{Z_n}^{\#}$ - multiplikacinė pusgrupė, kurią sudaro grupės $\boldsymbol{Z_n}^*$ ir idealo $Id(\boldsymbol{Z_n})$ elementai.

1. ĮVADAS

Šiuolaikinio pasaulio jau negalime įsivaizduoti be informacinių technologijų. Elektroninis paštas, socialiniai tinklai, elektroninė bankininkystė, elektroninis balsavimas – štai tik maža dalis paslaugų, naudojančių elektroninę terpę, kurios yra siūlomos šiuolaikiniam vartotojui. Dažnai vartotojai, naudodami elektroninę terpę, siunčia slaptą informaciją. Šis procesas reikalauja saugumo, kadangi slaptos informacijos paviešinimas gali turėti nemalonių pasekmių ne tik neatsargiam vartotojui, bet ir jo pažįstamiems, o kartais net ir valstybei.

Kriptografinis saugumas plačiąja prasme apima tokius aspektus: informacijos konfidencialumą, autentiškumą, vientisumą, asmens identifikaciją [1], [2]. Šiems tikslams užtikrinti yra kuriami tokie kriptografiniai primityvai kaip raktų apsikeitimo protokolai, duomenų šifravimo protokolai, elektroniniai parašai. Šiame darbe mes suprantame kriptografinį protokolą kaip struktūrizuotą algoritmų rinkinį, naudojant kurį du arba daugiau vartotojų gali slaptai bendrauti tarpusavyje viešuoju kanalu. Protokolo veiksmai atliekami tam tikra tvarka. Be to protokole būtinai turi dalyvauti bent du vartotojai, t.y. vienas vartotojas negali įvykdyti protokolo [2].

Ypatingas dėmesys kriptografijai buvo skirtas ir antrojo pasaulinio karo metu, kai buvo bandoma nulaužti vieno žinomiausių šifravimo aparato – Enigmos mašinos šifrą. Šią užduotį pirmasis išsprendė Alanas Tiuringas, kuris šiandien yra laikomas vienu iš šiuolaikinės kriptografijos mokslo pradininkų. Tačiau, nors Enigmos mašinos šifras buvo nulaužtas, pati šios mašinos idėja yra naudojama ir dabar kuriant kvantinius šios mašinos modelius [3].

Enigmos mašina yra simetrinės kriptografijos pavyzdys, t.y. slapto pranešimo šifravimas ir iššifravimas vykdavo naudojant tuos pačius rotorių bei jungiklių porų nustatymus [4]. Šiuolaikine kalba šie mašinos nustatymai yra bendras šifravimo raktas. Šios kriptografijos šakos šifravimo protokolų supaprastinta struktūra atrodo taip [2]:

- Aldona ir Bronius susitaria dėl simetrinio šifravimo sistemos.
- Aldona ir Bronius susitaria dėl bendrojo šifravimo rakto.
- Bronius užšifruoja savo pranešimą, naudodamas simetrinio šifravimo protokolą ir bendrąjį šifravimo raktą. Tokiu būdu Bronius gauna šifrogramą, kurią jis išsiunčia Aldonai.
- Aldona iššifruoja Broniaus šifrogramą, naudodama simetrinio šifravimo protokolą ir bendrąjį šifravimo raktą. Tokiu būdu Aldona gali perskaityti pradinį Broniaus pranešimą.

Kadangi šiuo atveju Aldona ir Bronius naudoja tą patį raktą duomenims užšifruoti ir iššifruoti, tai simetrinio šifravimo protokolus galima pavadinti „kriptografiniais seifais“, t.y. bet kuris vartotojas, kuris turi raktą nuo šio seifo gali perskaityti Broniaus pranešimą. Tačiau, jeigu Broniaus pranešimas yra skirtas ne Aldonai, o kitam

vartotojui – Kamilei, tai šie du vartotojai turi susitarti dėl tarpusavio bendro šifravimo rakto. Tai reiškia, kad Bronius turi laikyti paslaptyje jau du raktus, kurių pirmas skirtas bendrauti su Aldona, o antras – su Kamilę. Tai yra vienas iš protokolo trūkumų. Be to susitarimas dėl bendrojo rakto turi vykti slaptuoju kanalu, kadangi kitu atveju apie šį raktą gali sužinoti ir kenkėjai, kurių tikslas yra trukdyti vartotojų tarpusavio bendravimui [2].

Šias problemas sėkmingai sprendžia 1976 metais atsiradusi nauja kriptografijos šaka – asimetrinė kriptografija, kuri nuo simetrinės skiriasi tuo, kad naudoja dviejų tipų raktus – viešąjį ir slaptąjį. Asimetrinės kriptografijos pradininkai Witfield‘as Diffie ir Martin‘as Hellman‘as savo straipsnyje [5] pasiūlė būdą kaip naudojant slaptąjį ir viešąjį raktus protokolo dalyviai Aldona ir Bronius gali sudaryti bendrąjį šifravimo raktą. Slaptasis raktas yra žinomas tik pačiam rakto savininkui, o viešasis raktas yra matematiškai surištas su slaptuoju raktu ir yra žinomas visiems tinklo vartotojams. Vienas iš pagrindinių reikalavimų viešajam raktui yra tai, kad šis raktas neatskleistų jokios informacijos apie slaptąjį raktą. Šios kriptografijos šakos šifravimo protokolų supaprastinta struktūra atrodo taip [2]:

- Aldona ir Bronius susitaria dėl asimetrinio šifravimo sistemos.
- Aldona persiunčia Broniui savo viešąjį raktą. Šis raktas yra matematiškai surištas su Aldonos slaptuoju raktu, kurį ji turi laikyti paslaptyje.
- Bronius užšifruoja savo pranešimą, naudodamas Aldonos viešąjį raktą. Tokiu būdu Bronius gauna šifrogramą, kurią jis išsiunčia Aldonai.
- Aldona iššifruoja Broniaus šifrogramą, naudodama savo slaptąjį raktą. Tokiu būdu Aldona gali perskaityti pradinį Broniaus pranešimą.

Kadangi šiuo atveju duomenims užšifruoti ir iššifruoti naudojami skirtingi raktai, tai tokio tipo protokolus galima pavadinti „kriptografine pašto dėžute“, t.y. bet kas gali užšifruoti pranešimą Aldonos viešuoju raktu, tačiau iššifruoti gautą šifrogramą gali tik pati Aldona, kadangi tik ji žino slaptąjį raktą. Taip pat, kadangi pranešimai yra šifruojami viešuoju raktu, tai literatūroje asimetrinę kriptografiją dažnai vadina viešojo rakto kriptografija. Dažnai vartotojų, kurie naudoja tą pačią šifravimo sistemą yra daug. Asimetrinės šifravimo sistemos leidžia vartotojui turėti tik vieną porą raktų – slaptąjį ir viešąjį. Tokiu atveju viešieji vartotojų raktai gali būti saugojami viešoje duomenų bazėje. Dėl šios priežasties Aldonai nereikia siusti savo viešojo rakto Broniui, kadangi šį raktą Bronius pasiima iš duomenų bazės. Matome, kad Aldona nedalyvauja protokole tol, kol negauna šifrogramos iš Broniaus [2].

Yra pasiūlyta nemažai asimetrinės kriptografijos primityvų. Du pagrindiniai aspektai, leidžiantys konkuruoti su žinomais protokolais, yra saugumas ir efektyvi realizacija ribotų resursų sistemose. Nuolat besivystant informacinėms technologijoms vis lengviau tampa įgyvendinti žinomas atakas prieš populiarius protokolus. Norint apsiginti prieš šias atakas reikia keisti sistemos parametrus, o tai neigiamai atsispindi protokolo efektyvumui. Taip pat netyli ir kriptoanalitikai, kurie kuria naujus kriptoanalizės metodus ir atakas. Vienos greičiausių kriptoatakų prieš šiuo metu plačiausiai taikomus protokolus (Diffie-Hellman, RSA) remiasi kvantinę

kriptoanalizę, kadangi šie protokolai nėra paremti NP-pilnais uždaviniais. Dėl šių priežasčių turi būti kuriami nauji kriptografiniai primityvai, kurie būtų atsparūs esamiems kriptoanalizės metodams. Pagrindiniai reikalavimai kuriant naują asimetrinio šifravimo protokolą yra šie:

- Korektiškas šifro veikimas. Šifravimo protokolas turi būti sudarytas taip, kad sugebėtų saugiai užšifruoti slaptą informaciją bei teisingai ją iššifruoti naudojant bendrąją asimetrinės kriptografijos schemą. Tai yra esminis reikalavimas bet kokiam asimetrinio šifravimo protokolui, kadangi neteisingai iššifruoti duomenys yra beverčiai.
- Viešojo rakto saugumas. Kadangi vartotojo viešasis ir slaptasis raktai yra tarpusavyje matematiškai surišti, tai potencialus kenkėjas, turėdamas vartotojo viešąjį raktą, negali jo panaudoti slaptajam vartotojo raktui gauti, t.y. viešasis vartotojo raktas neatskleidžia kenkėjui jokios informacijos apie jo slaptąjį raktą.
- Šifrogramos saugumas. Šis reikalavimas reiškia, kad bus išsaugotas siunčiamos informacijos slaptumas, t.y. tik pranešimo adresatas, naudodamas savo slaptąjį raktą, gali pasakyti kokia informacija yra siunčiama. Čia laikoma, kad kenkėjas gali žinoti ne tik užšifruotą pranešimą, bet ir viešuosius sistemos parametrus bei pranešimo adresato viešąjį raktą. Saugus asimetrinio šifravimo protokolas turi būti sudarytas taip, kad naudojant šiuos duomenis atstatyti pradinį pranešimą per priimtiną laiką būtų neįmanoma.
- Efektyvi protokolo realizacija. Kadangi šiuo metu populiarėja išmanieji prietaisai, kurie turi ribotus skaičiavimo bei atminties resursus, tai sukurtas protokolas turi būti pakankamai greitai realizuojamas tokiose sistemose.

Nors šiuo metu kavntiniai kompiuteriai dar nėra realizuoti, jau dabar yra aktualu kurti kriptografinius protokolus, kurie būtų atsparūs kvantinei kriptoanalizei. Mes darome prielaidą, kad siūlomas šiame darbe protokolas yra atsparus kvantinei kriptoanalizei, kadangi šis protokolas remiasi MLF uždaviniu, kurio sudėtingumas yra panašus į MQ (multivariate quadratic system of equations) uždavinį. Yra įrodyta, kad MQ uždavinys priklauso NP-pilnųjų uždavinių klasei. Tokio tipo uždaviniai yra atsparūs kvantinei kriptoanalizei.

Siekiant efektyvios protokolo realizacijos yra svarbu ne tik užtikrinti šifrogramos saugumą, bet ir optimizuoti realizacijos efektyvumą. Svarbų vaidmenį čia turi elektros energijos suvartojimas. Dėl šios priežasties skyrelyje 7.3 mes pateiksime mūsų protokolo greitaveikos palyginimo rezultatus su klasikiniais protokolais. Šio tyrimo tikslas yra parodyti, kad protokolui įvykdyti reikia sunaudoti mažiau skaičiavimo resursų, taip taupant energiją ir pinigus.

Darbo tikslas ir uždaviniai

Pagrindinis šio darbo tikslas yra sudaryti naują asimetrinio šifravimo protokolą, kurio saugumas būtų paremtas matricinio laipsnio funkcijos apgręžiamumo sudėtingumu, ir kuri postuluojama esanti vienkrypte funkcija.

Suformuluotam tikslui pasiekti buvo sprendžiami šie uždaviniai:

1. Ištirti postuluojamos vienkryptės funkcijos algebrines ir statistines savybes.
2. Pritaikyti postuluojamą vienkryptę funkciją asimetrinio šifravimo protokolui sudaryti.
3. Sudaryti siūlomo protokolo saugumo tyrimo metodiką bei įvertinti jo atsparumą statistinei ir algebrinei kriptoanalizei.
4. Nustatyti siūlomo protokolo pagrindinius saugumo parametrus ir jų saugias reikšmes.
5. Įvertinti protokolo greitaveiką lyginant jį su praktikoje naudojamais El-Gamal, elipsinių kreivių, RSA asimetrinio šifravimo protokolais.

Tyrimų metodika

Sprendžiant darbe suformuluotus uždavinius yra taikomi algebros, skaičių, tikimybių teorijos, statistikos metodai. Siūlomo protokolo korektiškumas, saugumas bei realizacijos efektyvumas buvo tirti analiziniais metodais ir eksperimentiniu būdu, naudojant sukurtą programinę priemonę pasiūlytam protokolui realizuoti.

Darbo mokslinis naujumas

1. Darbe sukurtas originalus asimetrinio šifravimo protokolas paremtas nauja vienkrypte funkcija, kuri iki šiol nebuvo panaudota asimetriniam šifravimui.
2. Naudojamos algebrinės struktūros leidžia pagrįsti siūlomo protokolo saugumą statistinės kriptoanalizės atžvilgiu.
3. Vartotojo viešojo rakto saugumas yra paremtas naujo sudėtingo uždavinio sprendimu baigtinėje multiplikacinėje matricų pusgrupėje.
4. Sukurtas protokolas yra realizuojamas ribotų skaičiavimo resursų aplinkose efektyviau, negu šiuo metu plačiausiai taikomi protokolai, kadangi algebrinėms operacijoms atlikti naudoja paieškos lenteles ir nereikalauja specialių procesorių, skirtų dirbti su dideliais skaičiais.

Pagrindiniai ginamieji teiginiai

1. Sukurtas originalus asimetrinio šifravimo protokolas, kurio saugumas yra paremtas nekomutatyviosios kriptografijos metodais.
2. Šiuo metu nežinomi kriptoanalizės lygčių sprendimo metodai, leidžiantys per polinominį laiką sukompromituoti pasiūlytą protokolą.
3. Siūlomas protokolas gali būti efektyviai realizuotas ribotų resursų sistemose.

Darbo rezultatų aprobavimas

Disertacijos tema yra paskelbti du straipsniai, kurie turi mokslinės duomenų bazės „ISI Web of Science“ citavimo indeksą. Dar du straipsniai yra konferencijų pranešimų medžiagoje. Disertacijos tema buvo pristatyta Lietuvos Matematikos Draugijos 53-ioje konferencijoje Klaipėdoje bei tarptautinėse konferencijoje „BulCrypt 2012“ Sofijoje ir „Electronics 2013“ Palangoje.

Darbo struktūra

Kadangi darbe taikomos įvairių matematikos šakų sąvokos, tai šio darbo antrame skyriuje pateiksime tuos matematinės teorijos skyrius, kurie yra naudojami sudarant mūsų protokolą ir atliekant sukurto protokolo tyrimą.

Trečiame skyriuje pateiksime literatūros apžvalgą. Šiame skyriuje mes orientuojamės į asimetrinės kriptografijos protokolus. Skyriuje pateiksime dažniausiai praktikoje naudojamus protokolus, su kuriais mes lyginame mūsų protokolą, o taip pat ir nekomutatyviosios kriptografijos asimetrinio šifravimo protokolus. Aptarsime šių protokolų privalumus ir trūkumus.

Ketvirtame skyriuje pristatysime vienkryptę funkciją, kuri yra naudojama mūsų darbe. Pateiksime šios funkcijos algebrines ir statistines savybes.

Penkto skyriaus pagrindinis tikslas yra pristatyti mūsų protokolą ir ištirti jį naudojant vienkryptės funkcijos algebrines savybes. Protokolo veikimas yra demonstruojamas naudojant pavyzdį. Šiame skyriuje taip pat aprašoma protokolo silpnoji vieta bei siūlomi protokolo patobulinimai, kurie padeda išvengti šio trūkumo.

Šeštame skyriuje yra nagrinėjami pagrindiniai siūlomo protokolo parametrai ir pateikiama saugių viešųjų duomenų generavimo metodika. Taip pat šiame skyriuje siūlomas protokolas yra palyginamas su klasikiniais protokolais elementariųjų operacijų prasme.

Septintame skyriuje yra pateikiami pagrindiniai algoritmai, kurie buvo naudojami kuriant programinę priemonę siūlomam protokolui realizuoti. Naudojant sukurtą programinę priemonę siūlomas protokolas yra palyginamas su klasikiniais protokolais greitaveikos prasme.

Paskutiniuose skyriuose yra pateikiamos bendros šios disertacijos išvados, cituotos literatūros sąrašas ir paskelbtų disertacijos tema publikacijų sąrašas. Disertacijos priede pateikti pagrindiniai algoritmai, kurie buvo naudojami protokolui realizuoti.

2. MATEMATINIS APARATAS

Kriptografijoje dažnai yra naudojamos baigtinės algebrinės struktūros tokios kaip grupės arba pusgrupės, žiedai arba laukai. Šį skyrių pradėsime nuo šių sąvokų apibrėžimų:

2.1 apibrėžimas. Aibė $\boldsymbol{G}$ su joje apibrėžta operacija $*$ vadinama *grupe* jeigu yra tenkinamos šios aksiomos:

1. Operacija $*$ yra uždara, t.y. bet kuriems aibės $\boldsymbol{G}$ elementams a ir b elementas $a * b \in \boldsymbol{G}$.
2. Operacija $*$ yra asociatyvi, t.y. bet kuriems aibės $\boldsymbol{G}$ elementams a, b ir c galioja lygybė $(a * b) * c = a * (b * c)$
3. Operacijos $*$ atžvilgiu egzistuoja toks elementas e, kad bet kuriam aibės elementui a yra teisinga lygybė $a * e = e * a = a$. Toks elementas e vadinamas *neutraliuoju elementu*.
4. Operacijos $*$ atžvilgiu bet kuriam aibės $\boldsymbol{G}$ elementui a egzistuoja toks šios aibės elementas $\tilde{a}$, kad yra teisinga lygybė $a * \tilde{a} = \tilde{a} * a = e$. Toks elementas $\tilde{a}$ vadinamas *atvirkštiniu elementu*.

Pastaba. Jeigu 4 aksioma yra netenkinama, tai aibė $\boldsymbol{G}$ su joje apibrėžta operacija $*$ vadinama *pusgrupe*.

Mokslinėje literatūroje dažnai grupė $\boldsymbol{G}$ arba pusgrupė $\boldsymbol{S}$ yra žymimos nenurodant apibrėžtos operacijos. Taip yra todėl, kad šios operacijos apibrėžimas dažnai yra aiškus iš konteksto. Kitas būdas nusakyti apibrėžtą operaciją yra žodinis. Kadangi dažnai tarp elementų įvedamos sudėties arba daugybos operacijos, tai pabrėžiant sudėties operaciją sakoma, kad (pus)grupė yra *adicinė*, o pabrėžiant daugybos operaciją tarp aibės elementų sakoma, kad (pus)grupė yra *multiplikacinė*. Adicinės grupės neutralusis elementas dažnai vadinamas grupės nuliu, o elementui x atvirkštinis elementas – priešingu elementu ir žymimas $(-x)$. Multiplikacinės grupės neutralusis elementas vadinamas grupės vienetu o elementui x atvirkštinis elementas žymimas x^{-1}.

Darbe dažnai iš grupių ar pusgrupių išskiriame tam tikrus elementų poaibius, kurie su juose apibrėžta operaciją taip pat sudaro grupę. Tokia algebrinė struktūra yra vadinama *pogrupiu*. Ypatingą svarbą mūsų darbe turi konkretus pogrupis, kuris yra vadinamas idealu.

2.2 apibrėžimas. Komutatyvios pusgrupės $\boldsymbol{S}$ pogrupis $Id(\boldsymbol{S})$ yra vadinamas *idealu*, jeigu su visais $a \in \boldsymbol{S}$ ir $i \in Id(\boldsymbol{S})$ yra tenkinama sąlyga

$$a * i \in Id(\boldsymbol{S}).$$

Darbe taip pat dažnai vartojamos žiedo ir lauko savokas. Apibrėžkime jas.

2.3 apibrėžimas Aibė $\boldsymbol{R}$ su dviejomis apibrėžtomis joje operacijomis + ir · yra vadinama *žiedu*, jeigu

a) aibė $\boldsymbol{R}$ su operacija + sudaro komutatyvią grupę;
b) aibė $\boldsymbol{R}$ operacijos · atžviligiu yra uždara;
c) operacijoms + ir · galioja distributivumo dėsniai, t.y.

$$(a + b) \cdot c = a \cdot c + b \cdot c;$$

$$c \cdot (a + b) = c \cdot a + c \cdot b.$$

Priklausomai nuo to, ar operacija · yra komutatyvi, žiedas gali būti atitinkamai komutatyvus arba nekomutatyvus.

2.4 apibrėžimas. Žiedas $\boldsymbol{F}$ su dviejomis apibrėžtomis joje operacijomis + ir · yra vadinamas *lauku*, jeigu aibės $\boldsymbol{F}$ ir $\boldsymbol{F}\backslash\{0\}$ su operacijomis + ir · atitinkamai sudaro komutatyvias grupes.

Mes darbe naudosime minėtas algebrines struktūras, kurios yra sudarytos iš sveikųjų skaičių. Tokio tipo struktūras mes vadinsime *skaitinėmis*. Šios struktūros yra patogios tuo, kad jos gali būti lengvai realizuotos naudojant bet kokią programavimo kalbą. Be to mes sieksime to, kad visus aibės elementus galima būtų atvaizduoti vienu baitu, t.y. visi nagrinėjamos struktūros elementai priklausytų intervalui [0; 255].

Mūsų darbe nagrinėjamos struktūros yra *baigtinės*, t.y. jos sudarytos iš baigtinio elementų kiekio. Dėl šios priežasties aritmetines operacijas tarp baigtinės aibės elementų reikia apibrėžti taip, kad jos tenkintų reikalavimus, kurie yra nurodyti aukščiau. Šiam tikslui mes naudosime modulinę aritmetiką, t.y. aritmetinius veiksmus atliksime skaičiuojant dalybos iš baigtinės grupės parametro n liekaną. Tokiu atveju sakysime, kad visi veiksmai yra atliekami moduliu n. Dažnai šioms operacijoms yra sudaromos Keilio lentelės, kurios leidžia greitai surasti operacijos rezultatą. Dėl šios priežasties Keilio lentelės dažnai vadinamos *paieškos lentelėmis* (angl. lookup tables). Nors šios lentelės taupo skaičiavimo laiką, tačiau jos turi būti saugojamos papildomai, o tai reikalauja papildomos atminties.

Tarkime, turime baigtinį sveikųjų skaičių žiedą $\boldsymbol{Z_n} = \{0, 1, 2, ..., n - 1\}$. Sudėties ir daugybos operacijos šiame žiede atliekamos moduliu n. Kadangi šiame darbe dažnai iš konteksto aišku kokiu moduliu atliekamos operacijos, tai skaičiavimo modulis dažnai yra praleidžiamas. Žinoma, kad jeigu parametras n yra pirminis skaičius, tai $\boldsymbol{Z_n}$ yra laukas.

Pavyzdys. Nagrinėkime sveikųjų skaičių žiedą $\boldsymbol{Z_5} = \{0, 1, 2, 3, 4\}$. Dviejų šio žiedo elementų a ir b sudėtį galima apibrėžti taip:

$$a + b = (a + b) \bmod 5,$$

$$a \cdot b = (a \cdot b) \bmod 5 .$$

čia žymėjimas mod 5 reiškia liekaną dalinant skaičių $(a+b)$ arba $(a \cdot b)$ iš 5. Sudėties ir daugybos operacijoms galima sudaryti paieškos lenteles, kurios atrodo taip:

2.1. lentelė Sudėties lentelė žiede $\boldsymbol{Z_5}$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

2.2. lentelė Daugybos lentelė žiede $\boldsymbol{Z_5}$

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	1	2	3	4
3	0	2	4	1	3
4	0	3	1	4	2

Iš šių lentelių matome, kad žiedas $\boldsymbol{Z_5}$ turi neutralius elementus sudėties atžvilgiu (šis elementas yra 0) ir daugybos atžvilgiu (šis elementas yra 1). Iš 2.1 lentelės matome, kad sudėties atžvilgiu kiekvienam elementui egzistuoja priešingas elementas $(-x) = 5 - x$, o iš 2.2 lentelės matome, kad kiekvienam elementui, išskyrus 0, egzistuoja atvirkštinis elementas, t.y. $1^{-1} = 1$, $2^{-1} = 3$, $3^{-1} = 2$, $4^{-1} = 4$. Taigi aibės $\boldsymbol{Z_5}$ ir $\boldsymbol{Z_5}\backslash\{0\}$ su sudėties ir daugybos operacijomis sudaro komutatyvias grupes, t.y. žiedas $\boldsymbol{Z_5}$ yra laukas.

Lauką $\boldsymbol{Z_5}$ palyginkime su žiedu $\boldsymbol{Z_6}$ = {0, 1, 2, 3, 4, 5}. Šiame žiede sudėties ir daugybos lentelės atrodo taip:

2.3. lentelė Sudėties lentelė žiede $\boldsymbol{Z_6}$

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

2.4. lentelė Daugybos lentelė žiede $\boldsymbol{Z_6}$

·	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Iš 2.3 lentelės matome, kad $\boldsymbol{Z_6}$ su sudėties operacija sudaro komutatyvią grupę. Tačiau iš 2.4 lentelės matome, kad atvirkštiniai elementai egzisuoja tik elementams 1 ir 5, t.y. $1^{-1}=1$ ir $5^{-1}=5$, o kitiems elementams atvirkštiniai elementai neegzistuoja. Taigi aibė $\boldsymbol{Z_6}\backslash\{0\}$ su daugybos operacija sudaro tik pusgrupę. Dėl šios priežasties žiedas $\boldsymbol{Z_6}$ nėra laukas.

Kadangi lauke kiekvienam elementui (išskyrus 0) egzistuoja atvirkštinis elementas, tai dažnai sakoma, kad laukas yra žiedas su dalyba, t.y. kiekvieną elementą galima padalinti iš nenulinio elemento. Ši operacija lauke $\boldsymbol{Z_n}$ apibrėžiama taip:

$$\frac{a}{b}=\left(a\cdot b^{-1}\right)\bmod n.$$

Pavyzdys. Lauke $\boldsymbol{Z_5}$ = {0, 1, 2, 3, 4} turime

$$\frac{3}{2}=\left(3\cdot 2^{-1}\right)\bmod 5=\left(3\cdot 3\right)\bmod 5=4.$$

Tačiau žiede $\boldsymbol{Z_6}$ šios dalybos atlikti negalime, nes jame neegzistuoja elementas 2^{-1}. Nors bendruoju atvėju šiame žiede dalybos atlikti negalime, tačiau turi prasmę trupmena

$$\frac{3}{5}=\left(3\cdot 5^{-1}\right)\bmod 6=\left(3\cdot 5\right)\bmod 6=3.$$

Jeigu iš aibės $\boldsymbol{Z_n}$ išrenkame tik tuos elementus, kurie yra tarpusavyje pirminiai su parametru n, tai, apibrėžę šioje aibėje daugybos operaciją minėtu būdu, gauname multiplikacinę grupę. Šią grupę toliau žymėsime $\boldsymbol{Z_n}^*$. Šios grupės elementų kiekį (šį kiekį vadinsime *grupės eile*) nusako Oilerio funkcija, kurios apibrėžimas yra toks [6]:

2.5 apibrėžimas. Natūraliųjų skaičių, mažesnių už n ir tarpusavyje pirminių su n skaičius yra vadinamas *Oilerio funkcija nuo n* ir žymimas $\phi(n)$.

Taigi multiplikacinės grupės $\boldsymbol{Z_n}^*$ eilė $|\boldsymbol{Z_n}^*|=\phi(n)$. Oilerio funkcija tenkina multiplykatyvumo savybę, t.y.

$$\phi(pq) = \phi(p)\phi(q). \tag{2.1}$$

Taip pat yra akivaizdu, kad pirminiam skaičiui p Oilerio funkcijos reikšmė $\phi(p) = p - 1$.

Pavyzdys. Nagrinėkime sveikųjų skaičių žiedą $\boldsymbol{Z_6}$ = {0, 1, 2, 3, 4, 5}. Multiplikacinė grupė $\boldsymbol{Z}_6^* = \{1, 5\}$, nes tik šie elementai yra tarpusavyje pirminiai su 6. Būtent šiems elementams galioja dalybos operacija. Šios grupės eilė yra:

$$\left|\boldsymbol{Z}_6^*\right| = \phi(6) = \phi(2\cdot 3) = \phi(2)\cdot\phi(3) = (2-1)\cdot(3-1) = 2.$$

Kadangi multiplikacinės grupės $\boldsymbol{Z_n}^*$ daugybos operacija yra asociatyvi, tai šioje struktūroje galima apibrėžti kėlimo laipsniu operaciją mums įprastu būdu. Svarbų vaidmenį mūsų darbe vaidina šios mutiplikacinės grupės struktūra. Savo darbe kėlimo laipsniu operaciją praplėsime į matricų algebrines struktūras. Šią operaciją ir Oilerio funkciją sieja mažoji Oilerio teorema, kuri skamba taip [6]:

Oilerio teorema. Bet kuriam elementui $a \in \boldsymbol{Z_n}^*$ teisinga lygybė $a^{\phi(n)} = 1$.

Bendruoju atveju sakysime, kad multiplikacinės grupės $\boldsymbol{Z_n}^*$ elemento *g periodas* yra natūralusis skaičius *T*, jeigu šis skaičius yra mažiausias iš skaičių, kuriems yra tenkinama sąlyga

$$g^T \bmod n = 1. \tag{2.2}$$

Reikia pastebėti, kad nors Oilerio funkcija ir tenkina šią sąlygą, tačiau šis skaičius nebūtinai yra mažiausias, taigi Oilerio funkcijos reikšmė nebūtinai yra elemento *a* periodas. Jeigu egzistuoja toks elementas $g \in \boldsymbol{Z_n}^*$, kad Oilerio funkcijos reikšmė yra elemento *g* periodas, tai grupė $\boldsymbol{Z_n}^*$ vadinama *cikline*, o elementas *g* vadinamas grupės $\boldsymbol{Z_n}^*$ *generatoriumi*. Pastaroji savoka yra svarbi tuo, kad generatorius turi didžiausią iš grupės $\boldsymbol{Z_n}^*$ elementų periodą. Dėl šios priežasties grupės generatoriai yra svarbūs kriptografijos požiuriu ir yra naudojami tokiuose primityvuose kaip Difio-Helmano raktų apsikeitimas ir El-Gamalio elektroninis parašas arba asimetrinis šifravimas. Priešingu atveju grupė $\boldsymbol{Z_n}^*$ vadinama *necikline*, o šios grupės elementų periodams galioja Lagranžo teorema. Šią teoremą galima suformuluoti taip:

Lagranžo teorema. Kiekvieno baigtinės grupės ***G*** ciklinio pogrupio **Γ** eilė dalija grupės ***G*** eilę.

Iš šios teoremos matome, kad tuo atveju, kai elemento $a \in \boldsymbol{Z_n}^*$ periodas yra mažesnis už (*n*), šis periodas dalija Oilerio funkcijos reikšmę. Neciklinėms grupėms didžiausias galimas elemento periodas yra nusakomas Karmaiklo funkcijos $\lambda(n)$ reikšme, t.y. periodo reikšmė $T = \lambda(n)$ yra didžiausia galima. Šis rezultatas yra žinomas kaip Karmaiklo teorema ir skamba taip [7]:

Karmaiklo teorema. Bet kuriam elementui $a \in \boldsymbol{Z}_n^*$ teisinga lygybė $a^{\lambda(n)} = 1$.

Šios teoremos taikymas leidžia mums bet kokius laipsnius redukuoti moduliu $\lambda(n)$. Bendruoju atveju Karmaiklo funkcijos apibrėžimas yra pakankamai ilgas [7], todėl čia pateiksime tik šios funkcijos atskirą atvejį, kai multiplikacinės grupės parametras n yra dviejų pirminių skaičių p ir q sandauga, t.y. $n = pq$. Šiuo atveju Karmaiklo funkcija yra apskaičiuojama taip:

$$\lambda(n) = \text{lcm}(p-1; q-1), \tag{2.3}$$

čia išraiška lcm(a, b) pažymėtas skaičių a ir b mažiausias bendras kartotinis.

Pavyzdys. Nagrinėkime sveikųjų skaičių žiedą $\boldsymbol{Z_{15}}$. Tada $\phi(15) = 8$. Taigi pagal Oilerio teoremą bet kokiam elementui $a \in \boldsymbol{Z}_{15}^*$ galioja lygybė $a^8 = 1$. Tačiau didžiausias galimas elemento a periodas yra 4, nes

$$\lambda(15) = \text{lcm}(5-1; 3-1) = 4,$$

t.y. elementui a galioja lygybė $a^4 = 1$. Šios grupės elementams galima sudaryti kėlimo laipsniu lentelę, kuri atrodo taip:

2.5. lentelė Kėlimo laipsniu lentelė multiplikacinėje grupėje $\boldsymbol{Z}_{15}^*$.

^	1	2	3	4
1	1	1	1	1
2	2	4	8	1
4	4	1	4	1
7	7	4	13	1
8	8	4	2	1
11	11	1	11	1
13	13	4	7	1
14	14	1	14	1

Pastaba. Užrašą a^b suprantame kaip a^b.

Iš 2.5 lentelės matome, kad keliant ketvirtuoju laipsniu gauname tą patį rezultatą, kaip ir keliant elementą nuliniu laipsniu. Dėl šios priežasties galime šį laipsnį pakeisti nuliniu laipsniu, t.y. atlikti laipsnio redukciją moduliu 4.

Iš Karmaiklo funkcijos apibrėžimo matome, kad ši funkcija tenkina sąlygą

$$\lambda(n) \le \phi(n). \tag{2.4}$$

Iš pastarosios išraiškos matome, kad grupė $\boldsymbol{G}$ yra ciklinė tada ir tik tada, kai šioje išraiškoje galioja lygybė.

Tačiau mūsų darbe yra svarbu ne tik žinoti didžiausią galimą elementų periodą, bet ir užtikrinti tam tikrų grupės $\boldsymbol{Z_n^*}$ pogrupių egzistavimą. Dėl šios priežasties darbe yra pritaikomos norvegų matematiko Liudvigo Sylovo teoremos. Šios teoremos leidžia apgręžti Lagranžo teoremą tam tikriems periodams. Bendruoju atveju šios teoremos formuluojamos nekomutatyviosioms grupėms. Tačiau, kadangi mes Sylovo teoriją taikysime komutatyviosioms grupėms, tai kai kurios iš šių teoremų tenkinamos automatiškai. Čia pateiksime Sylovo teorijos pagrindinį rezultatą, kurį mes taikome savo darbe. Jį suformuluosime taip [8]:

Sylovo teorema. Tarkime, kad pirminis skaičius p dalija komutatyvios grupės $\boldsymbol{G}$ eilę. Be to $|\boldsymbol{G}| = kp$, čia p nedalija k. Tada grupėje $\boldsymbol{G}$ egzistuoja ciklinis pogrupis, kurio eilė yra p. Šis pogrupis vadinamas *p-tosios eilės Sylovo pogrupiu*.

Pavyzdys. Multiplikacinės grupės $\boldsymbol{Z}_{13}^*$ eilė $|\boldsymbol{Z}_{13}^*| =$ (13) = 12 = 3 · 4. Šioje grupėje egzistuoja 3-osios eilės Sylovo pogrupis $\boldsymbol{\Gamma}_{3,\,13} = \{1, 3, 9\}$. Šis 3-osios eilės Sylovo pogrupis yra vienintelis.

Bendruoju atvėju Sylovo pogrupio eilė gali būti ir pirminio skaičiaus laipsnis. Minėtame pavyzdyje tai būtų pogrupis $\boldsymbol{\Gamma}_{4,\,13} = \{1, 5, 8, 12\}$. Tačiau tokie pogrupiai mūsų nedomina. Taip pat reikia paminėti, kad pogrupis {1, 12} nėra Sylovo pogrupis, kadangi (13) = 12 = 2 · 6, tačiau 2 dalija 6.

Nagrinėjant multiplikacinės grupės $\boldsymbol{Z_n^*}$ svarbų vaidmenį turi kinų liekanų teorema. Šią teoremą taikome specifinėms grupėms, todėl pateiksime supaprastintą šios teoremos versiją, kuri taikoma tuo atveju, kai $n = pq$ [1], [6].

Kinų liekanų teorema. Tarkime, kad skaičiai p ir q (nebūtinai pirminiai) yra tarpusavyje pirminiai, t.y. gcd(p, q) = 1. Tada egzistuoja vienintelis multiplikacinės pusgrupės $\boldsymbol{Z_n}$ elementas x, kuris tenkina lyginių sistemą

$$\begin{cases} x \bmod p = x_p \\ x \bmod q = x_q \end{cases}, \tag{2.5}$$

kai elementai x_p ir x_q yra žinomi.

2.1 išvada. Egzistuoja izomorfizmas tarp grupės $\boldsymbol{G_n}$ ir grupių $\boldsymbol{G_p}$ ir $\boldsymbol{G_q}$ tiesioginės sandaugos $\boldsymbol{G_p} \times \boldsymbol{G_q}$. Šis rezultatas nepriklauso nuo grupės operacijos [1].

Pavyzdys. Tarkime turime baigtinį žiedą $\boldsymbol{Z_{15}} = \{1, 2, 3, ..., 14\}$. Nagrinėkime aditinę grupę $\boldsymbol{Z_{15}}$ ir multiplikacinę grupę $\boldsymbol{Z_{15}^*}$. Redukavę kiekvieną aibės $\boldsymbol{Z_{15}}$ elementą moduliais 3 ir 5 turime tokius rezultatus:

2.6 lentelė. Aibės $\boldsymbol{Z_{15}}$ elementai, redukuoti moduliais 3 ir 5.

a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a mod 3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
a mod 5	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4

Matome, kad kiekvienas aibės $\boldsymbol{Z_{15}}$ elementas a gali būti vieninteliu būdu atvaizduotas į elementų a_3 ir a_5 porą (a_3, a_5), čia indeksas rodo redukcijos modulį. Be to kiekvieną porą (a_3, a_5) atitinka vienintėlis elementas a. Tai reiškia, kad adicinė grupė $\boldsymbol{Z_{15}}$ yra izomorfinė tiesioginei sandaugai $\boldsymbol{Z_3 \times Z_5}$, o multiplikacinė grupė $\boldsymbol{Z_{15}^*}$ yra izomorfinė grupei $\boldsymbol{Z_3^* \times Z_5^*}$, kadangi toks vaizdavimas išsaugo visas adicinės ir multiplikacinės grupių savybes.

Pastaroji išvada kartu su Kairmaiklo, Ležandro ir Sylovo teoremomis atskleidžia visą informaciją apie grupės $\boldsymbol{Z_n^*}$ struktūrą, t.y. informaciją apie šios grupės elementų periodus, pogrupius ir izomorfines grupes. Ši informacija yra svarbi sudarant mūsų protokolą ir analizuojant jo saugumą.

Nors kinų liekanų teorema ir užtikrina unikalaus elemento x egzistavimą, tačiau ji nenurodo jokio būdo kaip šį elementą apskaičiuoti. Tam tikslui reikia panaudoti multiplikacinės pusgrupės $\boldsymbol{Z_n}$ netrivialius idempotentus u ir v, t.y. tokius idempotentus, kad $u \neq 1$ ir $v \neq 1$. Šie idempotentai yra atitinkamai vaizduojami pusgrupės $\boldsymbol{Z_p \times Z_q}$ elementais (1, 0) ir (0, 1), o tai reiškia, kad bet kurį pusgrupės $\boldsymbol{Z_n}$ elementą galima išreikšti per šiuos du bazinius pusgrupės $\boldsymbol{Z_p \times Z_q}$ elementus vieninteliu būdu naudojant formulę [1]:

$$x = (ux_p + vx_q) \bmod n. \tag{2.6}$$

Pavyzdys. Išspręskime lyginių sistemą

$$\begin{cases} x \bmod 3 = 2 \\ x \bmod 5 = 3 \end{cases}.$$

Kadangi multiplikacinės pusgrupės $\boldsymbol{Z_{15}}$ netrivialūs idempotentai yra $u = 10$ ir $v = 6$, tai

$$x = (10 \cdot 2 + 6 \cdot 3) \bmod 15 = 8.$$

Iš 2.6 lentelės matome, kad elementas surastas teisingai.

Protokolui sukurti yra naudojamos kvadratinės matricos, o atliekant protokolo saugumo tyrimus nagrinėjamos matricinės lygtys, kurios yra apibrėžtos virš skaitinio žiedo, t.y. visų matricų elementai turi priklausyti šiam žiedui. Nors mums nepavyko rasti literatūros, kurioje būtų aprašomi tokių lygčių sprendimo metodai, tačiau naudojant kinų liekanų teoremą visiems matricų elementams galima panaudoti

matricinių lygčių virš laukų sprendimo būdus. Šis faktas yra plačiai naudojamas mūsų darbe.

Darbe naudojant kvadratines matricas yra apibrėžiama vienkryptė funkcija, kuri yra naudojama protokolui konstruoti. Vienkryptės funkcijos yra svarbi šiuolaikinės kriptografijos dalis, kadangi būtent šiomis funkcijomis remiasi visi šiuolaikiniai kriptografiniai primityvai. Šias funkcijas galima apibrėžti taip [9]:

2.6 apibrėžimas. Funkcija $f : \boldsymbol{A} \rightarrow \boldsymbol{B}$ yra vadinama *vienkrypte funkcija*, jeigu

a) bet kuriam $x \in \boldsymbol{A}$, galima per polinominį laiką apskaičiuoti $y = f(x)$;
b) jeigu yra žinoma funkcijos f reikšmė $y \in \boldsymbol{B}$, tai yra sunku (angl. algorithmically infeasible) apskaičiuoti tokį $x \in \boldsymbol{A}$, kad $f(x) = y$, t.y. nėra tokio algoritmo, kuris leistų apskaičiuoti funkcijos f argumento x per polinominį laiką.

Pastaba. Šiame darbe žodžius „sunku“, „sudėtinga“ arba „lengva“ reikia suprasti sudėtingumo teorijos prasme, t.y. algoritmas, kurį galima įvykdyti per polinominį laiką naudojant deterministinę Tiuringo mašiną (sakoma, kad toks algoritmas priklauso sudėtingumo klasei P) yra lengvas, o algoritmas, kurį galima įvykdyti per polinominį laiką tik naudojant nedeterministinę Tiuringo mašiną (sakoma, kad toks algoritmas priklauso sudėtingumo klasei NP), yra sunkus (sudėtingas).

Kadangi vienkrypčių funkcijų egzistavimas iki šiol nėra įrodytas (minėto fakto įrodymas yra susietas su sudėtingumo teorijos P = NP uždavinio sprendimu), tai funkcijas, kurios tenkina 2.6 apibrėžimą dažnai vadina kandidatėmis į vienkryptes funkcijas [10]. Šiuolaikinė asimetrinė kriptografija remiasi prielaida, jog vienkryptės funkcijos egzistuoja. Tokių funkcijų pavyzdžiai yra dviejų priminių skaičių sandauga arba diskretinio logaritmo funkcija. Pastarosios funkcijos idėją mes naudosime savo darbe. Šios funkcijos apibrėžimas yra toks:

2.7 apibrėžimas. Tarkime, turime du baigtinės grupės $\boldsymbol{G}$ elementus a ir g, tokius, kad

$$a = g^x . \tag{2.7}$$

Laipsnį x vadiname *diskrečiuoju logaritmu* (arba *indeksu* [6]) pagrindu g ir žymime

$$x = \mathrm{ld}_g\, a .$$

Jeigu neegzistuoja toks laipsnis x, su kuriuo lygybė (2.7) galioja, tai sakome, kad diskretusis logaritmas pagrindu g neegzistuoja. Aišku, kad diskretusis logaritmas visada egzistuoja, jeigu jo pagrindas g yra grupės $\boldsymbol{G}$ generatorius. Kadangi grupėje $\boldsymbol{Z}_n^*$ visi laipsniai turi būti redukuojami moduliu $\lambda(n)$, tai šios algebrinės struktūros elementui a galioja nelygybė

$$0 \le \mathrm{ld}_g\, a \le \lambda(n) - 1, \tag{2.8}$$

be to $\mathrm{ld}_g\, a = 0$ tada ir tik tada, kai a = 1, ir $\mathrm{ld}_g\, a = \lambda(n) - 1$ tada ir tik tada, kai $a = g^{-1}$. Diskretiniam logaritmui galioja šios savybės:

$$\mathrm{ld}_g\,(a \cdot b) = \mathrm{ld}_g\, a + \mathrm{ld}_g\, b\ ; \tag{2.9}$$

$$\mathrm{ld}_g\, a^x = x \cdot \mathrm{ld}_g\, a\ . \tag{2.10}$$

Taigi matome, kad diskretinis logaritmas yra logaritmo begaliniame lauke analogas baigtiniame lauke. Diskretinio logaritmo reikšmei nustatyti galima panaudoti kėlimo laipsniu paieškos lentelę.

Pavyzdys. Nagrinėkime multiplikacinę grupę $\boldsymbol{Z}_{15}^{*}$. Pasirinkime g = 7 ir raskime $\mathrm{ld}_7\, 2$, $\mathrm{ld}_7\, 4$, $\mathrm{ld}_7\, 13$. Tą galima padaryti naudojant 2.5 lentelę nagrinėjat lentelės eilutę, kurioje surašyti visi įmanomi elemento g = 7 kėlimo laipsniu rezultatai. Kadangi elemento 2 tarp jų nėra, tai $\mathrm{ld}_7\, 2$ neegzistuoja, o kitiems elementams turime $\mathrm{ld}_7\, 4 = 2$ ir $\mathrm{ld}_7\, 13 = 3$. Be to $13 = 7^{-1}$, kadangi $\mathrm{ld}_7\, 13 = \lambda(15) - 1$.

Vienkryptės funkcijos taip pat rišasi ir su pseudoatsitiktinių skaičių generatoriais. Yra žinomas faktas [11], [12], kad geros vienkryptės funkcijos generuojamos sekos negalima atskirti nuo pseudoatsitiktinės sekos. Tam tikslui svarbu yra užtikrinti maksimalią generuojamų reikšmių entropiją. Taip pat kiekviena vienkryptės funkcijos reikšmė turi būti generuojama su vienoda tikimybe, t.y. šios reikšmės turi būti pasiskirsčiusios pagal tolygųjį skirstinį. Statistines mūsų funkcijos savybes mes plačiai nagrinėsime 4.2 skyrelyje.

3. ASIMETRINĖS KRIPTOGRAFIJOS PROTOKOLAI

Literatūros apžvalgą pradėsime nuo klasikinės kriptografijos protokolų. Šie protokolai yra plačiai taikomi šiuolaikinėje kriptografijoje, ir naudoja komutatyvias algebrines struktūras. Dėl šios priežasties šiame skyrelyje pateikti protokolai priskiriami komutatyviosios asimetrinės kriptografijos šakai.

3.1. Komutatyvioji kriptografija

Trumpai pristatysime patį pirmą asimetrinės kriptografijos protokolą – Difio-Helmano raktų apsikeitimo protokolą. Mokslininkai kaip platformą savo protokolui panaudojo komutatyvią struktūrą – baigtinį lauką $\boldsymbol{Z_p}$, čia p yra pirminis skaičius ir $p \neq 2$. Praktikoje šis parametras dažnai pasirenkamas 1024 arba 2048 bitų ilgio, t.y. $p \sim 2^{1024}$ arba $p \sim 2^{2048}$. Parametrui p papildomai reikalaujama, jog bent vienas iš skaičiaus $p - 1$ daugiklių būtų pakankamai didelis. Difio-Helmano raktų apsikeitimo protokolas atrodo taip [5]:

- Aldona ir Bronius susitaria dėl viešųjų parametrų: lauko $\boldsymbol{Z_p}$ ir šio lauko generatoriaus g;
- Aldona atsitiktinai pasirenka skaičių x ir apskaičiuoja $a = g^x \bmod p$. Skaičių a Aldona siunčia Broniui;
- Bronius atsitiktinai pasirenka skaičių y ir apskaičiuoja $b = g^y \bmod p$ Skaičių b Bronius siunčia Aldonai;
- Aldona ir Bronius apskaičiuoja bendrą slaptą raktą $K = a^y \bmod p = (g^x)^y \bmod p = (g^y)^x \bmod p = b^x \bmod p$.

Protokolo saugumas remiasi tuo, kad, turint informaciją apie g ir a, rasti laipsnio rodiklio x reikšmę yra sudėtinga, jeigu baigtinio lauko parametras p yra didelis. Laipsnio rodiklio x paieškos uždavinį, kai duotas pagrindas g ir modulinės eksponentės reikšmė a, vadiname *diskretinio logaritmo uždaviniu* (DLU) [10]. Nors šis uždavinys yra sunkus, tačiau jis nepriklauso NP-pilnųjų uždavinių (angl. non-deterministic polynomial complete problem) klasei. Aišku, kad šį uždavinį visada galima išspręsti naudojant pilnąjį visų galimų variantų perrinkimą. Tačiau toks sprendimo būdas yra praktiškai neefektyvus, kai skaičius p yra didelis, nes reikalauja didelių laiko sąnaudų. Kiti DLU sprendimo algoritmai dažniausiai yra eksponentiniai arba subeksponentiniai. Eksponentinių algoritmų pavyzdžiai yra Šenkso algoritmas (angl. baby-step giant-step), kurio sudėtingumas yra $O\left(\sqrt{p}\right)$ ir Pohligo ir Helmano algoritmas, kurio sudėtingumas priklauso nuo skaičiaus $p - 1$ faktorizacijos [13]. Subeksponentinių algoritmų pavyzdžiai yra COS (autoriai – Don Coppersmith, Andrew Odlyzko, Richard Schroeppel) algoritmas ir skaičių lauko gardelės algoritmas, kuris šiuo metu yra plačiausiai taikomas pirminiams skaičiams $p \geq 10^{100}$ [13]. Taip pat Šenkso algoritmas gali būti patobulintas taip, kad galėtų spręsti skaičiaus faktorizacijos uždavinį [14].

Nuo 1976 metų buvo sukurta nemažai asimetrinės kriptografijos primityvų, tačiau dauguma jų netinka praktiniam taikymui. Pagrindinės tokių protokolų problemos yra viešojo ir slaptojo raktų ilgiai ir šifrogramos dydis. Kiti asimetriniai protokolai, tokie kaip RSA [15] ir ElGamal [16], kurie gali būti naudojami ne tik duomenims šifruoti, bet ir elektroniniam parašui, rado savo vietą šiuolaikiniame pasaulyje ir yra plačiai naudojami praktikoje. Šie protokolai taip pat naudoja komutatyvias struktūras – baigtinę skaičių multiplikacinę grupę. RSA šifravimo protokolas remiasi sudėtinio skaičiaus faktorizacijos uždaviniu ir yra naudojamas bankuose, socialiniuose tinkluose Facebook ir Twitter, elektroninio pašto sistemose (pvz., Gmail). 2012 metais buvo pasiūlyta panaudoti šį protokolą Google Cloud serveryje svarbiems duomenims saugoti užšifruotu pavidalu [17]. Šis protokolas atrodo taip [15]:

- Generuojami du dideli panašaus dydžio pirminiai skaičiai p ir q. Apskaičiuojama šių skaičių sandauga $n = pq$ ir Oilerio funkcijos reikšmė $(n) = (p - 1)(q - 1)$. Parametras n skelbiamas viešai.
- Aldona pasirenka atsitiktinį sveiką skaičių e ($1 < e < (n)$) tokį, kad $\gcd(e, (n)) = 1$, t.y. skaičiai e ir (n) yra tarpusavyje pirminiai. Naudojant išplėstinį Euklido algoritmą ji randa elementą d tokį, kad $ed = 1 \bmod (n)$. Jos viešasis raktas $PuK_A = e$, o slaptasis raktas $PrK_A = d$.

Bronius užšifruoja pranešimą m atvaizduodamas jį į skaičių arba skaičių seką iš intervalo $[0, n - 1]$ ir atlikdamas šiuos veiksmus:

- Bronius pasiima Aldonos viešąjį raktą $PuK_A = e$.
- Jis apskaičiuoja šifrogramą $c = m^e \bmod n$ ir siunčia šifrogramą c Aldonai.

Aldona iššifruoja Broniaus pradinį pranešimą m atlikdama šiuos veiksmus:

- Naudojant savo slaptąjį raktą Aldona apskaičiuoja $c^d \bmod n = m^{ed} \bmod n = m^{(n)+1} \bmod n = m$.
- Gautą rezultatą Aldona atvaizduoja į pradinę tekstogramą.

Reikia pastebėti, kad Oilerio funkcija nėra optimalusis modulis, kuriuo gali būti redukuojami laipsniai. Dėl šios priežasties praktikoje vietoj šios funkcijos dažnai naudojama Karmaiklo funkcijos reikšmė $\lambda(n)$. Tai supaprastina atliekamus skaičiavimus, kadangi tokiu atveju naudojami laipsniai yra mažesni.

ElGamalio algoritmas yra naudojamas elektroniniams parašams ir asimetriniam šifravimui. Šis algoritmas remiasi DLU ir yra Difio-Helmano algoritmo pritaikymas minėtiems tikslams. El-Gamalio asimetrinio šifravimo protokolas atrodo taip [16]:

- Generuojami viešieji parametrai: didelis atsitiktinis pirminis skaičius p, kuris apibrėžia multiplikacinę grupę $\boldsymbol{Z}_p^*$ ir randamas šios grupės generatorius g.
- Aldona pasirenka atsitiktinį skaičių x ($2 \le x \le p - 1$) ir apskaičiuoja $a = g^x \bmod p$. Jos viešasis raktas $PuK_A = a$, o slaptasis raktas $PrK_A = x$.

Bronius užšifruoja pranešimą m atlikdamas šiuos veiksmus:

- Bronius pasirenka atsitiktinį skaičių y ($2 \le y \le p - 1$) ir apskaičiuoja $\gamma = g^y \bmod p$ ir $\delta = ma^y \bmod p$.
- Šifrograma (γ, δ) siunčiama Aldonai.

Aldona iššifruoja Broniaus pradinį pranešimą m atlikdama šiuos veiksmus:

- Naudojant savo slaptąjį raktą Aldona apskaičiuoja $\gamma^{-x} \bmod p$.
- Pradinis pranešimas $m = (\gamma^{-x})\delta \bmod p$.

ElGamalio asimetrinio šifravimo protokolas išsaugo visus Difio-Helmano protokolo reikalavimus parametrui *p*. Taip pat nagrinėdami ElGamalio šifravimo protokolą matome, kad gaunama šifrograma yra dvigubai didesnė už pradinį pranešimą. Protokolo privalumas yra naujo atsitiktinio skaičiaus *y* generavimas kiekvienam pranešimui *m*. To nepadarius žinant vieną pranešimą galima skaityti ir likusius [13].

Nuo praeito amžiaus devinto dešimtmečio pradėta vystyti kvantinių kompiuterių idėja. Tokios mašinos, nors ir hipotetinės dabar, ateityje galėtų žymiai pagreitinti kai kurių sunkiai spendžiamų klasikiniais kompiuteriais uždavinių sprendimą naudojant kvantinius skaičiavimus. 1996 metais Piteris W. Šoras straipsnyje [18] parodė, kad naudojant kvantinius kompiuterius skaičiaus faktorizacijos ir DLU uždaviniai gali būti išspręsti per polinominį laiką. Nors praėjus 18 metų nuo straipsnio paskelbimo tokie algoritmai kaip RSA ir El-Gamal yra aktyviai naudojami įvairiose sistemose, polinominis skaičių faktorizacijos ir diskretinio logaritmo uždavinių sprendimas reikštų, kad šios sistemos ateityje gali tapti nesaugios.

Reikia pabrėžti ir tai, kad nors iki šiol nagrinėtuose pavyzdžiuose platformai buvo naudojama baigtinė multiplikacinė skaičių grupė, yra ir kitų komutatyviosios kriptografijos algoritmų, kurie platformai naudoja kitas komutatyviasias grupes. Viena iš tokių plačiai naudojamų platformų yra elipsinės kreivės adicinė taškų grupė. Geometrijoje elipsinė kreivė yra trečios eilės kreivė, kurios kanoninė lygtis yra:

$$y^2 = x^3 + ax + b\,. \tag{3.1}$$

Kriptografijoje elipsinės kreivės dažnai yra apibrėžiamos virš lauko $\boldsymbol{Z_p}$. Ši platforma buvo panaudota Difio-Helmano algoritmui patobulinti. 1987 metais Nilas Koblicas savo staipsnyje „Elipsinių kreivių kriptosistemos“ (angl. Elliptic Curve Cryptosystems) [19] parodė kaip šią platformą galima pritaikyti El-Gamalio asimetrinio šifravimo protokolui realizuoti.

Taip vadinamų elipsinių kreivių kriptosistemų pagrindinis privalumas yra trumpesni raktai. Taip pat, kadangi šios sistemos yra dažnai taikomos praktikoje, JAV Nacionalinis Standartų ir Technologijų Institutas (angl. National Institute of Standarts and Technology – NIST) įtraukė šias sistemas į rekomenduojamų duomenų apsaugos sistemų sąrašą [20]. Vienas iš tokio taikymo pavyzdžių yra elipsinių kreivių kriptografija paremtas ir šiuo metu galiojantis Rusijos Federacijos elektroninio parašo standartas ГОСТ Р 34.10-2012 [21]. Pagal 2013 metais paskelbtus standartus FIPS PUB 186-4 yra rekomenduojama pasirinkti elipsinės kreivės parametro *p* 192, 224, 256, 384 arba 521 bitų reikšmę [20]. Taip pat kiekvienai rekomenduojamai *p* reikšmei yra nurodytas kitų sistemos parametrų skaičiavimo algoritmas. Tačiau, nors elipsinių kreivių DLU yra sudėtingesnis už įprastą DLU, naudojant Šoro algoritmą šis uždavinys gali būti efektyviai išspręstas per polinominį laiką naudojant kvantinius kompiuterius. Kadangi kriptografiniai protokolai elipsinių kreivių pagrindu yra greiti, tai savo protokolą palyginsime su elipsinių kreivių asimetrinio šifravimo protokolu. Tačiau elipsinių kreivių kriptografija nėra pagrindinis šio darbo tyrimo objektas. Informaciją apie elipsines kreives galima rasti [13] ir [19] šaltiniuose.

Literatūroje galima rasti ir kitų Difio-Helmano protokolo praplėtimų. 2002 metais JAV kriptografas Kristoferis Monico pasiūlė panaudoti kvadratinių *m*-tos eilės

matricų virš baigtinės skaitinės grupės $\boldsymbol{Z_p}$ multiplikacinę pusgrupę $\boldsymbol{M}_m(\boldsymbol{Z_p})$ platformai apibrėžti [22]. Jis apibrėžė šios pusgrupės veiksmą baigtinėje vektorinėje komutatyvioje grupėje $\boldsymbol{G}^m$, kurios galia $|\boldsymbol{G}^m| = p$, t.y vaizdavimą $\boldsymbol{M}_m(\boldsymbol{Z_p}) \times \boldsymbol{G}^m \rightarrow \boldsymbol{G}^m$ tokiu būdu

$$\left(A_{m\times m}, X_{m\times 1}\right) \rightarrow Y_{m\times 1}, \tag{3.2}$$

čia $A_{m\times m} \in \boldsymbol{M}_m(\boldsymbol{Z_p})$ yra m-tos eilės kvadratinė matrica, kurios elementai $a_{ij} \in \boldsymbol{Z_p}$, o $X_{m\times 1}, Y_{m\times 1} \in \boldsymbol{G}^m$. Vektoriaus Y elementai y_i yra apskaičiuojami pagal formulę

$$y_i = \prod_{j=1}^{m} x_j^{a_{ij}} \tag{3.3}$$

Naudojant šį veiksmą Monico realizavo Difio-Helmano schemą [22]. Tačiau nepaisant atliktų pakeitimų autoriaus pasiūlyto protokolo saugumas, nors ir priklauso nuo matricų eilės ir skaitinės grupės dydžio, vis tiek remiasi DLU sudėtingumu, o tai reiškia, kad ir šis protokolas nėra atsparus Šoro kvantinei atakai. Svarbu pastebėti ir tai, kad šis protokolas negali būti priskirtas prie nekomutatyvios kriptografijos primityvų, kadangi nesiremia nei vienu iš nekomutatyviosios kriptografijos sunkių uždavinių, nors ir naudoja nekomutatyvią matricų pusgrupę. Taip yra dėl to, kad fiksavus matricą A yra formuojamas komutuojančių matricų popusgrupis.

Taigi matome, kad šiame skyrelyje nagrinėti komutatyviosios kriptografijos protokolai nėra atsparūs kvantinei kriptoanalizei, kadangi jie remiasi diskretinio logaritmo arba sudėtinio skaičiaus faktorizacijos uždaviniais, kurie gali būti išspręsti naudojant Šoro kvantinį algoritmą. Dar vienas tokių algoritmų trūkumas yra dideli skaičiavimo resursai, nes skaičiavimus reikia atlikti su dideliais sveikaisiais skaičiais.

Šiuolaikinių technologijų pasaulyje vis daugėja įrenginių, kurių energijos, atminties, skaičiavimo resursai yra riboti. Ribotų resursų įrenginių populiarumas mūsų kasdieniniame gyvenime lemia poreikį turėti ne tik saugius bet ir efektyviai realizuojamus tokiuose įrenginiuose kriptografinius primityvus. Šiuolaikiniai kriptografiniai primityvai, kurie gali būti sėkmingai realizuoti ribotų resursų sistemoje turi būti greiti, naudoti mažai atminties ir taupyti įrenginio energiją. Straipsniuose [23] – [25] autoriai įvertino klasikinių protokolų (Difio-Helmano raktų apsikeitimas, RSA šifravimas, elipsinių kreivių šifravimas) realizaciją ribotų resursų sistemose. Rezultatai parodė, jog tokiose sistemose efektyviausiai realizuojami algoritmai, kurie yra paremti elipsinių kreivių kriptografija. Elipsinių kreivių kriptosistemos turi pranašumą prieš ElGamalio ir RSA protokolus (t.y. šios kriptosistemos yra greitesnės), nes naudojami skaičiai yra žymiai mažesni, nors aritmetiniai veiksmai, kurie yra atliekami su elipsinės kreivės taškais, yra sudėtingesni. Dėl šios priežasties, dauguma šiuolaikinių įrenginių duomenims šifruoti arba pasirašyti naudoja elipsinių kreivių kriptografiją.

3.2. Nekomutatyvioji kriptografija

Ieškant naujų efektyvių kriptografinių primityvų XX amžiaus pabaigoje buvo paskelbti pirmieji kriptografiniai algoritmai, kurie naudoja nekomutatyviasias algebrines struktūras. Taip atsirado nekomutatyvioji kriptografija. Vienas iš pagrindinių šios kriptografijos šakos tikslų yra tokių naujų vienkrypčių funkcijų, kurių apgręžimo uždavinys būtų tokio paties sudėtingumo, kaip ir kokio nors NP-pilnojo uždavinio sudėtingumas, sukūrimas. Tuo atveju tokio kriptografinio primityvo kriptoanalizė taptų ekvivalentiška NP-pilnojo uždavinio sprendimui, t.y. neįveikiama tradiciniams kompiuteriams. Kadangi iki šiol nėra žinomi efektyvūs kvantinių skaičiavimų metodai, leidžiantys spręsti NP-pilnuosius uždavinius, yra manoma, kad nekomutyviosios kriptografijos kryptis yra perspektyvi.

Idėja pritaikyti nekomutatyviasias grupes kriptografijoje atsirado 1985 metais, kai Nilas R. Vagneris ir Mariana R. Magyarik paskelbė straipsnį [26], kuriame pasiūlė asimetrinį raktų apsikeitimo protokolą, kurio saugumas buvo paremtas žodžio uždavinio baigtinėje grupėje sudėtingumu. Šis uždavinys yra vienas iš kombinatorinės grupės teorijos sprendžiamumo uždavinių, kuris formuluojamas taip: turint du žodžius, sudarytus iš grupės generatorių, reikia nustatyti ar šie žodžiai atitinka tą patį grupės elementą. Galutinis šio uždavinio atsakymas pateikiamas formoje „taip“ arba „ne“.

Dauguma iki šiol pasiūlytu nekomutatyviosios kriptografijos protokolų remiasi paieškos uždaviniais, kurie yra tradicinių kombinatorinės grupės teorijos nagrinėjamų sprendžiamumo uždavinių variantai. Vienas iš galimų tokio uždavinio pavyzdžių yra jungtinio elemento paieškos uždavinys (angl. conjugacy search problem). Tarkime, turime nekomutatyvią grupę $\boldsymbol{G}$, kurioje yra sprendžiamas žodžio uždavinys. Tada galime suformuluoti šiuos du uždavinius [27]:

- Jungtinio elemento sprendžiamumo uždavinys: žinomi du grupės $\boldsymbol{G}$ elementai a ir b. Ar egzistuoja toks elementas $x \in \boldsymbol{G}$, kad

$$b = x^{-1}ax. \tag{3.4}$$

- Jungtinio elemento paieškos uždavinys (JPU): žinomi du grupės $\boldsymbol{G}$ elementai a ir b. Be to grupėje $\boldsymbol{G}$ egzistuoja elementas x toks, kad elementai a ir b yra susieti (3.4) sąryšiu. Rasti bent vieną tokį elementą x.

JPU vaidina svarbų vaidmenį sudėtingumo teorijoje, tačiau yra neįdomus grupių teorijos specialistams, nes, jeigu toks elementas x egzistuoja, tai galima perrinkti visus žodžius, turinčius (3.4) pavidalą ir palyginti juos su b, kol bus rastas tinkamas variantas. Tačiau toks tiesioginis JPU sprendimas yra praktiškai neefektyvus, nes šio uždavinio sprendimo laikas eksponentiškai priklauso nuo žodžio b ilgio. Jeigu nežinomi kiti JPU sprendimo būdai, efektyvesni už minėtąjį, tai vaizdavimą

$$x \to x^{-1}ax \tag{3.5}$$

galime laikyti vienkrypte funkcija. Reikia pabrėžti, kad atvirkštinio elemento paieškos uždavinys gali būti išspręstas per polinominį laiką. Dažnai yra žinomi teoriniai šio uždavinio sprendimo būdai. Taigi vaizdavimo

$$x \to x^{-1} \tag{3.6}$$

negalime laikyti vienkrypte funkcija

Crypto 2000 konferencijoje Korėjos mokslininkai pristatė raktų apsikeitimo protokolą, kurio saugumas remiasi JPU sudėtingumu. Jų pasiūlytas protokolas yra toks [28]:

- Skelbiamas viešasis parametras $\omega \in \boldsymbol{G}$.
- Aldona pasirenka atsitiktinį elementą $x \in \boldsymbol{G}$ ir apskaičiuoja $a = x^{-1}\omega x$. Jos slaptasis raktas $PrK_A = x$, o viešasis raktas $PuK_A = a$. Viešąjį raktą Aldona persiunčia Broniui.
- Bronius pasirenka atsitiktinį elementą $y \in \boldsymbol{G}$ ir apskaičiuoja $b = y^{-1}\omega y$. Jo slaptasis raktas $PrK_B = y$, o viešasis raktas $PuK_B = a$. Viešąjį raktą Bronius persiunčia Aldonai.
- Aldona apskaičiuoja raktą $K_A = x^{-1}bx = x^{-1}y^{-1}\omega yx$, o Bronius – raktą $K_B = y^{-1}ay = y^{-1}x^{-1}\omega xy$. Jeigu elementai x ir y yra pasirinkti taip, kad $xy = yx$, tai $K_A = K_B$. Šį bendrą raktą galime pažymėti K, t.y. $K = K_A = K_B$.

Korėjos mokslininkai pasiūlė naudoti kasų grupę $\boldsymbol{B_n}$ platforminei grupei $\boldsymbol{G}$ apibrėžti, t.y. $\boldsymbol{G} = \boldsymbol{B_n}$, nes kasų grupės turi natūraliai komutuojančius pogrupius. Literatūroje galima rasti ir žymėjimą $a^x = x^{-1}ax$, kurį galima paaiškinti tuo, kad jungtinumo funkcijos savybės yra panašios į įprasto kėlimo laipsniu savybes. Jei šį žymėjimą pritaikysime Korėjos mokslininkų protokolui, gausime protokolą, kuris yra panašus į Difio-Helmano protokolą. Taigi galima teigti, kad šis nekomutatyviosios kriptografijos protokolas yra Difio-Helmano protokolo analogas.

Kitas protokolas, kuris buvo paskelbtas 1999 metais išsiskiria iš nekomutatyviosios kriptografijos protokolų tuo, kad nekelia jokių komutatyvumo reikalavimų ir gali platformai naudoti bet kokią nekomutatyvią grupę, kurioje yra išsprendžiamas žodžio uždavinys. Protokolo autoriai Irisa Anšelė, Maiklas Anšelis ir Dorianas Goldfeldas panaudojo nekomutatyvios grupės komutatoriaus sąvoką. Jų protokolas atrodo taip [29]:

- Aldonai ir Broniui viešai priskiriama po vieną nekomutatyviosios grupės $\boldsymbol{G}$ pogrupį $\boldsymbol{X} = \langle x_1, x_2, \ldots, x_n \rangle$ (Aldonos pogrupis) ir $\boldsymbol{Y} = \langle y_1, y_2, \ldots, y_m \rangle$ (Broniaus pogrupis), t.y. šių pogrupių generatoriai yra atitinkamai $x_1, x_2, \ldots, x_n$ ir $y_1, y_2, \ldots, y_m$.

- Aldona pasirenka atsitiktinį elementą $a = y_{i_1} y_{i_2} \ldots y_{i_k}$ ir apskaičiuoja $\boldsymbol{A} = a^{-1}\boldsymbol{X}a = \left\langle a^{-1}x_1 a, a^{-1}x_2 a, \ldots, a^{-1}x_n a \right\rangle$. Jos slaptasis raktas $PrK_A = a$, o viešasis raktas $PuK_A = \boldsymbol{A}$. Viešąjį raktą Aldona persiunčia Broniui.
- Bronius pasirenka atsitiktinį elementą $b = x_{j_1} x_{j_2} \ldots x_{j_l}$ ir apskaičiuoja $\boldsymbol{B} = b^{-1}\boldsymbol{Y}b = \left\langle b^{-1}y_1 b, b^{-1}y_2 b, \ldots, b^{-1}y_m b \right\rangle$. Jo slaptasis raktas $PrK_B = b$, o viešasis raktas $PuK_B = \boldsymbol{B}$. Viešąjį raktą Bronius persiunčia Aldonai.
- Aldona, žinodama iš kokių generatorių sudarytas jos slaptasis raktas a, naudoja Broniaus viešąjį raktą $\boldsymbol{B}$ ir apskaičiuoja $b^{-1}ab$, daugindama atitinkamus aibės $\boldsymbol{B}$ elementus. Gautą elementą Aldona daugina iš a^{-1} iš kairės ir gauna raktą $K_A = a^{-1}b^{-1}ab$.
- Bronius, naudodamas Aldonos viešąjį raktą $\boldsymbol{A}$, apskaičiuoja elementą $a^{-1}ba$ ir jo atvikštinį elementą $\left(a^{-1}ba\right)^{-1} = a^{-1}b^{-1}a$. Gautą elementą $a^{-1}b^{-1}a$ Bronius iš dešinės padaugina iš b ir tokiu būdu gauna raktą $K_B = a^{-1}b^{-1}ab$, kuris yra elementų a ir b komutatorius. Kadangi $K_A = K_B$, tai Aldona ir Bronius turi bendrą raktą, kurį galime pažymėti K, t.y. $K = K_A = K_B$.

Šio protokolo pagrindinis trūkumas yra viešųjų raktų ilgiai. Kadangi reikia transliuoti tiek elementų, kiek viešasis raktas turi generatorių, tai esant dideliam generatorių kiekiui šis procesas gali būti imlus laiko sąnaudų ir atminties prasme.

Pastebėkime, kad norint apskaičiuoti Aldonos slaptąjį raktą a reikia spręsti JPU sistemą. 2004 metais rusų kilmės kriptografai Vladimiras Špilrainas ir Aleksandras Ušakovas paskelbė straipsnį [30], kuriame atliko Korėjos mokslininkų ir Anšelio-Anšelės-Goldfeldo protokolų palyginimą ir parodė, kad JPU spręsti nebūtina ir nepakankama. Autoriai nustatė, kad norint nulaužti Ko-Lee protokolą vietoj JPU galima spręsti kitą uždavinį. Būtent, galima rasti tokius elementus x_1 ir x_2, kurie tenkina lygybę

$$x^{-1}\omega x = x_1 \omega x_2 = a \tag{3.7}$$

Šis uždavinys vadinamas *dekompozicijos uždaviniu*. Be to šie elementai turi komutuoti su elementu y. Reikia pabrėžti, kad, nors kenkėjas ir nežino pačio y, tačiau pogrupis, iš kurio jis yra pasirenkamas yra žinomas. Taigi elementai x_1 ir x_2 turi komutuoti su visais šio pogrupio elementais. Špilrainas iš Ušakovas parodė [30], kad užtenka nustatyti vieną tokią elementų porą, t.y. galima nulaužti tik Aldonos viešąjį raktą, arba tik Broniaus viešąjį raktą, kadangi

$$x_1 b x_2 = x_1 y^{-1}\omega y x_2 = y^{-1}x_1 \omega x_2 y = y^{-1}x^{-1}\omega x y = K \tag{3.8}$$

Aišku, kad JPU yra atskiras dekompozicijos uždavinio atvejis. Tačiau rasti dekompozicijos uždavinio sprendinį yra paprasčiau, negu JPU, kadangi turime du

nežinamuosius x_1 ir x_2 vietoj vieno nežinamojo x. Rasti bet kokią porą $(x_1;x_2)$ yra nesudėtinga (pvz. $(1;\omega^{-1}a)$), tačiau nustatyti, ar abu šie elementai komutuoja su y gali būti nelengva.

Anšelio-Anšelės-Goldfeldo protokolas yra pranašesnis tuo, kad kenkėjas, net ir išsprendęs JPU lygčių sistemą (t.y. gavęs slaptąjį raktą), vis tiek negali nulaužti bendrojo Aldonos ir Broniaus rakto K, kadangi jam trūksta informacijos apie tai, kaip sudaryti elementą $b^{-1}ab$ iš aibės $\boldsymbol{B} = \langle b^{-1}y_1b, b^{-1}y_2b, \ldots, b^{-1}y_mb \rangle$ elementų [27]. Tai reiškia, kad kenkėjas turi spręsti papildomą priklausomumo paieškos uždavinį, t.y. žinodamas Aldonos slaptąjį raktą a ir aibės $\boldsymbol{X}$ generatorius kenkėjas turi nustatyti kaip elementas a yra išreiškiamas aibės $\boldsymbol{X}$ elementais. Tačiau net ir priklausomumo sprendžiamumo uždavinys, t.y. nustatyti ar galima elementą a išreikšti aibės $\boldsymbol{X}$ elementais, gali būti sudėtingas ir, kai kuriose grupėse, netgi neišsprendžiamas.

XXI amžiaus pradžioje buvo publikuota daug mokslinių straipsnių, kuriuose buvo analizuojamas Anšelio-Anšelės-Goldfeldo protokolo saugumas, kai platformai apibrėžti buvo naudojamos kasų grupės. Buvo aprašyti algoritmai, kurie leidžia spręsti JPU arba jų sistemas kasų grupėse [31], [32]. Straipsnyje [31] šiam tikslui buvo panaudotas kasų grupės generatorių vaizdavimas į spalvotas Burau matricas. Naudojant šiuos algoritmus autoriai nulaužė bendrąjį Aldonos ir Broniaus raktą K prie tų parametrų reikšmių, kurias pasiūlė Anšelio-Anšelės-Goldfeldo protokolo autoriai. 2007 metais Ušakovas kartu su Miasnikovu Anšelio-Anšelės-Goldfeldo protokolui pritaikė žodžių ilgiais paremtą ataką (angl. Length based attack) [33], kurią 2002 metais pasiūlė Dž. Hjusas ir A. Tanenbaumas [34].

Taigi matome, kad norint sukurti saugų kriptografinį primityvą svarbu pasirinkti tinkamą algebrinę struktūrą platformai apibrėžti. 2008 metų knygoje [27] jos autoriai suformulavo pagrindinius reikalavimus kriptografinio primityvo platforminiai grupei:

- Platforminė grupė turi buti plačiai žinoma ir gerai išnagrinėta.
- Žodžio uždavinys platforminėje grupėje turi buti greitai sprendžiamas (priklausomybė nuo laiko tiesinė arba kvadratinė).
- Turi būti įmanoma paslėpti grupės elementus taip, kad jų negalima būtų atstatyti. Pavyzdžiui, žinant dviejų grupės elementų x ir y sandaugą xy negalima per priimtiną laiką rasti šių elementų (skaičių faktorizacijos uždavinys).
- Platforminės grupės augimas yra eksponentinis arba didesnis. Tai reiškia, kad elementų, kurių ilgis yra n (čia ilgis gali būti suprantamas, pavyzdžiui, žodžių prasme), kiekis grupėje turi augti greičiau už bet kokį daugianarį nuo n. Šis reikalavimas neleidžia organizuoti atakų, paremtų mažu raktų aibės dydžiu.

Viena iš tokių grupių yra matricų grupė virš baigtinio lauko arba žiedo. Naudojant šią grupę ir matricų savybes buvo sukurti kriptografiniai protokolai, kurių saugumas yra paremtas sunkių uždavinių matricų (pus)grupėje arba žiede sudėtingumu.

2007 metais Džerardas Mazė, Kristoferis Monico ir Žoakimas Rosentalis pasiūlė raktų apsikeitimo protokolą, kurio saugumas yra ekvivalentus pusgrupės veiksmo

uždaviniui (angl. semigroup action problem) [35]. Prieš pateikiant protokolą paminėsime, kad nekomutatyvaus pusžiedžio $\boldsymbol{R}$ centrą $\boldsymbol{C}$ sudaro pusžiedžio $\boldsymbol{R}$ elementai, kurie komutuoja su visais likusiais šio pusžiedžio elementais daugybos operacijos atžvilgiu.

- Aldona ir Bronius susitaria dėl viešojo baigtinio pusžiedžio $\boldsymbol{R}$ (bendruoju atveju nekomutatyvaus), kurio centras yra netuščias ir nėra kokio nors lauko dalis. Taip pat jie susitaria dėl kvadratinių m-tos eilės matricų grupės $\boldsymbol{M}_m(\boldsymbol{R})$ ir viešųjų matricų M_1, M_2, $S \in \boldsymbol{M}_m(\boldsymbol{R})$.
- Aldona atsitiktinai pasirenka du daugianarius p_a, $q_a \in \boldsymbol{C}[t]$, čia $\boldsymbol{C}[t]$ yra daugianarių, kurių koeficientai priklauso centrui $\boldsymbol{C}$, aibė. Naudojant šių daugianarių porą Aldona apskaičiuoja slaptąjį raktą $PrK_A = \{p_a(M_1), q_a(M_2)\}$ ir matricą $A = p_a(M_1) \cdot S \cdot q_a(M_2)$. Savo viešąjį raktą $PuK_A = A$ Aldona siunčia Broniui.
- Bronius atsitiktinai pasirenka du daugianarius p_b, $q_b \in \boldsymbol{C}[t]$. Naudojant šių daugianarių porą Bronius apskaičiuoja slaptąjį raktą $PrK_B = \{p_b(M_1), q_b(M_2)\}$ ir matricą $B = p_b(M_1) \cdot S \cdot q_b(M_2)$. Savo viešąjį raktą $PuK_A = B$ Aldona siunčia Broniui.
- Aldona apskaičiuoja $K_A = p_a(M_1) \cdot B \cdot q_a(M_2)$, o Bronius - $K_B = p_b(M_1) \cdot A \cdot q_b(M_2)$. Kadangi matricos $p_a(M_1)$ ir $p_b(M_1)$ yra gaunamos apskaičiuojant daugianarius nuo tos pačios matricos M_1, tai šios matricos komutuoja. Tas pats galioja ir matricoms $q_a(M_2)$ ir $q_b(M_2)$. Taigi Aldona ir Bronius susitaria dėl bendrojo rakto $K = K_A = K_B$.

Matome, kad šiuo atveju bendras raktas K yra gaunamas naudojant matricų žiedo savybę, kad du daugianariai, kurie buvo apskaičiuoti naudojant tą pačią matricą komutuoja. Šią savybę yra patogu naudoti formuojant natūraliai komutuojančius matricų poaibius.

2007 metais Lietuvos mokslininkai Eligijus Sakalauskas, Povilas Tvarijonas ir Andrius Raulynaitis savo straipsnyje pasiūlė protokolą, kuris šiuo metu yra žinomas kaip STR protokolas (pagal pirmąsias autorių pavardžių raides). Šį protokolą galima trumpai charakterizuoti taip [36]:

- Aldona ir Bronius susitaria dėl kvadratinių m-tos eilės matricų žiedo $\boldsymbol{M}$, viešosios matricos $Q \in \boldsymbol{M}$ ir dviejų matricų poaibių $\boldsymbol{G_1}$ ir $\boldsymbol{G_2}$, kurių elementai (matricos) komutuoja tarpusavyje, t.y. $AB = BA$, jei $A \in \boldsymbol{G_1}$ ir $B \in \boldsymbol{G_2}$.
- Aldona atsitiktinai pasirenka neišsigimusią matrica $A \in \boldsymbol{G_1}$ ir natūralųjį skaičių r. Ši informacija sudaro Aldonos slaptąjį raktą, t.y. $PrK_A = \{A, r\}$. Jos viešasis raktas $PuK_A = AQ^rA^{-1} = R$. Viešąjį raktą Aldona persiunčia Broniui.
- Bronius atsitiktinai pasirenka neišsigimusią matrica $B \in \boldsymbol{G_2}$ ir natūralųjį skaičių s. Broniaus slaptas raktas $PrK_B = \{B, s\}$, o viešasis raktas – $PuK_B = BQ^sB^{-1} = S$. Viešąjį raktą Bronius persiunčia Aldonai.
- Aldona apskaičiuoja $K_A = AS^rA^{-1} = ABQ^{sr}B^{-1}A^{-1}$, o Bronius – $K_B = BR^sB^{-1} = BAQ^{rs}A^{-1}B^{-1}$. Kadangi $AB = BA$ ir $Q^{sr} = Q^{rs}$, tai Aldona ir Bronius susitaria dėl bendrojo rakto $K = K_A = K_B$.

Aišku, kad galima pasirinkti poaibius $\boldsymbol{G_1}$ ir $\boldsymbol{G_2}$ taip, kad $\boldsymbol{G_1} = \boldsymbol{G_2} = \boldsymbol{G}$. Poaibiui $\boldsymbol{G}$ formuoti patogu naudoti daugianarius nuo viešai žinomos matricos $\boldsymbol{M}$, dėl kurios Aldona ir Bronius turi susitarti iš anksto.

Galima pastebėti, kad kadangi JPU sprendimo algoritmai matricų žiede yra žinomi, tai STR protokolo saugumas negali būti paremtas šiuo uždaviniu. Straipsnio [24] autoriai nagrinėdami protokolo saugumą nustatė, kad norint nulaužti vartotojo (Aldonos arba Broniaus) slaptą raktą reikia spręsti ne tik JPU, bet ir matricinę DLU versiją.

2010 metais italų mokslininkai Vitorio Otavianis, Alberto Zanonis ir Masimo Regolis paskelbė straipsnį [24], kuriame STR protokolo realizacija buvo palyginta su Difio-Helmano protokolų (klasikinio ir elipsinių kreivių) realizacija. Rezultatai parodė, kad STR protokolas yra kelis kartus greitesnis už savo konkurentus. Tačiau nagrinėdami STR protokolo saugumą mokslininkai pastebėjo protokolo silpnąją vietą. Būtent, jeigu matricos Q determinantas $\det(Q) \neq 1$, tai galima surasti matricos Q laipsnį r arba s apskaičiavus vartotojo viešojo rakto determinantą ir išsprendus klasikinį DLU. Šią idėją savo straipsnyje praplėtė prancūzų mokslininkas Muhamedas Eftekharis [37]. Jo ataka yra paremta tikrinių matricos Q reikšmių invariantiškumo JPU atžvilgiu savybe.

2007 metais buvo pasiūlyta nauja vienkryptė funkcija, kuri buvo apibrėžta matricų multiplikacinėje grupėje ir pavadinta matricinio laipsnio funkcija. Ši funkcija buvo aprašyta Eligijaus Sakalausko ir Kęstučio Lukšio straipsnyje [38] ir panaudota S blokams sudaryti. 2008 metais E. Sakalauskas kartu su Narimantu Listopadskiu ir Povilu Tvarijonu pasiūlė panaudoti šią funkciją raktų apsikeitimo protokolui sukurti. Autorių pasiūlytas protokolas atrodo taip [39]:

- Aldona ir Bronius susitaria dėl multiplikacinės kvadratinių m-tos eilės matricų grupės $\boldsymbol{M}$, viešosios matricos $Q \in \boldsymbol{M}$ ir komutuojančių matricų poaibių $\boldsymbol{R}_L$ ir $\boldsymbol{R}_R$.
- Aldona atsitiktinai pasirenka matricas $X \in \boldsymbol{R}_L$ ir $Y \in \boldsymbol{R}_R$ ir naudodama matricinio laipsnio funkcija apskaičiuoja matricą $A = {}^{X}Q^{Y}$. Jos slaptasis raktas $PrK_A = \{X, Y\}$, o viešasis raktas $PuK_A = A$. Viešąjį raktą Aldona persiunčia Broniui.
- Bronius atsitiktinai pasirenka matricas $U \in \boldsymbol{R}_L$ ir $V \in \boldsymbol{R}_R$ ir naudodamas matricinio laipsnio funkcija apskaičiuoja matricą $B = {}^{U}Q^{V}$. Jo slaptasis raktas $PrK_B = \{U, V\}$, o viešasis raktas $PuK_B = B$. Viešąjį raktą Bronius persiunčia Aldonai.
- Aldona apskaičiuoja $K_A = {}^{X}B^{Y} = {}^{XU}Q^{VY}$, o Bronius - $K_B = {}^{U}A^{V} = {}^{UX}Q^{YV}$. Kadangi matricų poros X ir U bei Y ir V priklauso komutuojančių matricų poaibiams $\boldsymbol{R}_L$ ir $\boldsymbol{R}_R$, tai $XU = UX$ ir $YV = VY$. Taigi Aldona ir Bronius susitaria dėl bendrojo rakto $K = K_A = K_B$.

Šis protokolas, kaip ir Ko-Lee protokolas, yra panašus į Difio-Helmano protokolą ir yra jo analogas kai yra naudojama maticinio laipsnio funkcija. Pagrindinis šio protokolo privalumas yra tai, kad nereikia atlikti skaičiavimų su dideliais skaičiais, o tai reiškia, kad aritmetiniai veiksmai atliekami greičiau.

2013 metais Kęstutis Lukšys apginė daktaro disertaciją tema „Matricinio laipsnio šifras ir jo analizė“ [40]. Savo darbe Lukšys pateikė matricinio laipsnio funkcijos apibrėžimą bei įrodė pagrindines šios funkcijos savybes. Naudojant matricinio laipsnio funkciją Lukšys suformavo S blokus, kuriuos pritaikė simetrinio šifravimo protokolui sudaryti. Jo rezultatai parodė, kad jau pirmoji matricinio laipsnio simetrinis šifro iteracija yra atspari pilnojo perrinkimo atakai, jeigu tinkamai parinkti šifro pagrindiniai parametrai. Didėjant iteracijų skaičiui šifro atsparumas didėja.

2012 metų straipsniuose [41] ir [42] autoriai palygino matricinio laipsnio funkcija paremtų protokolų realizaciją ribotų resursų sistemose su klasikiniais protokolais. Buvo nustatyta, jog autorių pasiūlyti protokolai yra greitesni ir realizuojami efektyviau už elipsinių kreivių protokolus.

3.3. Išvados

Apibendrinant skyriuje pateiktą literatūros apžvalgą galima padaryti tokias išvadas:

- Komutatyviosios kriptografijos protokolai nėra atsparūs kvantinei kriptoanalizei, todėl atsiradus kvantiniams kompiuteriams gali tapti nesaugūs.
- Klasikiniai komutatyviosios asimetrinės kriptografijos primityvai naudoja ilgus slaptuosius ir viešuosius raktus. Dėl šios priežasties naudojant klasikinius protokolus, paremtus DLU sudėtingumu neišvengiamai susiduriama su itin didelių skaičių problema, t.y. reikia atlikti aritmetinius veiksmus su labai dideliais skaičiais. Nors tokiems veiksmams atlikti yra sukurti specialūs algoritmai, tačiau skaičiavimo sąnaudos neleidžia efektyviai realizuoti tokių protokolų ribotų resursų sistemose.
- Elipsinių kreivių kriptografija turi pranašumą prieš DLU paremtus protokolus, kadangi naudojami raktai yra žymiai trumpesni, nepaisant to, kad aritmetiniai veiksmai su elipsinės kreivės taškais yra sudėtingesni.
- Nekomutatyviosios kriptografijos protokolai, kurie remiasi JPU sudėtingumu (pvz., Ko-Lee protokolas), gali būti nulaužti sprendžiant žymiai lengvesnį dekompozicijos uždavinį.
- Anšelio-Anšelės-Goldfeldo protokolas yra pranašesnis už Ko-Lee protokolą tuo, kad kenkėjas, net ir išsprendęs JPU lygčių sistemą (t.y. gavęs slaptąjį raktą), vis tiek negali nulaužti bendrojo Aldonos ir Broniaus rakto K, kadangi jam trūksta informacijos apie tai, kaip formuojamas slaptasis raktas.
- Protokolų, paremtų matricinio laipsnio funkcijos sudėtingumu, analizė rodo, kad jie yra greitesni už klasikinius komutatyviosios kriptografijos protokolus ir gali būti efektyviai realizuoti ribotų resursų sistemose. Tai reiškia, kad šiuos kriptografinius primityvus yra perspektyvu taikyti praktikoje, jeigu yra patenkinti primityvo saugumo reikalavimai.
- Kūriant naujus kriptografinius primityvus svarbu tinkamai pasirinkti algebrinę platformą nagrinėjamam uždaviniui spręsti, kadangi protokolas gali būti nulaužtas ir dėl netinkamų platforminės struktūros algebrinių arba statistinių savybių.

4. MATRICINIO LAIPSNIO FUNKCIJA

4.1. Matricinio laipsnio funkcijos apibrėžimas ir algebrinės savybės

Žymėkime $\boldsymbol{M_A}$ kvadratinių m-tos eilės matricų žiedą, kai matricų elementai priklauso aibei $\boldsymbol{A}$, kuri sudėties ir daugybos atžvilgiu sudaro asociatyvų žiedą. Matricų sudėtis ir daugyba šiame žiede apibrėžiamos standartiniu būdu. Šiame skyrelyje įvesime dar vieną matricinę operaciją, kurią vadinsime *matricinio laipsnio funkcija* (MLF).

MLF idėja yra panaši į dviejų matricų daugybą, tačiau yra paremta kėlimo laipsniu ir daugybos operacijomis. Formaliai galima teigti, kad MLF praplečia kėlimo laipsniu operaciją į nekomutatyvią matricų pusgrupę. Tačiau nuo įprasto kėlimo laipsniu MLF skiriasi tuo, kad šiuo atveju kėlimo laipsnis taip pat yra matrica. Dėl šios priežasties MLF saugumas nėra paremtas klasikiniu DLU.

MLF buvo plačiai nagrinėjama Kęstučio Lukšio disertacijoje [40]. Taip pat kai kurios šios funkcijos savybės ir taikymai aprašyti [43] ir [44]. Formaliai šią funkciją galima apibrėžti bet kokio formato matricoms. Tačiau, kadangi funkcijos idėja yra panaši į matricinę daugybą, šių matricų formatai taip pat turi būti suderinti. Dėl šios priežasties funkcijai apibrėžti mes naudojame kvadratines matricas, kadangi tokiu atveju formatų suderinamumo sąlyga yra išpildoma automatiškai. Be to funkcijos reikšmės – naujos matricos formatas sutampa su argumentų formatu.

MLF turi vieną parametrą – matricą Q, ir vieną arba du argumentus – matricas X ir Y, priklausomai nuo to, ar funkcija yra vienpusė ar dvipusė. Bendruoju atveju matricos Q elementai gali priklausyti bet kokiai multiplikacinei komutatyviai pusgrupei $\boldsymbol{S}$, o matricų X ir Y elementai turi būti pasirenkami iš skaitinio žiedo $\boldsymbol{R}$. Kvadratinių m-tos eilės matricų pusgrupę, kuri yra apibrėžta virš pusgrupės $\boldsymbol{S}$, žymėsime $\boldsymbol{M_S}$ ir vadinsime ją *platformine pusgrupe*. Kvadratinių m-tos eilės matricų žiedą, kuris yra apibrėžtas virš skaitinio žiedo $\boldsymbol{R}$, žymėsime $\boldsymbol{M_R}$ ir vadinsime jį *laipsniniu žiedu*. Šis žiedas yra pasirenkamas pagal platrofminės pusgrupės savybes. MLF rezultatas priklauso platforminei pusgrupei.

Pradėsime nuo vienpusių MLF. Tarkime, turime dvi kvadratines m-tos eilės matricas Q ir Y. Tada galima apibrėžti *MLF iš dešinės*, kurios rezultatas yra matrica C. Šį veiksmą žymėsime taip:

$$C = Q^Y. \tag{4.1}$$

Matricos C elementai c_{ij} yra apskaičiuojami naudojant formulę [43]:

$$c_{ij} = \prod_{k=1}^{m} q_{ik}^{\,y_{kj}}. \tag{4.2}$$

Matricą Q (4.1) išraiškoje vadinsime *platformine matrica*, matricą Y – *matriciniu laipsniu* arba *laipsnine matrica*, o matricą C – *dešiniąja matricinio laipsnio* (ML) *eksponente*. Tokiu atveju mes sakysime, kad matricą Q pakėlėme matriciniu laipsniu Y iš dešinės.

Pavyzdys. Tarkime, kad platforminė pusgrupę sudaro antros eilės kvadratinės matricos, kurių elementai priklauso grupei $\boldsymbol{Z}_7^*$. Tada laipsninį žiedą sudaro antros eilės kvadratinės matricos, kurių elementai priklauso žiedui $\boldsymbol{Z}_6$. Taigi platforminės matricos Q elementai pasirenkami iš $\boldsymbol{Z}_7^*$, o laipsninės matricos Y elementai priklauso žiedui $\boldsymbol{Z}_6$. Tarkime, kad šios matricos yra lygios:

$$Q = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix},\ Y = \begin{pmatrix} 3 & 2 \\ 5 & 4 \end{pmatrix}$$

Tada turime

$$C = \begin{pmatrix} 2^3 \cdot 3^5 & 2^2 \cdot 3^4 \\ 4^3 \cdot 5^5 & 4^2 \cdot 5^4 \end{pmatrix} = \begin{pmatrix} 1 \cdot 5 & 4 \cdot 4 \\ 1 \cdot 3 & 2 \cdot 2 \end{pmatrix} = \begin{pmatrix} 5 & 2 \\ 3 & 4 \end{pmatrix}.$$

Visi kėlimai laipsniu ir visos daugybos yra atliekamos moduliu 7.

Analogiškai yra apibrėžiama ir *MLF iš kairės*. Tarkime, turime dvi kvadratines m-tos eilės matricas Q ir X. MLF iš kairės žymėsime

$$D = {}^{X}Q, \tag{4.3}$$

o kairiosios ML eksponentės D elementus apskaičiuosime pagal formulę [43]:

$$d_{ij} = \prod_{k=1}^{m} q_{kj}^{x_{ik}} \tag{4.4}$$

Šiuo atveju sakysime, kad matricą Q pakėlėme matriciniu laipsniu X ir kairės.

Pavyzdys. Nagrinėkime tas pačias algebrines struktūras, kaip ir anksčiau. Tarkime, turime šias matricas X ir Q:

$$X = \begin{pmatrix} 3 & 2 \\ 5 & 4 \end{pmatrix},\ Q = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$$

Tada turime

$$D = \begin{pmatrix} 2^3 \cdot 4^2 & 3^3 \cdot 5^2 \\ 2^5 \cdot 4^4 & 3^5 \cdot 5^4 \end{pmatrix} = \begin{pmatrix} 1 \cdot 2 & 6 \cdot 4 \\ 4 \cdot 4 & 5 \cdot 2 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 2 & 3 \end{pmatrix}.$$

Pastebėkime, kad kairioji ir dešinioji MLF tenkina šias lygybes:

$$Q^{(XY)} = \left(Q^X\right)^Y; \tag{4.5}$$

$${}^{(XY)}Q = {}^X\left({}^YQ\right); \tag{4.6}$$

$$\left(Q^X\right)^{X^{-1}} = Q^{XX^{-1}} = Q^I = Q; \tag{4.7}$$

$${}^{X^{-1}}\left({}^XQ\right) = {}^{X^{-1}X}Q = {}^IQ = Q, \tag{4.8}$$

čia XY yra dviejų matricų X ir Y daugyba. Lygybės (4.5) – (4.8) parodo, kad kairioji ir dešinioji matricinio laipsnio operacijos yra apibrėžtos korektiškai. Jų įrodymus galima rasti [40] šaltinyje.

Turėdami vienpuses MLF galime apibrėžti *dvipusę MLF*, kuri bus naudojama šiame darbe. Tarkime, turime tris kvadratines m-tos eilės matricas Q, X ir Y. Tada pakėlę matricą Q laipsnine matrica X iš kairės ir laipsnine matrica Y iš dešinės gauname dvipusę MLF, kurią žymėsime

$$E = {}^XQ^Y. \tag{4.9}$$

Dvipusės ML elsponentės (arba trumpiau – ML eksponentės), t.y. matricos E elementai e_{ij} yra apskaičiuojami pagal formulę:

$$e_{ij} = \prod_{k=1}^{m}\prod_{l=1}^{m} q_{kl}^{x_{ik}y_{lj}} \tag{4.10}$$

Toliau sakydami MLF turėsime omenyje dvipusę funkciją. Šiame darbe dažnai išraiška (4.9) bus traktuojama ne tik kaip žymėjimas, bet ir kaip MLF lygties apibrėžimas. Šios lygties sprendimo uždavinys gali būti suformuluotas taip:

Matricinio laipsnio funkcijos (MLF) uždavinys. Rasti matricas X ir Y, tenkinančias lygtį (4.9), kai matricos Q ir E yra žinomos.

Iš MLF apibrėžimo matome, kad matricų X ir Y elementai yra matricos Q elementų laipsniai. Būtent dėl šios priežasties laipsninių matricų elementai yra pasirenkami iš skaitinio žiedo. Ryšį tarp multiplikacinės pusgrupės ir skaitinio žiedo pademonstruosime pavyzdžiu:

Pavyzdys. Tarkime, kad platforminės matricos Q elementai priklauso multiplikacinei grupei $\boldsymbol{Z}_n^*$. Tada laipsninių matricų X ir Y elementai turi būti pasirenkami iš skaitinio žiedo $\boldsymbol{Z}_{\lambda(n)}$.

Lemų pavidalu pateiksime dvi svarbias MLF savybes, kurios yra reikalingos kriptografiniams protokolams, paremtiems šia funkcija, konstruoti. Šių savybių įrodymus galima rasti K. Lukšio disertacijoje [40]. Jos kartu su lygybėmis (4.5) – (4.8) pagrindžia dvipusės MLF apibrėžimo korektiškumą.

4.1. lema. Jeigu ***R*** yra komutatyvus skaitinis žiedas ir ***S*** yra multiplikacinė komutatyvi pusgrupė, tai MLF tenkina asociatyvumo dėsnį

$$\left({}^{X}Q\right)^{Y} = {}^{X}\left(Q^{Y}\right) = {}^{X}Q^{Y} \tag{4.11}$$

4.2. lema. Jeigu ***R*** yra komutatyvus skaitinis žiedas ir ***S*** yra multiplikacinė komutatyvi pusgrupė, tai MLF tenkina tapatybę

$${}^{X}\left({}^{U}Q^{V}\right)^{Y} = {}^{(XU)}Q^{(VY)} \tag{4.12}$$

Pateiksime magistriniame darbe „Netiesinės algebrinės lygčių sistemos sprendinių skaičiaus analizė“ [45] nagrinėtas MLF algebrines savybes. Galima nesunkiai įsitikinti tuo, kad vienetinė matrica I yra MLF neutralusis matricinis laipsnis, t.y. kėlimas šiuo matriciniu laipsniu iš bet kurios pusės nekeičia platforminės matricos Q. Be to egzistuoja elementai $Q = \mathbf{1}$ ir $Z = \mathbf{0}$, čia $\mathbf{1}$ – matrica, kurios visi elementai yra lygūs vienetui, $\mathbf{0}$ – nulinė matrica, kuriems yra teisingos lygybės:

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mm} \end{pmatrix} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ \dots & \dots & \dots & \dots \\ y_{m1} & y_{m2} & \dots & y_{mm} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{pmatrix} \tag{4.13}$$

$$\begin{pmatrix} x_{11} & \dots & x_{1m} \\ \dots & \dots & \dots \\ x_{m1} & \dots & x_{mm} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & \dots & q_{1m} \\ q_{21} & q_{22} & \dots & q_{2m} \\ \dots & \dots & \dots & \dots \\ q_{m1} & q_{m2} & \dots & q_{mm} \end{pmatrix} \begin{pmatrix} 0 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{pmatrix} \tag{4.14}$$

$$\begin{pmatrix} 0 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1m} \\ q_{21} & q_{22} & \cdots & q_{2m} \\ \dots & \dots & \dots & \dots \\ q_{m1} & q_{m2} & \dots & q_{mm} \end{pmatrix} \begin{pmatrix} x_{11} & \dots & x_{1m} \\ \dots & \dots & \dots \\ x_{m1} & \dots & x_{mm} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{pmatrix} \quad (4.15)$$

Dabar paanalizuokime ML eksponentės elementą e_{ij}, kuris yra apskaičiuojamas pagal formulę (4.10). Iš elemento e_{ij} išraiškos matome, kad jis priklauso nuo visų platforminės matricos Q elementų bei nuo i-tosios matricos X eilutės ir j-tojo matricos Y stulpelio elementų. Kadangi elementas e_{ij} yra apskaičiuojamas naudojant tik daugybos bei kėlimo laipsniu operacijas, tai, jeigu nors vienas iš daugiklių yra lygus 0, tai ir $e_{ij} = 0$. Taigi turime, kad jeigu bent vienas iš matricos Q elementų yra lygus 0, ML eksponentė E yra nulinė matrica. Ši išvada galioja kiekviename žiede $\boldsymbol{Z_n}$, kai grupės parametras n yra pirminis skaičius. Tačiau jeigu n yra sudėtinis skaičius, tai žiede $\boldsymbol{Z_n}$ egzistuoja nulio dalikliai. Apibrėžkime tokią sąvoka [45]:

4.1 apibrėžimas Dviejų nenulinių žiedo $\boldsymbol{Z_n}$ elementų a ir b pora (a, b) vadinama *nulio daliklių pora*, jeigu $(a \cdot b) \bmod n = 0$.

Pavyzdys. Žiedas $\boldsymbol{Z_8}$ turi tris nulio daliklius: 2, 4, 6. (2, 4) ir (4, 6) yra nulio daliklių poros, bet (2, 6) nėra nulio daliklių pora, nes $(2 \cdot 6) \bmod 8 = 4$.

Pastebėkime, kad jeigu tarp matricos Q elementų pasitaiko bent viena nulio daliklių pora, tai ML eksponentė E yra nulinė matrica.

Pavyzdys. Nagrinėkime žiedą $\boldsymbol{Z_{10}}$. Šiame žiede egzistuoja nulio daliklių pora (2, 5). Tarkime, kad tarp matricos Q elementų egzistuoja nulio daliklių pora, o matricos X ir Y neturi nulinių elementų.

$$\begin{pmatrix} 3 & 1 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}^{\begin{pmatrix} 4 & 2 \\ 1 & 1 \end{pmatrix}} = \begin{pmatrix} 1^3 \cdot 3^1 & 2^3 \cdot 5^1 \\ 1^4 \cdot 3^3 & 2^4 \cdot 5^3 \end{pmatrix}^{\begin{pmatrix} 4 & 2 \\ 1 & 1 \end{pmatrix}} = \begin{pmatrix} 3 & 0 \\ 7 & 0 \end{pmatrix}^{\begin{pmatrix} 4 & 2 \\ 1 & 1 \end{pmatrix}} = \begin{pmatrix} 3^4 \cdot 0^1 & 3^2 \cdot 0^1 \\ 7^4 \cdot 0^1 & 7^2 \cdot 0^1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Matome, kad matrica E yra nulinė.

Aišku, kad šios MLF savybės kenkia kriptografinių protokolų, kurie paremti MLF, saugumui, todėl reikia parinkti tokią platforminę matricą, kad šių savybių būtų galima išvengti. Vienas iš galimų šios problemos sprendimo variantų yra parinkti matricos Q elementus iš multiplikacinės grupės $\boldsymbol{Z_n^*}$.

Nors įrodymo, jog MLF uždavinys priklauso NP-pilnųjų funkcijų klasei dar nėra, K. Lukšio darbuose yra nurodomi šie esminiai funkcijos privalumai [43]:

1. Matricinio laipsnio funkcija yra atspari pilno perrinkimo atakai. Šis rezultatas pasiekiamas ir naudojant mažesnius skaičius, kai yra keičiama kvadratinių matricų eilė.
2. Funkcija yra apskaičiuojama pakankamai greitai, kadangi daugybos ir kėlimo laipsniu operacijos yra atliekamos su mažais sveikais skaičiais, kurių dydis neviršija 8 bitų.
3. Kai skaičiavimams naudojami pakankamai maži skaičiai, tai daugybos ir kėlimo laipsniu operacijas galima atlikti naudojant paiešką iš anksto suformuotose šių operacijų lentelėse.
4. Kiekvienas išvesties matricos E elementas e_{ij} priklauso nuo visų platforminės matricos Q elementų q_{ij}. Ši savybė yra itin svarbi kriptografiniu požiuriu.

Antroji ir trečioji savybės yra svarbios tuo, kad leidžia apskaičiuoti ML eksponentės reikšmę naudojant polinominio laiko algoritmą nuo parametro m. Iš išraiškų (4.2) ir (4.4) matome, kad apskaičiuojant vieną vienpusės ML eksponentės elementą c_{ij} arba d_{ij} reikia atlikti m kėlimų laipsniu keliant visus matricos Q elementus atitinkamais laipsniais bei $(m - 1)$ daugybą tarp gautų elementų. Iš anksto paruoštos paieškos lentelės žymiai pagreitina šiuos skaičiavimus, nes šiuo atveju visos minėtos operacijos tampa ekvivalenčiomis ir yra atliekamos naudojant paprastą paiešką. Tai leidžia apskaičiuoti vieną vienpusės ML eksponentės elementą atlikus $(2m - 1)$ operaciją. Kadangi vienpuses ML eksponentes C ir D sudaro m^2 elementų, tai kiekvienai šiai matricai apskaičiuoti reikia atlikti $m^2(2m - 1)$ operacijų. Kadangi dvipusės MLF rezultatas gali būti apskaičiuotas naudojant savybę (4.11), tai dvipusė ML eksponentė E gali būti apskaičiuota atliekant $4m^3 - 2m^2$ operacijų. Šis rezultatas rodo, kad MLF reikšmę galima apskaičiuoti pakankamai greitai.

Pagrindinė parametro n įtaka skaičiavimams naudojant paieškos lenteles yra šių lentelių dydis. Kadangi mes siekiame sukurti protokolą, kurį galima būtų efektyviai realizuoti ribotų resursų skaičiavimo sistemose, tai turime įvertinti tą faktą, kad pasirenkant dideles parametro n reikšmes galime susidurti su atminties problema, t.y. paieškos lentelės gali užimti per didelę dalį turimos atminties.

Reikia pabrėžti, kad savo disertacijoje K. Lukšys nagrinėjo simetrinę kriptografiją bei sukonstravo S-blokus MLF pagrindu [40]. Šiame darbe MLF panaudosime asimetrinio šifravimo protokolui sukonstruoti. Norint sukonstruoti šį kriptografinį protokolą vienos MLF yra per mažai. Dėl šios priežasties buvo įvesti papildomi jungtinumo apribojimai. Naudojant šiuos apribojamus gaunamos svarbios tapatybės, kurios naudojamos konstruojant algoritmus.

Prieš konstruojant kriptografinį protokolą tam tikros funkcijos pagrindu pirmiausia būtina įsitikinti, kad naudojama funkcija yra vienkryptė. Šiam tikslui kitame skyrelyje nagrinėsime statistines MLF savybes.

4.2. Statistinės MLF savybės

MLF nėra atspari statistinėms atakoms, jeigu ML eksponentės E elementai nėra tolygiai pasiskirstę. Kadangi šiame darbe nagrinėjame baigtines struktūras, tai

sakysime, jog atsitiktinis dydis ξ yra pasiskirstęs pagal tolygųjį skirstinį baigtinėje aibėje $\boldsymbol{A}$, jeigu tikimybė, kad atsitiktinis dydis ξ įgyja reikšmę $a \in \boldsymbol{A}$ yra:

$$prob(\xi = a) = \frac{1}{|\boldsymbol{A}|} \ . \tag{4.16}$$

Tolygusis skirstinys turi labai svarbią savybę: atsitiktinio dydžio, pasiskirsčiusio pagal šį skirstinį baigtinėje aibėje $\boldsymbol{A}$, entropija[1] yra didžiausia iš visų diskrečiųjų dydžių šioje aibėje. Didelė entropija yra svarbi kriptografijoje, kadangi tokiu atvėju piktavalis negali suteikti prioreteto nei vienam iš pasirinkimo variantų.

Tarkime, kad matricinių laipsnių X ir Y elementai x_{ij} ir y_{ij} yra pasiskirstę tolygiai. Norint išvengti statistinių atakų reikia pasirinkti matricos Q elementus q_{ij} taip, kad ML ekponentės elementai e_{ij} būtų pasiskirstę pagal tolygųjį skirstinį. Kadangi, pagal Lagrandžo teoremą, didžiausias grupės $\boldsymbol{Z_n}^*$ elementų periodas λ(n) dalija šios grupės eilę $|\boldsymbol{Z_n}^*|$, tai joje egzistuoja cikliniai pogrupiai, kurių eilė yra λ(n). Iš ankstesnio skyrelio žinome, kad tokiu atvėju laipsninis žiedas yra $\boldsymbol{Z_r}$, čia r = λ(n). Pasirinkime tokį sudėtinį skaičių $n = pq$ (p ir q – pirminiai skaičiai), kad grupėje $\boldsymbol{Z_n}^*$ egzistuotų bent du tokie pogrupiai ir pažymėkime juos $\boldsymbol{C}_{r.1}$ ir $\boldsymbol{C}_{r.2}$. Iš Lagranžo teoremos žinome, kad jeigu $\boldsymbol{Z_n}^*$ necilkinė grupė, tai didžiausia galima ciklinio pogrupio $\boldsymbol{C_r}$ eilė, t.y. Karmailklo funkcijos reikšmė, negali viršyti $\phi(n)/2$. Tai reiškia, kad didžiausio galimo dydžio cikliniai pogrupiai yra gaunami kai tenkinama salyga

$$\phi(n) = 2\lambda(n). \tag{4.17}$$

Tarkime, kad grupėje $\boldsymbol{Z_n}^*$ egzistuoja minėti pogrupiai. Laisvąją šių pogrupių sandaugą žymėkime $\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2}$. Pagal apibrėžimą ši sandauga atrodo taip:

$$\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2} = \{c = c_1 \cdot c_2 \mid c_1 \in \boldsymbol{C}_{r.1}, c_2 \in \boldsymbol{C}_{r.2}\}, \tag{4.18}$$

čia · yra daugybos operacija grupėje $\boldsymbol{Z_n}^*$.

Toliau panaudosime šiuos teiginius:

4.3 teiginys. $\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2}$ yra grupė tada ir tik tada, kai $\boldsymbol{C}_{r.1}$ ir $\boldsymbol{C}_{r.2}$ yra komutatyvios grupės [1].

Matome, kad šio teiginio sąlygos yra išpildytos.

4.4 teiginys. Tarkime, kad baigtinė grupė $\boldsymbol{G}$ turi du pogrupius $\boldsymbol{G_1}$ ir $\boldsymbol{G_2}$. Šių pogrupių laisvąją sandaugą $\boldsymbol{G_1} * \boldsymbol{G_2}$ sudaro lygiai $|\boldsymbol{G_1}|\ |\boldsymbol{G_2}| \ / \ |\boldsymbol{G_1} \cap \boldsymbol{G_2}|$ skirtingų elementų [1].

---

[1] Entropija – tai atsitiktinio dydžio charakteristika, kuri kiekibyškai apibūdina to dydžio neapibrėžtumo. Kuo ši charakteristika didesnė, tuo sunkiau prognozuoti atsitiktinio dydžio reikšmę.

Tarkime, kad $\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2}$ yra pogrupis. Kadangi $\lambda(n)$ yra lyginis skaičius, tai didžiausia galima šio pogrupio eilė yra $\lambda(n) / 2$. Remiantis teiginiu 4.4 mes galime suformuluoti tokią lemą:

4.5 lema. Tarkime, kad $\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2}$ yra pogrupis, kurio eilė $|\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2}| = \lambda(n) / 2$. Tada $\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2} = \boldsymbol{Z}_{\boldsymbol{n}}^{*}$.

Įrodymas. Jei galioja visos aukščiau aprašytos salygos, tai turime tokią tapatybę:

$$\left|\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2}\right| = \left|\boldsymbol{C}_{r.1}\right|\left|\boldsymbol{C}_{r.2}\right| / \left|\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2}\right| = \frac{\lambda^2(n)}{\lambda(n)/2} = 2\lambda(n) = \phi(n).$$

Taigi grupės $\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2}$ ir $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$ yra sudarytos iš (n) skirtingų elementų. Tačiau $\boldsymbol{C}_{r.1}$, $\boldsymbol{C}_{r.2}$ ir $\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2}$ yra grupės $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$ pogrupiai, o tai reiškia, kad $\boldsymbol{C}_{r.1} * \boldsymbol{C}_{r.2} = \boldsymbol{Z}_{\boldsymbol{n}}^{*}$.■

Raskime šių grupių generatorius ir pažymėkime jų aibes atitinkamai Γ_1 ir Γ_2. Tada galioja šie teiginiai:

4.6 teiginys. Tarkime, kad x yra atsitiktinis grupės $\boldsymbol{Z}_{\boldsymbol{r}}$ elementas, o γ – fiksuotas generatorius iš Γ_1 (Γ_2). Tada elementas γ^x yra pasiskirstęs grupėje $\boldsymbol{C}_{\boldsymbol{r.1}}$ ($\boldsymbol{C}_{\boldsymbol{r.2}}$) pagal tokį skirstinį, kokį elementas x turi žiede $\boldsymbol{Z}_{\boldsymbol{r}}$ [46].

4.7 teiginys. Tarkime, kad elementas $x \in \boldsymbol{Z}_{\boldsymbol{n}}^{*}$ yra fiksuotas, o elementas $y \in \boldsymbol{Z}_{\boldsymbol{n}}^{*}$ pasirinktas atsitiktinai. Tada elementų y ir $z = xy$ skirstiniai sutampa [46].

4.8 teiginys. Tarkime, kad x ir y yra nepriklausomi tolygūs atsitiktiniai grupės $\boldsymbol{Z}_{\boldsymbol{r}}$ elementai, o γ_1 ir γ_2 – fiksuoti generatoriai atitinkamai iš Γ_1 ir Γ_2. Tada elementas $z = \gamma_1^x \gamma_2^y$ yra pasiskirstęs tolygiai grupėje $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$.

Įrodymas. Grupę $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$ suskaidome į šias dalis: $\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2}$, $\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}$, $\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1}$ ir $\boldsymbol{Z}_{\boldsymbol{n}}^{*} \setminus \left(\boldsymbol{C}_{r.1} \cup \boldsymbol{C}_{r.2}\right)$. Pirmųjų trijų aibių struktūros yra tokios:

$$\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2} = \left\{\gamma_1^{2k} \mid k = 0,1,\ldots,\frac{\lambda(n)}{2} - 1\right\} = \left\{\gamma_2^{2k} \mid k = 0,1,\ldots,\frac{\lambda(n)}{2} - 1\right\};$$

$$\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2} = \left\{\gamma_1^{2k+1} \mid k = 0,1,\ldots,\frac{\lambda(n)}{2} - 1\right\};$$

$$\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1} = \left\{\gamma_2^{2k+1} \mid k = 0,1,\ldots,\frac{\lambda(n)}{2} - 1\right\}.$$

Aibę $\boldsymbol{Z}_{\boldsymbol{n}}^{*} \setminus \left(\boldsymbol{C}_{r.1} \cup \boldsymbol{C}_{r.2}\right)$ sudaro likusieji aibės $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$ elementai. Matome, kad pirmųjų trijų aibių galios yra lygios

$$\left|\boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}\right| = \left|\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}\right| = \left|\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1}\right| = \lambda(n)/2 = \phi(n)/4 .$$

Iš čia gauname, kad ir aibę $\boldsymbol{Z}_{\boldsymbol{n}}^{*} \setminus \left(\boldsymbol{C}_{r.1} \bigcup \boldsymbol{C}_{r.2}\right)$ sudaro $\phi(n)/4$ elementų, kadangi aibės $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$ galia $|\boldsymbol{Z}_{\boldsymbol{n}}^{*}| = \phi(n)$.

Pasirinkime vieną iš kintamųjų (pvz. x) ir fiksuokime jį. Tada tikimybė, kad elementas γ_1^x priklauso aibei $\boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$ yra:

$$prob\left(\gamma_1^x \in \boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}\right) = \frac{\lambda(n)/2}{\lambda(n)} = \frac{1}{2} .$$

Tokia pati yra ir tikimybė, kad elementas $\gamma_1^x \in \boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}$, kadangi šios aibės galia $\left|\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}\right| = \lambda(n)/2$. Elementas γ_2^y pagal teiginio salygą yra pasiskirstęs pagal tolygųjį skirstinį. Tada priklausomai nuo laipsnių x ir y reikšmių ir remiantis 4.7 teiginiu yra galimi šie keturi atvėjai:

- Jeigu γ_1^x ir γ_2^y priklauso $\boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$, tai elementas z yra pasiskirstęs tolygiai aibėje $\boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$. Šio įvykio tikimybė yra lygi:
$$prob\left(\gamma_2^y \in \boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2} \mid \gamma_1^x \in \boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}\right) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} ,$$
kadangi atsitiktiniai įvykiai $\gamma_1^x \in \boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$ ir $\gamma_2^y \in \boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$ yra nepriklausomi.
- Jeigu γ_1^x ir γ_2^y priklauso atitinkamai $\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}$ ir $\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1}$, tai elementas z yra pasiskirstęs tolygiai aibėje $\boldsymbol{Z}_{\boldsymbol{n}}^{*} \setminus \left(\boldsymbol{C}_{r.1} \bigcup \boldsymbol{C}_{r.2}\right)$, kadangi šis elementas negali priklausyti nei vienai iš trijų likusių aibių. Šio įvykio tikimybė yra lygi $\frac{1}{4}$.
- Jeigu γ_1^x ir γ_2^y priklauso atitinkamai $\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}$ ir $\boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$, tai elementas z yra pasiskirstęs tolygiai aibėje $\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2}$. Šio įvykio tikimybė yra lygi $\frac{1}{4}$.
- Jeigu γ_1^x ir γ_2^y priklauso atitinkamai $\boldsymbol{C}_{r.1} \bigcap \boldsymbol{C}_{r.2}$ ir $\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1}$, tai elementas z yra pasiskirstęs tolygiai aibėje $\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1}$. Šio įvykio tikimybė yra lygi $\frac{1}{4}$.

Kadangi kiekvienoje grupės $\boldsymbol{Z}_{\boldsymbol{n}}^{*}$ dalyje elementas z yra pasiskirstęs tolygiai ir šis rezultatas nepriklauso nuo fiksuoto kintamojo pasirinkimo, tai toks yra pasiskirstymas visoje grupėje. ■

Pavyzdys. Nagrinėkime grupę $\boldsymbol{Z}_{15}^{*} = \{1;2;4;7;8;11;13;14\}$. Šioje grupėje egzistuoja du cikliniai pogrupiai: $\boldsymbol{C}_{r.1} = \{1;2;4;8\}$ ir $\boldsymbol{C}_{r.2} = \{1;4;7;13\}$. Šių pogrupių generatorių

aibės yra $\Gamma_1 = \{2;\ 8\}$ ir $\Gamma_2 = \{7;\ 13\}$. Taip pat turime $\boldsymbol{C}_{r.1} \cap \boldsymbol{C}_{r.2} = \{1;4\}$, $\boldsymbol{C}_{r.1} \setminus \boldsymbol{C}_{r.2} = \{2;8\}$, $\boldsymbol{C}_{r.2} \setminus \boldsymbol{C}_{r.1} = \{7;13\}$ ir $\boldsymbol{Z_n}^* \setminus (\boldsymbol{C}_{r.1} \cup \boldsymbol{C}_{r.2}) = \{11;14\}$.

Pasirinkime $\gamma_1 = 2$ ir $\gamma_2 = 7$. Šių elementų laipsniai yra pasirenkami iš žiedo $\boldsymbol{Z_4}$, kadangi $\lambda(15) = 4$. Sudarykime lentelę, kurioje stebėsime kokias reikšmes įgyja elementas $z = \gamma_1^x \gamma_2^y$ priklausomai nuo x ir y reikšmių.

4.1 lentelė. Elemento $z = \gamma_1^x \gamma_2^y$ tolygaus pasiskirstymo pavyzdys.

x	y	γ_1^x	γ_2^y	$z = \gamma_1^x \gamma_2^y$
0	0	1	1	1
0	1	1	7	7
0	2	1	4	4
0	3	1	13	13
1	0	2	1	2
1	1	2	7	14
1	2	2	4	8
1	3	2	13	11
2	0	4	1	4
2	1	4	7	13
2	2	4	4	1
2	3	4	13	7
3	0	8	1	8
3	1	8	7	11
3	2	8	4	2
3	3	8	13	14

Iš lentelės matome, kad kiekviena elemento z reikšmė pasikartoja 2 kartus. Taigi elementas z yra tolygiai pasiskirstęs grupėje $\boldsymbol{Z_n}^*$.

Pažymėkime $\Gamma = \Gamma_1 \cup \Gamma_2$ ir suformuluokime šią išvadą:

4.9 išvada. Tarkime, kad 4.5 lemos salygos yra tenkinamos ir fiksuotos platforminės matricos Q elementai $q_{ij} \in \Gamma$. Jeigu atsitiktinių matricinių laipsnių X ir Y elementai turi tolygųjį pasiskirstymą, tai ML eksponentės E elementai yra pasiskirstę tolygiai grupėje $\boldsymbol{Z_n}^*$.

Įrodymas. Kadangi visi matricos Q elementai yra pogrupių generatoriai, tai

$$e_{ij} = \gamma_1^{\alpha} \gamma_2^{\beta},$$

čia α, β yra tiesiniai matricų X ir Y elementų dariniai su koeficientais, kurie yra tarpusavyje pirminiai su skaitinio žiedo parametru r. Remiantis teiginiu 4.7 laipsniai

α, β yra pasiskirstę tolygiai grupėje $\boldsymbol{Z_r}$. Tada remiantis teiginiu 4.8 matricos E elementai yra pasiskirstę tolygiai grupėje $\boldsymbol{Z_n^*}$. ■

Pažymėkime MLF lygties (4.9) sprendinių aibę $\boldsymbol{Pow}(Q, E) = \{(X, Y): {}^{X}Q^{Y} = E\}$. Pagal 4.9 išvadą šios aibės elementai yra pasiskirstę tolygiai.

4.3. Statistinės MLF su jungtinumo apribojimais savybės

Skyrelyje 4.1 paminėjome, kad norint konstruoti kriptografinius protokolus, paremtus MLF, prie šios funkcijos reikia pridėti papildomus jungtinumo apribojimus. Įveskime šiuos apribojimus [47]:

$$\begin{matrix} X^{-1}AX = C \\ Y^{-1}BY = D \end{matrix}, \tag{4.19}$$

čia matricos A, B, C, D yra žinomos ir priklauso laipsniniam žiedui $\boldsymbol{M_R}$, čia $\boldsymbol{R} = \boldsymbol{Z_r}$. Norint nulaužti MLF su jungtinumo apribojimais reiškia rasti tokias matricas X ir Y, kurios tenkintų lygčių sistemą

$$\begin{cases} {}^{X}Q^{Y} = E \\ X^{-1}AX = C \\ Y^{-1}BY = D \end{cases} \tag{4.20}$$

kai kitos matricos yra žinomos [47].

Jau įsitikinome, kad (4.20) sistemos pirmosios lygties sprendiniai turi tolygųjį pasiskirstymą. Dabar nagrinėsime jungtinumo lygčių (4.19) sprendinių pasiskirstymą. Tarkime, turime lygtį

$$X^{-1}AX = C \tag{4.21}$$

Įrodykime teiginį, kuris padės mums įvertinti jungtinumo lygties (4.21), apibrėžtos žiede $\boldsymbol{Z_r}$, sprendinių skaičių. Tarkime, turime abstrakčias kvadratines matricas A, B ir C, kurios yra apibrėžtos žiede $\boldsymbol{Z_{pq}}$, čia p ir q yra du skirtingi nelyginiai pirminiai skaičiai. Pagal kinų liekanų teoremą turime, kad žiedas $\boldsymbol{Z_{pq}}$ yra izomorfinis tiesioginei dviejų žiedų $\boldsymbol{Z_p}$ ir $\boldsymbol{Z_q}$ sandaugai $\boldsymbol{Z_p} \times \boldsymbol{Z_q}$. Pažymėkime $\mathbf{1}_p$ ir $\mathbf{1}_q$ žiedo $\boldsymbol{Z_{pq}}$ idempotentus. Taip pat apibrėžkime $A_p = \{a_{ij}\} \bmod p$, $A_q = \{a_{ij}\} \bmod q$.

4.10 teiginys. Jei $A_pB_p = C_p$ ir $A_qB_q = C_q$, tai matricos A, B ir C tenkina tapatybę $AB = C$.

Įrodymas. Pritaikius (2.6) lygybę visiems matricų A ir B elementams turime, kad

$AB = (A_p\mathbf{1}_p + A_q\mathbf{1}_q)(B_p\mathbf{1}_p + B_q\mathbf{1}_q) = A_pB_p\mathbf{1}_p + A_pB_q\mathbf{1}_p\mathbf{1}_q + A_qB_p\mathbf{1}_q\mathbf{1}_p + A_qB_q\mathbf{1}_q = A_pB_p\mathbf{1}_p + A_qB_q\mathbf{1}_q$,

kadangi $\mathbf{1}_p\mathbf{1}_q = 0$. Kadangi $A_pB_p = C_p$ ir $A_qB_q = C_q$, turime, kad $AB = C$. ■

4.11 išvada. Jei $A_pB_p = B_pA_p$ ir $A_qB_q = B_qA_q$, tai matricos A ir B komutuoja, t.y. $AB = BA$.

Dabar jau galime įvertinti (4.21) lygties, apibrėžtos žiede $\boldsymbol{Z}_r$, sprendinių skaičių kai $r = 2s$, čia s yra pirminis skaičius ir $s \neq 2$. Toks parametro r pasirinkimas pagrįstas tuo, kad Karmaiklo funkcija $\lambda(n)$ yra lyginis skaičius, o pasirenkant pirminį skaičių s mes sumažiname galimų multiplikacinės grupės $\boldsymbol{Z}_n^*$, kuri yra naudojama platforminiai pusgrupei apibrėžti, elementų periodų variantų skaičių bei padidiname šios grupės ciklinių pogrupių generatorių skaičių. Remiantis teiginiu 4.10 pirmiausia nagrinėkime (4.21) lygtį lauke $\boldsymbol{Z}_s$.

Tarkime, kad matricos A ir C yra panašios į Žordano matricą J, t.y. šias matricas galima išreikšti kanoninėje Žordano formoje

$$\begin{aligned} A &= K^{-1}JK \\ C &= L^{-1}JL \end{aligned}, \tag{4.22}$$

čia K ir L yra atitinkamai matricų A ir C tikrinių vektorių matricos ir

$$J = \begin{pmatrix} \mu & 1 & 0 & \dots & 0 & 0 \\ 0 & \mu & 1 & \dots & 0 & 0 \\ 0 & 0 & \mu & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \mu & 1 \\ 0 & 0 & 0 & \dots & 0 & \mu \end{pmatrix}, \tag{4.23}$$

čia μ yra matricų A ir C m-tojo kartotinumo tikrinė reikšmė. Kadangi $AX = XC$, tai turime lygtį

$$K^{-1}JKX = XL^{-1}JL\,. \tag{4.24}$$

Padauginkime (4.24) lygtį iš matricos K iš kairės ir iš matricos L^{-1} iš dešinės. Gauname

$$JKXL^{-1} = KXL^{-1}J\,. \tag{4.25}$$

Žymėkime $\tilde{X} = KXL^{-1}$. Turime

$$J\tilde{X} = \tilde{X}J\,. \tag{4.26}$$

Tačiau visos komutuojančios su Žordano matrica J turi formą

$$\tilde{X} = \begin{pmatrix} a_1 & a_2 & \dots & a_{m-1} & a_m \\ 0 & a_1 & \dots & a_{m-2} & a_{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_1 & a_2 \\ 0 & 0 & \dots & 0 & a_1 \end{pmatrix}. \tag{4.27}$$

Iš (4.27) išraiškos matome, kad turime m skirtingų parametrų $a_1,\dots,a_m$. Kadangi $|\boldsymbol{Z}_s| = s$, tai egzistuoja lygiai s^m skirtingų matricų, komutuojančių su matrica J. Šių matricų aibę žymėkime $\boldsymbol{Com}(J)$, o šių matricų skaičių – $|\boldsymbol{Com}(J)|$.

Tačiau ne visos matricos, komutuojančios su matrica J, taip pat tenkina ir (4.21) lygtį. Taip yra todėl, kad ne visos šios matricos turi atvirkštines matricas. Pastebėkime, kad atvirkštinės matricos egzistuoja tuo atveju, kai $a_1 \neq 0$. Tada turime lygiai $s^{(m-1)}(s-1)$ neišsigimusių matricų, komutuojančių su matrica J. Šių matricų aibę pažymėkime $\boldsymbol{Com}^*(J)$. Kadangi visus (4.21) lygties sprendinius gauname pagal formulę

$$X = K^{-1}\tilde{X}L \tag{4.28}$$

čia matrica $\tilde{X} \in \boldsymbol{Com}^*(J)$, tai galioja teiginys, kurio įrodymą ką tik pateikėme.

4.12 teiginys. Jei A ir C yra kvadratinės m-tos eilės matricos, apibrėžtos virš lauko $\boldsymbol{Z}_s$, kurios yra panašios į Žordano matricą (4.23), tai jungtinumo lygtis (4.21) turi lygiai $s^{(m-1)}(s-1)$ sprendinių.

Naudojant šį teiginį bei teiginį prieiname išvadą apie jungtinumo lygties (4.21), apibrėžtos žiede $\boldsymbol{Z}_r$, sprendinių skaičių N.

4.13 išvada. Jei A_2 ir A_s panašios į Žordano matricas (4.23) laukuose $\boldsymbol{Z}_2$ ir $\boldsymbol{Z}_s$, tai jungtinumo lygtis (4.21) turi lygiai $r^{(m-1)}(s-1)$ sprendinių. Taigi šiuo atveju $|\boldsymbol{Com}^*(J)| = r^{(m-1)}(s-1)$. Jungtinumo lygties (4.21) sprendinių aibę pažymėkime $\boldsymbol{Conj}(A, C)$.

Šaltinyje [48] buvo įrodyta, kad bet kuri matrica, komutuojanti su Žordano matrica (4.23), gali būti išreikšta per daugianarį nuo J. Daugianario laipsnis yra $(m-1)$, kadangi egzistuoja lygiai m tiesiškai nepriklausomų matricų, komutuojančių su matrica J. Tegul $t = [t_0, t_1, \dots, t_{(m-1)}]$ žymi daugianario koeficietų vektorių, o $T_m(\boldsymbol{Z}_s)$ žymi šių vektorių aibę. Tada $|T_m(\boldsymbol{Z}_s)| = s^m$. Kadangi visos aibės $\boldsymbol{Com}(J)$ matricos yra skirtingos ir $|\boldsymbol{Com}(J)| = s^m$ tai egzistuoja bijektyvus vaizdavimas β: $T_m(\boldsymbol{Z}_s) \to \boldsymbol{Com}(J)$, kuris yra apibrėžiamas taip:

$$\beta(t) = P, P \in \boldsymbol{Com}(J) \tag{4.29}$$

t.y. kiekvieną koeficientų vektorių atitinka vienintelė matrica ir atvirkščiai. Jeigu koeficientų vektorius pasirinktas atsitiktinai pagal tolygųjį skirstinį tai ir matricų skirstinys aibėje ***Com***(*J*) yra tolygusis. Taip pat kadangi vaizdavimas β yra bijektyvus, tai galima apibrėžti aibės $\boldsymbol{T_m(Z_r)}$ poaibį, kuri sudaro visi koeficientų vektoriai, atitinkantys neišsigimusias matricas. Kadangi vaizdavimas

$$\beta^*(t) = P, P \in \boldsymbol{Com}^*(J) \tag{4.30}$$

yra bijekcinio vaizdavimo (4.29) atskiras atvejis, tai šis vaizdavimas taip pat yra bijektyvus ir visi mūsų samprotavimai taip pat galioja ir aibei ***Com****(*J*). Kadangi tikrinių vektorių matricos yra fiksuotos, tai aibių ***Conj***(*A*, *C*) ir ***Conj***(*B*, *D*) elementai turi tolygųjį pasiskirstymą.

Aišku, kad (4.20) sistemos sprendinių aibė ***Sol***(*Q*, *E*, *A*, *B*, *C*, *D*) yra dviejų aibių ***Pow***(*Q*, *E*) ir ***Conj***(*A*, *C*) × ***Conj***(*B*, *D*) sankirta, t.y.

$$\boldsymbol{Sol}(Q,E,A,B,C,D) = \boldsymbol{Pow}(Q,E) \cap \left(\boldsymbol{Conj}(A,C) \times \boldsymbol{Conj}(B,D)\right) \tag{4.31}$$

Kadangi abiejų šių aibių elementai turi tolygųjį pasiskirstymą, tai tokį patį pasiskirstymą turi ir (4.20) sistemos sprendinių aibės ***Sol***(*Q*, *E*, *A*, *B*, *C*, *D*) elementai. Įrodėme teiginį [47]:

4.14 teiginys. Tegul platforminė matrica *Q* yra apibrėžta virš multiplikacinės grupės $\boldsymbol{Z_n^*}$, kurios laipsninis žiedas apibrėžtas virš žiedo $\boldsymbol{Z_{2^s}}$. Tarkime, kad laipsninės matricas *X* ir *Y* sudaro tolygūs atsitiktiniai skaitinio žiedo $\boldsymbol{Z_{2^s}}$ elementai. Tada MLF (4.9) su apribojimais (4.22) ML eksponentės *E* elementai pasiskirstę pagal tolygųjį skirstinį.

Iš šio teiginio mes darome išvadą, jog MLF gali būti panaudota pseudoatsitiktinių skaičių generatoriui sukurti ir, remiantis [11] šaltinio teiginiais 6.2 ir 6.3, darome prielaidą, jog MLF gali būti laikoma kandidate į vienkryptes funkcijas ir yra tinkama kriptografiniams protokolams sukurti.

Šį rezultatą galima paaiškinti taip: tolygusis elementų pasiskirstymas aibėse ***Pow***(*Q*, *E*) ir ***Conj***(*A*, *C*) × ***Conj***(*B*, *D*) reiškia, kad atsitiktinį šių aibių elementą galima pasirinkti su vienoda tikimybe, t.y. nei vienas elementas neturi prioriteto prieš kitus aibės elementus. Taigi prognozuoti koks atsitiktinis elementas bus pasirenkamas remiantis ankstesniais bandymais yra neįmanoma.

4.4. Išvados ir rezultatai

- MLF yra vaizdavimas, kuris laipsninio matricų žiedo elementų porai (*X*, *Y*) priskiria matricą *E* iš platforminės matricų pusgrupės. Šios funkcijos pagrindas – matrica *Q*, turi būti parinktas iš platforminės matricų pusgrupės.

- Naudojant MLF nereikia atlikinėti skaičiavimų su labai dideliais skaičiais, kadangi atsparumą pilno perrinkimo atakai galima užtikrinti ir naudojant mažesnius skaičius, kai yra keičiama kvadratinių matricų eilė.
- Kiekvienas ML eksponentės E elementas priklauso nuo visų platforminės matricos Q elementų.
- MLF apgręžiamumas yra susietas su kriptoanalizės ataka, siekiant nulaužti vartotojo slaptąjį raktą pagal turimus viešuosius duomenis. Šio uždavinio sprendimo sudėtingumas yra paremtas laipsninių algebrinių lygčių sistemos (4.10) sprendimo sudėtingumu.
- Siekiant užtikrinti maksimalią ML eksponentės E elementų entropiją, platforminės matricos Q elementai tūri būti pasirenkami iš multiplikacinės pusgrupės pogrupių generatorių aibės. Tokiu būdu yra užtikrinamos geros MLF statistinės savybės, kurias turi tenkinti vienkryptė funkcija.

5. ASIMETRINIO ŠIFRAVIMO PROTOKOLO APRAŠYMAS IR ALGEBRINIŲ SAVYBIŲ TYRIMAS

5.1. Pirmoji MLAŠ protokolo versija

Pradinį asimetrinio šifravimo, paremto MLF su jungtinumo apribojimais, protokolo variantą šio darbo autorius pasiūlė 2011 metų rudenį. Šį protokolą mes pavadinome matricinio laipsnio asimetriniu šifravimu (MLAŠ).

Protokolo idėja yra paremta prielaida, jog MLF su jungtinumo apribojimais yra vienkryptė funkcija. Naudojant šią fukciją siuntėjas (Bronius) užšifruoja savo pranešimą *M*, taip gaudamas šifrogramą *C*. Gavėja (Aldona), naudojant MLF iššifruoja šifrogramą *C*, taip gaudama Broniaus pranešimą *M*.

Protokolui sudaryti panaudosime XOR (pobitinio sumavimo moduliu 2) operaciją, kurią žymėsime $\oplus$. Šios operacijos lentelė atrodo taip:

5.1 lentelė. Operacijos XOR lentelė

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Iš šios lentelės matome, kad $x \oplus y = 0$ tada ir tik tada, kai $x = y$. Priešingu atveju $x \oplus y = 1$. Norint atlikti XOR operaciją abu elementai yra nagrinėjami dvejetainiu pavidalu. Nuo gauto dvejetainio rezultato grįžtame į pradinę aibę.

Pavyzdys. Nagrinėkime aibę $\boldsymbol{Z}_{16}$. Šioje aibėje apskaičiuokime $10 \oplus 14$. Turime:

$$10 \oplus 14 = (1010) \oplus (1110) = (0100) = 4.$$

Protokolo viešieji parametrai yra matrica *Q*, kuri priklauso platforminei pusgrupei, ir matrica *Z*, kuri priklauso laipsniniam žiedui. Aldona turi savo slaptą raktą $PrK_A = \{X, U\}$, čia matrica *X* atsitiktinai pasirenkama iš laipsninio žiedo taip, kad egzistuotų jos atvirkštinė matrica X^{-1}, o matrica *U* yra gaunama apskaičiuojant daugianario $P_A(Z)$, kurio koeficientai $\{a_0, a_1, \ldots, a_{m-1}\}$ yra atsitiktinai pasirenkami iš skaitinio žiedo $\boldsymbol{Z}_r$, reikšmę, t.y $U = P_A(Z) = a_0 I + a_1 Z + \ldots + a_{m-1} Z^{m-1}$. Aldonos viešasis raktas yra $PuK_A = \{XZX^{-1} = A, {}^{X}Q^{U} = E\}$. Bronius užšifruoja pranešimą *M* atlikdamas šiuos veiksmus [49]:

1. Jis pasirenka neišsigimusią matricą *Y*, kuri priklauso laipsniniam žiedui.
2. Bronius naudoja Aldonos viešą raktą:

a) Pasirenka $V = P_B(Z)$ ir apskaičiuoja $P_B(A) = XVX^{-1}$ bei ${}^{V}Q^{Y}$. Jo viešas raktas yra $PuK_B = \{ Y^{-1}ZY = B,\ {}^{V}Q^{Y} = F\}$;
b) Kelia matricą E laipsniu XVX^{-1} iš kairės;
c) Gautą matricą kelia laipsniu Y iš dešinės.

Tokiu būdu Bronius gauna matricą $K_B = {}^{XV}Q^{UY}$. Kadangi šios matricos elementai yra atsitiktiniai ir pasiskirstę tolygiai, tai matrica K_B gali būti naudojama kaip užšifravimo raktas pranešimui M užšifruoti. Šį raktą reikia generuoti kiekvieną kartą siunčiant pranešimą.

3. Bronius užšifruoja pranešimą M naudojant užšifravimo raktą K_B. Šifrograma $C = K_B \oplus M$.
4. Bronius siunčia Aldonai gautą šifrogramą C ir savo viešą raktą PuK_B.

Aldona, turėdama Broniaus atsiustus duomenis, iššifruoja jo pranešimą M atlikdama šiuos veiksmus:

1. Naudojant matricą B ir daugianarį $P_A(Z)$ Aldona apskaičiuoja $Y^{-1}UY = P_U(Y^{-1}ZY)$
2. Aldona kelia matricą F laipsniu $Y^{-1}UY$ iš dešinės;
3. Gautą matricą Aldona kelia laipsniu X iš kairės ir gauna iššifravimo raktą $K_A = {}^{XV}Q^{UY}$. Kadangi šis raktas sutampa su Broniaus šifravimo raktu, tai galime pažymėti $K_A = K_B = K$.
4. Aldona iššifruoja Broniaus pranešimą M naudojant gautą iššifravimo raktą K, šifrogramą C bei tapatybę $K \oplus C = K \oplus K \oplus M = M$, kadangi $K_B = K$.

Šio protokolo pagrindinis pranašumas lyginant su protokolais, paremtais jungtinio elemento paieškos uždaviniu, yra tai, kad tik matricos U ir V yra komutuojančios. Taip pat, kadangi abi matricos yra gaunamos apskaičiuojant atitinkamų daugianarių reikšmes nuo matricos Z, galima sutrumpinti slaptų raktų ilgius, įrašant į atmintį tik daugianarių koeficientų reikšmes.

Reikia paminėti, kad, nors šiuo atvėju matricos U ir V yra komutuojančios, tai nėra esminė salyga protokolui įvykdyti, t.y. šios matricos gali ir nekomutuoti. Tuo įsitikinsime vėliau, kai nagrinėsime patobulintą protokolo versiją.

Taigi matome, kad du vartotojai gali įvykdyti protokolą naudojant tik viešai paskelbta informacija ir savo slaptus duomenis. Pagrindinės savybės, kurios yra naudojamos yra (4.11), (4.12) bei jungtinumo savybė

$$P\left(XZX^{-1}\right) = X \cdot P\left(Z\right) \cdot X^{-1}, \tag{5.1}$$

čia $P(\ )$ yra daugianaris. Remiantis šiomis savybėmis Bronius gauna raktą K, nes

$$P_B(A) = P_B(XZX^{-1}) = X \cdot P_B(Z) \cdot X^{-1} = XVX^{-1};$$

$$\left(XVX^{-1}\right)E^{Y} = \left(XVX^{-1}\right)\left({}^{X}Q^{U}\right)^{Y} = {}^{XVX^{-1}X}Q^{UY} = {}^{XV}Q^{UY} = K.$$

Analogiškus veiksmus atlieka ir Aldona.

Pastebėkime, kad pasiūlytas protokolas turi panašumų su trečiame skyriuje minėtais protokolais. Iš MLF apibrėžimo matome, kad kiekvienas ML eksponentės elementas yra apskaičiuojamas kaip platforminės matricos elementų, pakeltų sveikaisiais laipsniais, sandauga. Tai reiškia, kad mes susiduriame su DLU sistemą, sudaryta iš m^2 lygčių. Tačiau, kadangi mes orientuojamės į mažus skaičius (nedaugiau kaip vieno baito ilgio), tai šis uždavinys yra lengvai sprendžiamas naudojant Pohling'o ir Hellman'o algoritmą. Dėl šios priežasties mes nenaudojame ciklinių (pus)grupių MLAŠ platformai apibrėžti.

Nagrinėjant Aldonos viešąjį raktą taip pat matome, kad ieškant slaptojo rakto dalies – matricos X – susiduriame su JPU, kurio teoriniai sprendimo matricų žieduose algoritmai yra žinomi. Tačiau MLAŠ atveju skirtingai nuo Ko-Lee protokolo JPU negalima pakeisti jį atitinkančiu dekompozicijos uždaviniu, kadangi (4.9) išraiškoje platforminė matrica Q yra pakelta matrica X. Tačiau žymiai svarbesnis kriptografiniu požiūriu faktas yra tas, kad tikrosios matricos X, t.y. būtent tos reikšmės, kurią panaudojo Aldona sudarant viešąjį raktą, negalima pakeisti kita matrica $\tilde{X}$, tenkinančia jungtinumo lygtį, kadangi iššifravimo algoritmo 3) žingsnyje Aldona kelia matricą ${}^{V}Q^{UY}$ laipsniu X iš kairės. Tačiau vietoj X naudojant matricą $\tilde{X}$ nėra jokių garantijų, kad $\tilde{X}V = XV$, o tai reiškia, kad kenkėjas negali iššifruoti šifrogramos C.

Pasiūlytas protokolas turi panašumų su Anšelio-Anšelės-Goldfeldo protokolu. Kadangi Aldonos slaptojo rakto dalis – matrica U – yra gaunama skaičiuojant daugianario nuo matricos Z reikšmę, tai daugianarių aibę galima interpretuoti kaip viešąjį Aldonos pogrupį. Dar vienas panašumas yra tai, kad kenkėjui neužtenka žinoti informacijos apie matricą U, kadangi iššifruojant pranešimą Aldona apskaičiuoja daugianario nuo matricos $Y^{-1}ZY$ reikšmę. Nežinant šio slaptojo daugianario koeficientų kenkėjas negali iššifruoti šifrogramos C.

Taigi matome du galimus tiesioginio MLAŠ protokolo nulaužimo būdus. Pirmasis būdas yra toks:

1. Spręsti JPU matricos X atžvilgiu.
2. Spręsti MLF lygtį (4.9) kai matrica X yra žinoma.
3. Nustatyti ar gautoji matrica U yra daugianario nuo matricos Z reikšmė. Jei taip, rasti daugianario koeficientus.

Antrasis būdas atrodo taip:

1. Atsitiktinai pasirinkti daugianarį ir apskaičiuoti šio daugianario reikšmę matricai Z. Taip gaunama matrica U
2. Spręsti MLF lygtį (4.9) kai matrica U yra žinoma.
3. Patikrinti ar gautoji matrica X tenkina jungtinumo lygtį.

Norint apsisaugoti nuo šių nulaužimų reikia tinkamai pasirinkti saugumo parametrus. Šį klausimą nagrinėsime kitame skyriuje.

5.2. MLAŠ protokolo pavyzdys

Geresniam MLAŠ protokolo supratimui pateiksime paprastą pavyzdį. Pirmiausiai pasirinksime protokolo saugumo parametrų reikšmes $n = 15$ ir $m = 3$, t.y. platforminė multiplikacinė pusgrupė yra $\boldsymbol{M}_3(\boldsymbol{Z}_{15}^*)$ – 3-ios eilės kvadratinių matricų, kurių elementai priklauso grupei $\boldsymbol{Z}_{15}^* = \{1,2,4,7,8,11,13,14\}$, pusgrupė. Platforminei matricai Q apibrėžti mes naudosime tuos grupės $\boldsymbol{Z}_{15}^*$ elementus, kurių periodas yra lygus $\lambda(n)$ reikšmei. Šie elementai yra ciklinių pogrupių generatoriai 2, 7, 8, 13. Kadangi $\lambda(15) = 4$, tai laipsninis žiedas yra $\boldsymbol{M}_3(\boldsymbol{Z}_4)$. Viešai paskelbtos šios matricos:

$$Q = \begin{pmatrix} 8 & 13 & 2 \\ 2 & 7 & 7 \\ 13 & 13 & 8 \end{pmatrix}, \; Z = \begin{pmatrix} 3 & 0 & 3 \\ 2 & 2 & 3 \\ 1 & 1 & 2 \end{pmatrix}.$$

Aldona turi savo slaptą raktą $PrK_A = \{X, U\}$, čia matrica U gauta apskaičiavus daugianario

$$P_A(x) = x^2 + 2x + 3$$

reikšmę nuo matricos Z reikšmę, t.y $U = P_A(Z)$. Šios matricos atrodo taip:

$$X = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \end{pmatrix}, \; U = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Naudojant viešuosius duomenis Aldona apskaičiuoja savo viešąjį raktą $PuK_A = \{E, A\}$, čia

$$E = {}^X Q^U = \begin{pmatrix} 11 & 14 & 14 \\ 7 & 8 & 1 \\ 11 & 4 & 14 \end{pmatrix}, \; A = XZX^{-1} = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}.$$

Bronius naudoja Aldonos viešąjį raktą formuojant užšifravimo raktą. Pirmame žingsnyje jis atsitiktinai pasirenka neišsigimusią matricą Y. Ši matrica yra

$$Y = \begin{pmatrix} 3 & 3 & 1 \\ 0 & 2 & 1 \\ 0 & 3 & 0 \end{pmatrix}.$$

Naudodamas daugianarį $P_B(x) = 3x^2 + 1$ Bronius apskaičiuoja matricas $V = P_B(Z)$ ir $XVX^{-1} = P_B(A)$. Bronius gauna šias matricas:

$$V = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 2 \\ 1 & 0 & 3 \end{pmatrix}, \; XVX^{-1} = \begin{pmatrix} 2 & 3 & 1 \\ 1 & 3 & 3 \\ 1 & 0 & 1 \end{pmatrix}$$

Naudodamas turimą informaciją Bronius apskaičiuoja užšifravimo raktą K:

$$K = {}^{(XVX^{-1})}E^{Y} = \begin{pmatrix} 2 & 7 & 4 \\ 7 & 13 & 1 \\ 1 & 1 & 11 \end{pmatrix}.$$

Kadangi matricos K elementai k_{ij} priklauso grupei $\boldsymbol{Z}_{15}^{*}$, tai šiuos elementus galima užkoduoti naudojant 4 bitus. Tokiu atveju pranešimas M yra 3-ios eilės kvadratinė matrica, kurios elementai $m_{ij} \in \boldsymbol{Z}_{16}$. Tarkime, kad šis pranešimas atrodo taip:

$$M = \begin{pmatrix} 10 & 8 & 12 \\ 13 & 2 & 12 \\ 14 & 2 & 3 \end{pmatrix}.$$

Bronius užšifruoja pranešimą M naudodamas užšifravimo raktą K ir XOR operaciją. Rezultatas yra šifrograma C, kuri mūsų atveju atrodo taip:

$$C = K \oplus M = \begin{pmatrix} 8 & 15 & 7 \\ 10 & 15 & 13 \\ 15 & 3 & 8 \end{pmatrix}.$$

Bronius siunčia Aldonai šią šifrogramą kartu su savo viešuoju raktu $PuK_B = \{B, F\}$, čia matricos F ir B atrodo taip:

$$B = Y^{-1}ZY = \begin{pmatrix} 2 & 2 & 3 \\ 1 & 1 & 2 \\ 0 & 1 & 0 \end{pmatrix}, \; F = {}^{V}Q^{Y} = \begin{pmatrix} 7 & 4 & 4 \\ 8 & 4 & 14 \\ 11 & 14 & 11 \end{pmatrix}.$$

Aldona naudoja informaciją, kurią gavo iš Broniaus. Pirmame žingsnyje Aldona, naudodama Broniaus matricą B ir savo daugianarį $P_A(x)$ apskaičiuoja matricą $Y^{-1}UY = P_A(B)$:

$$Y^{-1}UY = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 3 & 1 \end{pmatrix}.$$

Dabar Aldona gali apskaičiuoti iššifravimo raktą. Jis atrodo taip:

$$K = {}^{X}F^{\left(Y^{-1}UY\right)} = \begin{pmatrix} 2 & 7 & 4 \\ 7 & 13 & 1 \\ 1 & 1 & 11 \end{pmatrix}.$$

Aldona iššifruoja šifrogramą C naudodama iššifravimo raktą K ir XOR operaciją. Rezultatas yra pranešimas M, kuris atrodo taip:

$$M = K \oplus C = \begin{pmatrix} 10 & 8 & 12 \\ 13 & 2 & 12 \\ 14 & 2 & 3 \end{pmatrix}.$$

Matome, kad pranešimas sėkmingai iššifruotas, nes užšifravimo ir iššifravimo raktai sutampa.

5.3. Diskretinio logaritmo ataka

Nagrinėdami MLAŠ tiesioginio nulaužimo būdus matome, kad kenkėjas nepriklausomai nuo pasirinkto būdo susiduria su laipsninių algebrinių lygčių sistemos sprendimo uždaviniu. Be to, kadangi vienas iš MLF privalumų yra atsparumas pilnojo perrinkimo atakai, kuris gali būti pasiektas pakankamai greitai, tai tiesioginis nulaužimas gali būti praktiškai neefektyvus. Kadangi mes nežinome laipsninių lygčių sistemų sprendimo algoritmų, tai kenkėjas turi ieškoti galimybių suvesti šią sistemą į tinkamesnę formą, t.y. į tokią formą, kad gautą ekvivalentų uždavinį galima būtų išspręsti naudojant jau žinomus algoritmus. Šiame skyrelyje aprašysime vieną iš tokių netiesioginių atakų. Pradėsime nuo matricinio diskretinio logaritmo funkcijos apibrėžimo, o vėliau šią funkciją pritaikysime mūsų protokolui.

Tegul matricos Q elementai priklauso ciklinei grupei $\boldsymbol{G}$, t.y. $\boldsymbol{S} = \boldsymbol{G}$. Tarkime, kad grupės $\boldsymbol{G}$ generatorius g yra žinomas. Diskretinį matricos Q logaritmą pagrindu g (žym. $\mathrm{ld}_g Q$) apibrėžkime taip [50]:

$$\mathrm{ld}_g Q = \{\mathrm{ld}_g q_{ij}\}, \tag{5.2}$$

t.y. skaičiuojant $\mathrm{ld}_g Q$ taikome diskretinio logaritmo funkciją kiekvienam matricos Q elementui q_{ij}. Kadangi visiems matricos Q^Y elementams galioja lygybės (2.9) ir (2.10), tai pritaikius matricinio diskretinio logaritmo funkciją lygybei (4.1) gauname:

$$\mathrm{ld}_g Q^Y = \left(\mathrm{ld}_g Q\right) \cdot Y = \mathrm{ld}_g C. \tag{5.3}$$

Jeigu laipsniniame žiede egzistuoja atvirkštinė matrica $(\mathrm{ld}_g\, Q)^{-1}$, tai padauginę abi (5.3) lygybės puses iš šios matricos gauname tokį rezultatą:

$$Y = \left(\mathrm{ld}_g\, Q\right)^{-1} \cdot \mathrm{ld}_g\, C\,. \tag{5.4}$$

Tokiu būdu gauname matricą *Y*. Analogiškai šią funkciją galima pritaikyti ir (4.9) lygybei. Tada turime tokį rezultatą:

$$\mathrm{ld}_g\left({}^{X}Q^{Y}\right) = X \cdot \left(\mathrm{ld}_g\, Q\right) \cdot Y = XTY = \mathrm{ld}_g\, E\,, \tag{5.5}$$

čia $T = \mathrm{ld}_g\, Q$.

Pavyzdys. Nagrinėkime multiplikacinę pusgrupę yra $\boldsymbol{M}_3(\boldsymbol{Z}_5^*)$. Kadangi $\lambda(5) = 4$, tai laipsninis žiedas yra $\boldsymbol{M}_3(\boldsymbol{Z}_4)$. Tarkime turime šias matricas:

$$Q = \begin{pmatrix} 3 & 3 & 2 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix},\ X = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \end{pmatrix},\ Y = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Tada gauname tokią matricą *E*:

$$E = {}^{X}Q^{U} = \begin{pmatrix} 1 & 4 & 4 \\ 2 & 3 & 1 \\ 1 & 4 & 4 \end{pmatrix}.$$

Pasirinkime ciklinės grupės $\boldsymbol{Z}_5^*$ generatorių $g = 2$ ir apskaičiuokime $\mathrm{ld}_2\, Q$ ir $\mathrm{ld}_2\, E$. Turime:

$$\mathrm{ld}_2\, Q = \begin{pmatrix} 3 & 3 & 1 \\ 1 & 1 & 1 \\ 3 & 3 & 3 \end{pmatrix},\ \mathrm{ld}_2\, E = \begin{pmatrix} 0 & 2 & 2 \\ 1 & 3 & 0 \\ 0 & 2 & 2 \end{pmatrix}.$$

Sudauginkime matricas *X*, $\mathrm{ld}_2\, Q$ ir *Y*. Sandaugos rezultatą redukuojame moduliu 4, kadangi šios matricos priklauso laipsniniam žiedui.Turime:

$$X \cdot \left(\mathrm{ld}_2\, Q\right) \cdot Y = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 & 3 & 1 \\ 1 & 1 & 1 \\ 3 & 3 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 & 1 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 2 \\ 1 & 3 & 0 \\ 0 & 2 & 2 \end{pmatrix}.$$

Matome, kad sandaugos rezultatas sutampa su matrica $\mathrm{ld}_2 E$. Taigi lygybė (5.5) galioja grupėje $\boldsymbol{Z}_5^*$.

Iš išraiškos (5.5) matome, kad naudojant diskretinio logaritmo funkciją laipsninių m^2 lygčių sistema yra transformuojama į daugelio kintamųjų kvadratinių (angl. multivariate quadratic – MQ) lygčių sistemą. Kadangi matome panašumų tarp dviejų uždavinių, tai MLAŠ saugumą mes vertiname nagrinėdami atitinkamos sistemos saugumo reikalavimus. Yra žinoma, kad bendras MQ spendžiamumo uždavinys virš bet kurio lauko priklauso NP-pilnųjų uždavinių klasei [9], [51]. Šiuo metu dažniausiai taikomi algoritmai MQ lygčių sistemoms spręsti yra paremti Grobnerio bazių teorija [52]. Tačiau yra laikoma, kad šie algoritmai yra praktiškai neefektyvūs tuo atveju, kai MQ sistemos yra atsitiktinai sugeneruotos virš lauko, jeigu lygčių ir nežinomųjų skaičius viršija 80 [53]. Akivaizdu, kad šis reikalavimas tenkinamas jau tuo atveju, kai $m = 9$. Šiuo atveju gauname 81 lygties sistemą su 162 nežinomaisiais. 2012 metais buvo parodyta, kad MQ lygčių sistemos sprendimo sudėtingumas eksponentiškai priklauso nuo lygčių skaičiaus [54]. Nepaisant to, kad mūsų atveju MQ lygčių sistema nėra atsitiktinė, mes neradome jokių metodų, naudojančių lygčių sistemos dėsningumus, kurie galėtų palengvinti šios sistemos sprendimą. Taigi galime suformuluoti tokį matricinį MQ uždavinio analogą [50]:

***Matricinis daugelio kintamųjų kvadratinių lygčių (angl. matrix multivariate quadratic equations - MMQ) uždavinys*:** Rasti matricas X ir Y, tenkinančias lygtį (5.5), kai matricos T ir $\mathrm{ld}_g E$ yra žinomos.

Taip pat reikia pabrėžti ir tą faktą, jog, nors mūsų atvėju MQ lygčių sistema yra apibrėžta virš žiedo $\boldsymbol{Z}_r$, čia $r = 2s$, tai nepalengvina šios sistemos sprendimo, kadangi naudojant kinų liekanų teoremą šis žiedas gali būti išskaidytas į dviejų laukų tiesioginę sandaugą.

Iš lygčių sistemos (4.20) apibrėžimo matome, kad ši sistema yra nepilnai apibrėžta (*angl.* underdefined system of equations), o tai reiškia, kad ši sistema gali turėti daugiau nei vieną sprendinį. Dėl šios priežasties net ir suradus vieną iš sistemos sprendinių, reikėtų įsitikinti ne tik tuo, kad jis atitinka tikrąją Aldonos (arba Broniaus) raktų porą, bet ir nustatyti daugianario $P_U(x)$ koeficientus. Yra žinomi keli polinominės eilės metodai nepilnai apibrėžtų MQ sistemoms spręsti [54], [55]. Tačiau šie metodai efektyviai veikia tik sistemoms, kuriose nežinomųjų skaičius ženkliai viršija lygčių skaičių. Dėl šios priežasties minėti metodai (arba jų modifikacijos laipsninių lygčių atvejui) negali būti taikomi lygčių sistemai (4.20) spręsti.

Kenkėjas gali nulaužti pasiūlytą protokolą, jeigu bus išspręstas MLF uždavinys arba jį atitinkantis MMQ uždavinys. Taigi matome, kad MLF uždavinio sudėtingumas yra ne mažesnis už MMQ uždavinio sudėtingumą. Tačiau MMQ uždavinys yra panašus į NP-pilnąjį MQ uždavinį. Dėl šios priežasties mes manome, kad abu aukščiau suformuluoti uždaviniai yra sudėtingi.

Nors MMQ uždavinys yra sudėtingas, mes norime išvengti MLF uždavinio suvedimo į MMQ uždavinį, nes manome, kad tokiu atveju MLF uždavinio sudėtingumas yra didesnis. Nagrinėkime kada toks suvedimas yra įmanomas.

Tegul $\boldsymbol{S} = \boldsymbol{Z}_n^*$ yra neciklinė multiplikacinė grupė, čia $n = pq$ yra sudėtinis skaičius, kuris yra dviejų pirminių skaičių p ir q sandauga. Tada, remiantis kinų liekanų teoremos išvada, egzistuoja izomofrizmas iš grupės $\boldsymbol{Z}_n^*$ į multiplikacinių grupių $\boldsymbol{Z}_p^*$ ir $\boldsymbol{Z}_q^*$ tiesioginę sandaugą $\boldsymbol{Z}_p^* \times \boldsymbol{Z}_q^*$. Kadangi grupės $\boldsymbol{Z}_p^*$ ir $\boldsymbol{Z}_q^*$ yra ciklinės, tai multiplikacinė grupė $\boldsymbol{Z}_p^* \times \boldsymbol{Z}_q^*$ yra izomorfinė adicinei grupei $\boldsymbol{Z}_{(p-1)} \times \boldsymbol{Z}_{(q-1)}$, kai izomorfizmas φ apibrėžiamas taip [50]:

$$\phi : \left(g_p^a ; g_q^b \right) \xrightarrow{\phi} (a;b), \tag{5.6}$$

čia g_p ir g_q yra atitinkamų grupių generatoriai.

Tačiau tokiu atveju ir pradinė grupė yra $\boldsymbol{Z}_n^*$ yra izomorfinė adicinei grupei $\boldsymbol{Z}_{(p-1)} \times \boldsymbol{Z}_{(q-1)}$, o tai reiškia, kad naudojant izomorfizmą φ galima apskaičiuoti matricos Q diskretinį logaritmą, jeigu jos elementai yra pasirenkami iš $\boldsymbol{Z}_n^*$. Kadangi matricinis diskretinis logaritmas yra apibrėžtas multiplikacinėse grupėse $\boldsymbol{Z}_p^*$ ir $\boldsymbol{Z}_q^*$, tai taikant kinų liekanų teoremą matrica Q yra vaizduojama į matricų porą (Q_p, Q_q), o $\mathrm{ld}_g Q$, kai $g = (g_p ; g_q)$, apskaičiuojamas taip:

$$\mathrm{ld}_g Q = \left(\mathrm{ld}_{g_p} Q_p ; \mathrm{ld}_{g_q} Q_q \right). \tag{5.7}$$

Matome, kad tokiu atvėju MLF uždavinio sudėtingumas yra nusakomas dviejų MMQ uždavinių sudėtingumu, kai šie uždaviniai yra apibrėžti atitinkamose grupėse $\boldsymbol{Z}_p^*$ ir $\boldsymbol{Z}_q^*$.

Pavyzdys. Nagrinėkime multiplikacinę pusgrupę yra $\boldsymbol{M}_3(\boldsymbol{Z}_{15}^*)$. Kadangi $\lambda(15) = 4$, tai laipsninis žiedas yra $\boldsymbol{M}_3(\boldsymbol{Z}_4)$. Tarkime turime šias matricas:

$$Q = \begin{pmatrix} 8 & 13 & 2 \\ 2 & 7 & 7 \\ 13 & 13 & 8 \end{pmatrix}, \; X = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \end{pmatrix}, \; Y = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Tada gauname tokią matricą E:

$$E = {}^{X}Q^{U} = \begin{pmatrix} 11 & 14 & 14 \\ 7 & 8 & 1 \\ 11 & 4 & 14 \end{pmatrix}.$$

Naudojant kinų liekanų teoremą vaizduokime matricą Q į matricas Q_5 ir Q_3, o matricą E į matricas E_5 ir E_3, kadangi grupė $\boldsymbol{Z}_{15}^*$ yra izomorfinė grupei $\boldsymbol{Z}_5^* \times \boldsymbol{Z}_3^*$. Turime:

$$Q_5 = \begin{pmatrix} 3 & 3 & 2 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix},\ Q_3 = \begin{pmatrix} 2 & 1 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix};$$

$$E_5 = \begin{pmatrix} 1 & 4 & 4 \\ 2 & 3 & 1 \\ 1 & 4 & 4 \end{pmatrix}, E_3 = \begin{pmatrix} 2 & 2 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

Pasirinkime ciklinės grupės $\boldsymbol{Z}_5^*$ generatorių $g_5 = 2$. Jau įsitikinome (žr. pavyzdį aukščiau), kad lygybė (5.5) galioja grupėje $\boldsymbol{Z}_5^*$. Ciklinė grupė $\boldsymbol{Z}_3^*$ turi vienintelį generatorių $g_3 = 2$. Apskaičiuokime $\mathrm{ld}_2 Q_3$ ir $\mathrm{ld}_2 E_3$. Turime:

$$\mathrm{ld}_2 Q_3 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ \mathrm{ld}_2 E_3 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Sudauginkime matricas X, $\mathrm{ld}_2 Q_3$ ir Y. Sandaugos rezultatą redukuojame moduliu 2, kadangi $\lambda(3) = 2$.Turime:

$$X \cdot \left(\mathrm{ld}_2 Q_3\right) \cdot Y = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 & 1 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Matome, kad sandaugos rezultatas sutampa su matrica $\mathrm{ld}_2 E_3$. Taigi lygybė (5.5) galioja grupėje $\boldsymbol{Z}_3^*$, o tuo pačiu ir grupėje $\boldsymbol{Z}_5^* \times \boldsymbol{Z}_3^*$, kuri yra izomorfinė grupei $\boldsymbol{Z}_{15}^*$.

Iš šio pavyzdžio matome, kad norint išvengti MLF uždavinio suvedimo į MMQ uždavinį, mūsų pirmasis žingsnis yra panaikinti izomorfizmą φ tarp dviejų multiplikacinių grupių. Tada reikia parinkti kitokią platforminę algebrinę struktūrą – multiplikacinę pusgrupę, kuriai neegzistuotų izomorfizmo, skaidančio ją į kelių ciklinių grupių tiesioginę sandaugą. Tokiu atvėju matricinis diskretinis logaritmas būtų neapibrėžtas, nes platforminė grupė neturėtų generatorių [50].

Platforminę pusgrupę, tenkinančią sąlygas, gauname naudojant dviejų grupių $\boldsymbol{Z}_n^*$ ir $Id_q(\boldsymbol{Z}_n) = \{j = i \cdot q;\ i = 1, \ldots, p-1\}$ sąjungą. Šią pusgrupę žymėsime $\boldsymbol{Z}_n^{\#}$. Taigi

$$\boldsymbol{Z}_{\boldsymbol{n}}^{\#} = \boldsymbol{Z}_{\boldsymbol{n}}^{*} \cup Id_q\left(\boldsymbol{Z}_{\boldsymbol{n}}\right). \tag{5.8}$$

Tačiau, nors diskretinis logaritmas negali būti apibrėžtas pusgrupėje $\boldsymbol{Z}_{\boldsymbol{n}}^{\#}$, galima išnaudoti kitą algoritmo silpną vietą – vienintelį jungtinumo apribojimą. Parodysime, kad ir šiuo atvėju MLF uždavinys gali būti suvestas į MMQ uždavinį tiesiogiai netaikant disktetinio logaritmo (4.9) lygčiai. Suvedimas yra įmanomas naudojant tą faktą, jog $Id_q(\boldsymbol{Z}_{\boldsymbol{n}})$ yra ciklinė grupė, kurios eilė $|Id_q(\boldsymbol{Z}_{\boldsymbol{n}})| = (p - 1)$.

Nagrinėkime Aldonos viešąjį raktą $PuK_A = \{XZX^{-1} = A, {}^{X}Q^{U} = E\}$. Matricą E keliame matriciniu laipsniu A iš kairės ir matriciniu laipsniu Z iš dešinės. Gauname tokį rezultatą [50]:

$${}^{A}E^{Z} = {}^{AX}Q^{UZ} = {}^{XZ}Q^{ZU}, \tag{5.9}$$

kadangi $UZ = ZU$ ir $AX = XZ$. Bet tada turime žinomą matricą ${}^{Z}Q^{Z}$, kurios visi elementai priklauso $Id_q(\boldsymbol{Z}_{\boldsymbol{n}})$. Žymėkime $P = {}^{Z}Q^{Z}$ ir $H = {}^{A}E^{Z}$. Tada turime:

$${}^{X}P^{U} = H. \tag{5.10}$$

Kadangi matricų P ir H elementai priklauso ciklinei grupei $Id_q(\boldsymbol{Z}_{\boldsymbol{n}})$, tai (5.10) lygčiai galima pritaikyti matricinio diskretinio logaritmo funkciją. Rezultatas yra toks:

$$X \cdot \left(\mathrm{ld}_g P\right) \cdot U = \mathrm{ld}_g H \tag{5.11}$$

Matome, kad, nors mes netaikėme matricinio diskretinio logaritmo funkciją matricai Q, vėl gavome MMQ uždavinį matricų X ir U atžvilgiu, kuris atitinka pradinį MLF uždavinį. Taigi suvedimas į MMQ uždavinį yra įmanomas ir tuo atveju, kai platforminėje pusgrupėje $\boldsymbol{Z}_{\boldsymbol{n}}^{\#}$ diskretinio logaritmo apibrėžti negalima. Kyla klausimas: ar bet kokia (5.11) lygties sprendinių pora $\left(\tilde{X}, \tilde{U}\right)$ taip pat tenkina ir pradinę (4.9) lygtį, t.y. ar išsprendus (5.11) lygtį mes galime nulaužti siūlomą protokolą? Norint atsakyti į šį klausimą reikia išnagrinėti du atvejus:

- Matrica Z yra neišsigimusi;
- Matrica Z yra išsigimusi.

Nagrinėjant pirmąjį atvejį mes galime pakelti (5.10) lygtį matriciniais laipsniais Z^{-1} iš kairės ir A^{-1} iš dešinės. Tokiu būdu gauname (4.9) lygtį. Aišku, kad atvirkštinė matrica A^{-1} egzistuoja, kadangi matrica A yra panaši į matricą Z. Taigi (5.11) lygties sprendinių pora taip pat tenkina ir (4.9) lygtį nepriklausomai nuo platforminės matricos Q parinkimo ir diskretinio logaritmo egzistavimo.

Nagrinėjant antrąjį atvejį pakelti (5.10) lygtį matriciniais laipsniais Z^{-1} ir A^{-1} negalime, nes šie laipsniai neegzistuoja. Tačiau tokiu atvėju matricą Z galima pakeisti matrica $\tilde{Z} = aZ + bI$, čia I – vienetinė matrica. Tada galima rasti tokius koeficientus a

ir b, kad egzistuotų $\tilde{Z}^{-1}$. Aišku, kad $\tilde{Z}$ komutuoja su Z ir U. Taip pat nesunkiai galima įsitikinti tuo, kad $X\tilde{Z} = \tilde{A}X$, čia $\tilde{A} = X\tilde{Z}X^{-1}$. Taigi piktavalis gali panaudoti matricas $\tilde{Z}$ ir $\tilde{A}$ ir atlikti suvedimą į MMQ uždavinį. Gavome, kad suvedimas į MMQ uždavinį nepriklauso nuo matricos Z.

Šią ataką toliau vadinsime *diskretinio logaritmo ataka* (DLA) [50].

Nagrinėkime atvejį, kai vietoj vieno yra naudojami du jungtinumo apribojimai. Tarkime, turime dvi tarpusavyje nekomutuojančias matricas Z_1 ir Z_2 bei du jungtinumo apribojimus $XZ_1X^{-1} = A_1$ ir $XZ_2X^{-1} = A_2$. Matrica U yra formuojama naudojant atsitiktinę funkciją $f_U(\ )$ nuo matricų Z_1 ir Z_2, t.y. $U = f_U(Z_1, Z_2)$. Bendruoju atvėju neįmanoma rasti netrivialios matricos (t.y. matricos, kuri nėra lygi bI), kuri komutuotų su matrica U. Tokiu atvėju suvedimas į MMQ uždavinį yra neįmanomas, jeigu neegzistuoja nei vienos iš matricų ${}^{Z_1}Q$, ${}^{Z_2}Q$, Q^{Z_1} ir Q^{Z_2} diskretinis logaritmas. Šią sąlygą galima užtikrinti pasirinkus vieną matricos Q elementą iš platforminės pusgrupės $\boldsymbol{Z_n}^{\#}$ idealo $Id_q(\boldsymbol{Z_n})$ generatorių aibės, o likusiuos – iš pusgrupės $\boldsymbol{Z_n}^{\#}$ pogrupio $\boldsymbol{Z_n}^{*}$ generatorių aibės.

Matome, kad norint išvengti DLA reikia turėti kelis jungtinumo apribojimus. Taip pat svarbus yra ir tinkamos platforminės pusgrupės parinkimas. Kitame skyrelyje mes pasiūlysime kitokią platfrominę grupę, kuri labiau tinka mūsų algoritmui pritaikyti.

5.4. Platforminės grupės parinkimas naudojant Sylovo grupes

Nagrinėkime grupę $\boldsymbol{Z_n}^{*}$. Didžiausias šios grupės elementų periodas yra $\lambda(n)$. Matome, kad šis skaičius akivaizdžiai yra sudėtinis. Taip pat yra aišku, kad didžiausia galima (5.12) lygties dėmens q_{ij}^{α} entropija gaunama tuo atvėju, kai elementas q_{ij} yra grupės $\boldsymbol{Z_n}^{*}$ generatorius. Šie samprotavimai yra teisingi ir tuo atvėju, kai vietoj $\boldsymbol{Z_n}^{*}$ yra naudojama ciklinė grupė $\boldsymbol{G}$. Todėl, siekiant užtikrinti didžiausią entropiją, t.y. norit, kad ML eksponentės E elementai būtų kiek įmanoma skirtingi, mes siūlome naudoti platforminės grupės $\boldsymbol{G}$ generatorius kaip matricos Q elementus. Toks siūlymas taip pat yra pagrįstas MLF statistinėmis savybėmis, kurios buvo aprašytos aukščiau.

Tegu grupės $\boldsymbol{Z_n}^{*}$ parametras n yra sudėtinis skaičius, šio skaičiaus Karmaiklo funkcija $\lambda(n) = pq$, čia p – nelyginis pirminis skaičius, o q tenkina salygą $\gcd(p, q) = 1$. Pagal Sylovo teoremą grupėje $\boldsymbol{Z_n}^{*}$ egzistuoja ciklinis Sylovo pogrupis, kurio eilė yra p [8]. Šį pogrupį žymėsime $\boldsymbol{\Gamma_{p,n}}$. Kadangi pagal Lagranžo teoremą elemento $\gamma \in \boldsymbol{\Gamma_{p,n}}$ periodas dalija šios grupės eilę, tai elemento γ periodas yra 1 arba p. Tai reiškia, kas visus šios grupės elementus galima suskirstyti į generatorius ir idempotentus. Šią grupę galima panaudoti siekiant didžiausios entropijos matricai E. Tačiau iš straipsnyje [49] gautų rezultatų matome, kad tokia grupė nėra atspari DLA. Dėl šios priežasties reikia suformuoti pusgrupę, panašią į $\boldsymbol{Z_n}^{\#}$.

Naują pusgrupę pradėsime formuoti fiksavus kokį nors atsitiktinį pirminį skaičių p. Tada mūsų užduotis yra, rasti tokią parametro n reikšmę, kad vienas iš Karmaiklo funkcijos $\lambda(n)$ paprastų pirminių daugiklių būtų lygus fiksuotam skaičiui p. Pagal Sylovo teoremą tokiu atveju grupėje $\boldsymbol{Z}_n^*$ egzistuoja ciklinis pogrupis $\boldsymbol{\Gamma}_{p,n}$, kurio galia $|\boldsymbol{\Gamma}_{p,n}| = p$. Tai reiškia, kad visi šio pogrupio elementai, išskyrus vienetą, turi periodą p, t.y. šie elementai yra pogrupio generatoriai. Pogrupiui sudaryti užtenka surasti vieną generatorių γ. Šio elemento paiešką galima atlikti skaičiuojant $a^p \bmod n$ grupės $\boldsymbol{Z}_n^*$ elementams a. Kai bus rastas pirmas elementas a, tenkinantis sąlygą $a^p \bmod n \equiv 1$, paiešką galima baigti. Tokiu atveju $\gamma = a$.

Raskime žiedo $\boldsymbol{Z}_n$ idempotentą j. Naudojant šį idempotentą galima suformuoti naują ciklinę grupę $\boldsymbol{J}_{p,n} = j\boldsymbol{\Gamma}_{p,n}$, kurios neutralusis elementas yra idempotentas j, o kitų elementų periodas yra p. Tada mes galime suformuoti naują pusgrupę $\Gamma^{\#}_{p,n}$ naudojant grupių $\boldsymbol{\Gamma}_{p,n}$ ir $\boldsymbol{J}_{p,n}$ sąjungą, t.y.

$$\Gamma^{\#}_{p,n} = \Gamma_{p,n} \bigcup \boldsymbol{J}_{p,n} \tag{5.13}$$

Šią pusgrupę mes naudosime platforminei pusgrupei formuoti, kadangi šiuo atveju išvengiama tiesioginio diskretinio logaritmavimo. Taip yra todėl, kad $\boldsymbol{J}_{p,n}$ yra pusgrupės $\Gamma^{\#}_{p,n}$ idealas. Matome, kad lyginant su $\boldsymbol{Z}_n^{\#}$ šiuo atveju nereikalingi papildomi apribojimai matricos Q elementams.

Formuojant $\Gamma^{\#}_{p,n}$ du svarbūs žingsniai yra tinkamo parametro n ir idempotento j paieška. Pradėsime nuo parametro n paieškos. Iš Karmaiklo funkcijos apibrėžimo matome, kad kai $\alpha_i = 1$, t.y. paprasto skaičiaus n daugiklio p_i atveju $\lambda(p_i) = p_i - 1$, o $\lambda(n)$ reikšmė yra lygi mažiausiam bendram kartotiniui nuo visų šio skaičiaus daugiklių. Norint rasti mažiausią skaičiaus n reikšmę, turime pasirinkti tokią šio parametro reikšmę, kad jis turėtų du paprastus pirminius daugiklius p_1 ir p_2. Vienas iš šių skaičių (pvz. p_1) turi būti toks, kad iš ansto fiksuotas pirminis skaičius p būtų paprastas skaičiaus $(p_1 - 1)$ daugiklis. Iš čia gauname pirminio skaičiaus p_1 išraišką

$$p_1 = pk + 1, \tag{5.14}$$

čia k yra mažiausias lyginis skaičius, su kuriuo ši salyga yra tenkinama. Tokį pasirinkimą lemia tas faktas, kad lauke $\boldsymbol{Z}_{p_1}$ neegzistuoja netrivialus idempotentas j. Siekiant minimizuoti parametro n reikšmę mes turime pasirinkti mažiausią galimą daugiklį p_2, kad egzistuotų mūsų salygas tenkinantis idempotentas. Dėl šios priežasties pasirenkame $p_2 = 3$. Dabar jau galime nesunkiai surasti idempotentą j, naudodami kinų liekanų teoremą. Kadangi elementas j yra žiedo $\boldsymbol{Z}_n$ idempotentas, jeigu tenkinama bet kuri iš sąlygų [1]:

$$\begin{cases} j \bmod p_1 \equiv 1 \\ j \bmod 3 \equiv 0 \end{cases} \tag{5.15}$$

arba

$$\begin{cases} j \bmod p_1 \equiv 0 \\ j \bmod 3 \equiv 1 \end{cases}, \tag{5.16}$$

tai turime pasirinkti tą iš elemento j reikšmių, kuri tenkina mūsų reikavimus pusgrupei $\Gamma^{\#}_{p,n}$ formuoti. Tačiau, kadangi pasirinkę j reikšmę pagal sąlygą (5.16), turime, jog dauginant jį iš kitų grupės $\boldsymbol{Z}_n^*$ elementų gauname tik du įmanomus sandaugos rezultatus j arba $2j$, tai keliant elementą ($j\gamma$) įvairiais laipsniais visuomet gauname tą patį rezultatą, t.y. $(j\gamma)^i = j$ bet kokiam laipsniui i. Taigi idempotentą j renkames pagal (5.15) salygą. Lentelėje 5.1 parodyta kaip atrodo pagrindiniai parametrai pusgrupei $\Gamma^{\#}_{p,n}$ sudaryti – pirminis skaičius p, sudėtinis skaičius n ir šio skaičiaus pirminis daugiklis p_1, ciklinio pogrupio $\boldsymbol{\Gamma}_{p,n}$ generatorius γ, ir idempotentas j. Atvejų $p = 2$ ir $p = 3$ nenagrinėsime, kadangi šių parametrų Sylovo pusgrupės $\Gamma^{\#}_{p,n}$ yra per mažos.

5.2. lentelė Pagrindiniai pusgrupės $\Gamma^{\#}_{p,n}$ parametrai.

p	5	7	11	13	17	19	23	29	31
n	33	87	69	159	309	573	141	177	933
p_1	11	29	23	53	103	191	47	59	311
γ	4	7	4	10	13	25	4	4	7
j	12	30	24	54	207	192	48	60	312

Pagrindinis pusgrupės $\Gamma^{\#}_{p,n}$ privalumas yra tas, kad visų elementų, kurie nėra idempotentai, periodas yra pirminis skaičius p. Tai reiškia, kad jungtinumo apribojimai yra apibrėžti virš lauko $\boldsymbol{Z}_p$. Be to kiekvienas pusgrupės $\Gamma^{\#}_{p,n}$ elementas (išskyrus vienetą ir idempotentą) generuoja lygiai p šios pusgrupės elementų.

Naudojant šio skyrelio rezultatus siūlomas protokolas buvo patobulintas. Šį protokolą vadinsime *patobulintu matricinio laipsnio asimetriniu šifravimu*.

5.5. Patobulintas MLAŠ protokolas

Protokolo viešieji parametrai yra matrica Q, kurios elementai priklauso pusgrupei $\boldsymbol{\Gamma}_{p,n}{}^{\#}$, bei dvi tarpusavyje nekomutuojančios matricos Z_1 ir Z_2, kurių elementai priklauso laukui $\boldsymbol{Z}_p$. Platforminė matrica Q yra pasirenkama taip, kad neegzistuotų nei vienos iš matricų ${}^{Z_1}Q$, ${}^{Z_2}Q$, Q^{Z_1} ir Q^{Z_2} diskretinis logaritmas. Aldona turi savo slaptą raktą $PuK_A = \{X, U\}$, čia matrica X atsitiktinai pasirenkama iš laipsninio žiedo taip, kad egzistuotų jos atvirkštinė matrica X^{-1}, o matrica U yra gaunama apskaičiuojant

atsitiktinai pasirinktos funkcijos $f_A(\ )$ nuo matricų Z_1 ir Z_2 reikšmę, t.y $U = f_U(Z_1, Z_2)$. Aldonos viešasis raktas yra $PuK_A = \{XZ_1X^{-1} = A_1,\ XZ_2X^{-1} = A_2,\ {}^{X}Q^{U} = E\}$. Bronius užšifruoja pranešimą M atlikdamas šiuos veiksmus:

1. Jis pasirenka neišsigimusią matricą Y, kurios elementai priklauso skaitiniam laukui $\boldsymbol{Z_p}$.
2. Bronius naudoja Aldonos viešą raktą:
    a) Pasirenka atsitiktinę dviejų kintamųjų funkciją $f_V(\ )$ ir apskaičiuoja $V = f_B(Z_1, Z_2)$.
    b) Bronius apskaičiuoja $XVX^{-1} = f_B(A_1, A_2)$ bei ${}^{V}Q^{Y}$. Jo viešas raktas yra $PuK_B = \{Y^{-1}Z_1Y = B_1,\ Y^{-1}Z_2Y = B_2,\ {}^{V}Q^{Y} = F\}$;
    c) Kelia matricą E laipsniu XVX^{-1} iš kairės;
    d) Gautą matricą kelia laipsniu Y iš dešinės.

Tokiu būdu Bronius gauna matricą $K = {}^{XV}Q^{UY}$. Kadangi šios matricos elementai yra atsitiktiniai ir pasiskirstę tolygiai, tai matrica K_B gali būti naudojama kaip šifravimo raktas pranešimui M užšifruoti.

1. Bronius užšifruoja pranešimą M naudojant šifravimo raktą K. Šifrograma $C = K \oplus M$
2. Bronius siunčia Aldonai gautą šifrogramą C ir savo viešą raktą PuK_B.

Aldona, turėdama Broniaus atsiustus duomenis, iššifruoja jo pranešimą M atlikdama šiuos veiksmus:

1. Naudojant matricas B_1 ir B_2 Aldona apskaičiuoja $Y^{-1}UY = f_A(B_1, B_2)$
2. Aldona kelia matricą F laipsniu $Y^{-1}UY$ iš dešinės;
3. Gautą matricą Aldona kelia laipsniu X iš kairės ir gauna iššifravimo raktą $K = {}^{XV}Q^{UY}$

Aldona iššifruoja Broniaus pranešimą M naudojant gautą iššifravimo raktą K, šifrogramą C bei tapatybę $K \oplus C = M$.

Matricinio laipsnio asimetrinio šifravimo protokolo saugumas paremtas prielaida, kad MLF yra vienkryptė funkcija bei dviem faktais, kuriuos pabrėšime dar kartą:

1) Tinkamai pasirinkus platforminę matricą Q kenkėjas negali pritaikyti diskretinio logaritmo funkcijos abiejų protokolo dalyvių viešiesiems raktams, t.y. negali suvesti MLF uždavinio į MMQ uždavinį tokiu būdu palengvinant protokolo kriptoanalizę. Taip yra todėl, kad DLA yra neefektyvi taikant ją (4.9) lygčiai, kadangi (5.5) lygtyje neegzistuoja $\mathrm{ld}_g\, Q$.
2) Naudojant specifinį matricų U ir V apskaičiavimo būdą išvengiama DLA šią funkciją taikant (5.10) lygčiai. Taip yra todėl, kad dėl to, jog matricos U, Z_1 ir Z_2 tarpusavyje nekomutuoja, (5.10) lygtyje negalima apskaičiuoti tokios matricos P, kuriai egzistuotų diskretinis logaritmas.

Kitame skyriuje aptarsime pagrindinius saugumo parametrus bei įvertinsime kokią įtaką šie parametrai turi protokolo saugumui. Tačiau prieš tai suformuluosime šio skyriaus išvadas ir rezultatus.

5.6. Išvados ir rezultatai

- Sudarytas originalus asimetrinio šifravimo protokolas, kurio saugumas yra paremtas MLF su papildomais jungtinumo apribojimais sprendimo sudėtingumu.
- MLF apgręžiamumas yra susietas su laipsninių lygčių sistemos (4.10) sprendimu, kurio sudėtingumas yra panašus į MQ lygčių sistemos sudėtingumą. Kadangi MQ lygčių sistemos sprendimo uždavinys priklauso NP-pilnųjų uždavinių klasei, tai galime daryti prielaidą, jog mūsų darbe nagrinėjamos laipsninių lygčių sistemos yra sudėtingos.
- Kenkėjas gali iššifruoti šifrogramą C tik tuo atveju, kai žino vieno iš vartotojų (pvz. Aldonos) tikrąjį slaptą raktą, t.y. tikrąją matricą X ir sugeba rasti daugianario koeficientus matricai U apskaičiuoti.
- Pirmoji MLAŠ protokolo versija nėra atspari DLA.
- Naudojant Sylovo grupių teoriją buvo suformuota nauja platforminė pusgrupė $\Gamma^{\#}_{p,n}$, kurią sudarantys elementai (išskyrus vienetą ir idempotentą) generuoja lygiai pusę visų šios pusgrupės elementų. Tokiu būdu yra išvengiama papildomų apribojimų matricai Q generuoti, tuo palengvinant generavimo algoritmą.
- Naudojant naują platforminę pusgrupę ir MLF saugumo analizės rezultatus MLAŠ protokolas buvo patobulintas taip, kad DLA būtų neefektyvi.

6. SAUGUMO PARAMETRŲ APIBRĖŽIMAS IR SAUGIŲ REIKŠMIŲ PARINKIMAS

6.1. MLAŠ saugumo parametrai

Iš skyrelio 5.1 gali atrodyti, kad pagrindiniai protokolo parametrai yra skaičius n, kuris apibrėžia necikliną multiplikacinę grupę $\boldsymbol{Z}_n^*$, ir kvadratinių matricų eilė – skaičius m. Patikslinkime šį rezultatą. Kadangi mes siekiame sumažinti multiplikacinės grupės eilę ir padidinti didžiausią įmanomą šios pusgrupės elementų skaičių periodą, tai pasirinksime parametrą $n = 3p$, čia $p = 2s + 1$ yra toks pirminis skaičius, kad skaičius s taip pat yra pirminis [49], [50]. Tokiu atveju $\lambda(n) = p - 1$, o tai reiškia, kad laipsninis žiedas yra apibrėžtas virš skaitinio žiedo $\boldsymbol{Z}_r$, čia $r = 2s$. Matome dar vieną šio protokolo trūkumą: parametro p reikšmei yra papildomi apribojimai.

Skyrelyje 4.2 jau minėjome, kad norint nulaužti MLF su jungtinumo apribojimais reikia išspręsti lygčių sistemą (4.20). Šio darbo autoriaus magistriniame darbe [45] buvo nagrinėjama panaši lygčių sistema

$$\begin{cases} Q^X = C \\ MX = XM \end{cases} \tag{6.1}$$

Kadangi tiesiogiai iš apibrėžimo turime, kad matricos C stulpeliai gali būti apskaičiuojami atskirai, tai lygties (4.1) sprendimas yra ekvivalentus lygčių sistemai, sudarytai iš m lygčių [45]:

$$Q^{X_{.j}} = C_{.j} \tag{6.2}$$

čia taškas reiškia, kad nagrinėjamas j-asis stulpelis. Tačiau, nors tai sumažina pilnojo perrinkimo variantų skaičių iki r^m, darbe [55] buvo parodyta, kad tinkamai pasirinkus matricų eilę m toks lygties (4.1) sprendimo būdas yra praktiškai neefektyvus.

Aišku, kad naudojant diskretinio logaritmo funkciją (5.2) MLF lygtis (4.9)virsta tiesine ir gali būti nesunkiai išspęsta, jeigu egzistuoja $\mathrm{ld}_g\, Q$ atvirkštinė matrica. Tačiau, nors diskretinio logaritmo funkcija suveda MLF lygtį (4.9) į MMQ lygtį skaitiniame žiede $\boldsymbol{Z}_r$, parametras p turi didesnę svarbą, negu parametras r, kadangi šis parametras tiesiogiai priklauso nuo parametro p. Taigi turime du pagrindinius MLAŠ saugumo parametrus – platforminio žiedo parametrą p ir matricų eilę m.

6.2. Saugių MLAŠ saugumo parametrų reikšmių parinkimas

Įveskime dar vieną svarbų parametrą – protokolo saugumo lygį L. Remiantis nusistovėjusia metodika šis parametras yra pasirenkamas iš anksto remiantis tokiais faktoriais kaip kompiuterinės įrangos galimybės, duomenų svarba ir aktualumas ir t.t. Vienas iš pagrindinių pasirinkimo faktorių yra matematinių operacijų, kurias reikia atlikti vykdant ataką prieš protokolą. Kadangi DLA tik supaprastina protokolo kriptoanalizę, tačiau nenulaužia paties protokolo, tai pagrindinė ataka prieš MLAŠ yra

pilnasis matricų perrinkimas. Remiantis šiuo faktu mes surišame protokolo saugumo lygį su (m – 1)-os eilės daugianarių virš skaitinio žiedo $\boldsymbol{Z}_r$ aibės galia. Fiksuokime šio parametro reikšmę ir nustatykime saugumo parametrų reikšmes, kurios atitinka pasirinktą saugumo lygį.

Nagrinėkime Aldonos slaptą raktą $PrK_A = \{X, U\}$. Kaip jau buvo nustatyta, matrica X pasirenkama atsitiktinai ir turi vienintelį apribojimą – atvirkštinės matricos egzistavimą. Matrica U komutuoja su viešai paskelbta matrica Z ir yra apskaičiuojama kaip daugianaris nuo šios matricos. Nustatant saugias parametrų p ir m reikšmes mes remsimės šiais bei žemiau pateiktais faktais [49]:

- Matricų, kurios komutuoja su viešai paskelbta laipsnine matrica Z, skaičius turi viršyti 2^L. Kiekviena tokia matrica turi būti apskaičiuojama kaip daugianaris nuo matricos Z.
- Matricų, kurios tenkina jungtinumo lygtį (4.21), skaičius turi viršyti 2^L.

Jeigu šie reikalavimai yra tenkinami, tai pilnas matricų X ir U perrinkimas MLF lygtyje (4.9) arba ją atitinkančioje MMQ lygtyje (5.11) yra neįmanomas. Reikia pabrėžti, kad šiuo metu mes nežinome jokių reikšmingai greitesnių už pilnąjį perrinkimą MMQ lygties (4.9) sprendimo būdų.

Pastebėkime, kad ta pati matrica Z yra naudojama ir jungtinumo lygčiai (4.21) ir komutatyvumo lygčiai

$$UZ = ZU\ . \tag{6.3}$$

Remiantis skyrelio 4.2 rezultatais turime, kad komutatyvumo lygtį tenkina r^m matricų, o jungtinumo lygtį – $r^{(m-1)}(s-1)$ matricų. Aišku, kad $r^m > r^{(m-1)}(s-1)$. Tada pagal mūsų reikalavimus:

$$r^{(m-1)}\left(s-1\right) \geq 2^L\ . \tag{6.4}$$

Tačiau abu parametrai $r = p - 1$ ir $s = \dfrac{p-1}{2}$ yra tiesiogiai surišti su parametru p. Tada turime:

$$\left(p-1\right)^{(m-1)}\left(\frac{p-3}{2}\right) \geq 2^L\ . \tag{6.5}$$

Tokiu būdu gauname matricos eilės m priklausomybę nuo parametro p:

$$m \geq \left\lceil \frac{\left(L+1\right)\ln 2 + \ln\left(p-1\right) - \ln\left(p-3\right)}{\ln\left(p-1\right)} \right\rceil \tag{6.6}$$

Pasirinkime saugumo lygio reikšmę $L = 80$. Ši reikšmė iki 2014 metų buvo laikoma saugia [20], todėl yra tinkama kaip minimalaus saugumo lygio riba.

Kadangi mes orientuojamės į ribotų resursų sistemas, tokias kaip 32 bitų mikroprocesoriai, tai turime pasirinkti parametrus taip, kad būtų išnaudojama kiek įmanoma mažiau įrenginio atminties ir skaičiavimo resursų. Naudojant mūsų protokolą įrenginio atmintyje yra saugojama ši informacija [56]:

- Daugybos ir kėlimo laipsniu lentelės, kurios yra naudojamos atliekant skaičiavimus platforminėje pusgrupėje;
- Sudėties ir daugybos lentelės, kurios yra naudojamos atliekant skaičiavimus laipsniniame žiede;
- Viešai paskelbta matrica Q, kuri priklauso platforminei pusgrupei;
- Viešai paskelbta matrica Z, kuri priklauso laipsniniam žiedui;
- Slaptoji matrica X, kuri priklauso laipsniniam žiedui ir daugianario koeficientų, kurie priklauso skaitiniam žiedui $\boldsymbol{Z}_r$, rinkinys. Ši informacija sudaro slaptąjį raktą;
- Slaptosios matricos ${}^{X}Q^{U} = E$ ir $XZX^{-1} = A$. Ši informacija sudaro slaptąjį raktą.

Kadangi dviejų skaičių sudėtis ir daugyba yra komutatyvios operacijos, tai sudarant šias lenteles skaitiniam žiedui $\boldsymbol{Z}_r$ nėra būtina saugoti visus lentelių elementus. Taigi turime $r(r + 1)/2$ elementų kiekvienoje iš šių lentelių. Taip pat turime $\frac{\varphi(n)\cdot\left(\varphi(n)+1\right)}{2}$ elementų multiplikacinės grupės $\boldsymbol{Z}_n^*$ daugybos lentelėje. Kadangi didžiausias multiplikacinės grupės $\boldsymbol{Z}_n^*$ elementų periodas yra r, tai kėlimo laipsniu šioje grupėje lentelę sudaro $\varphi(n) \cdot r$ elementų. Kiekvieną matricą sudaro m^2 elementų ir kiekvienas iš matricos elementų yra koduojamas $\lceil \log_2 n \rceil$ arba $\lceil \log_2 r \rceil$ bitais priklausomai nuo nagrinėjamos algebrinės struktūros. Nagrinėkime pirmas penkias parametro n reikšmes, tenkinančias mūsų keliamus reikalavimus. Šios reikšmės yra: 15, 21, 33, 69 ir 141. Pateikiame 6.1 lentelę, kurioje parodyta, kokią įtaką parametras n (o kartu ir parametras p) daro slaptojo ir viešojo raktų ilgiams bei bendriesiems atminties reikalavimams [56].

Kadangi protokolas turi du pagrindinius saugumo parametrus (p ir m), kurie turi tenkinti nelygybę (6.6), tai vienas iš šių parametrų turi būti pasirenkamas dėl kitų priežasčių. Kadangi mes orientuojamės į ribotų resursų sistemas, tai siūlome, kad parametras p būtų pasirenkamas vertinant bendruosius atminties reikalavimus bei sistemos skaičiavimo galimybes.

6.1. lentelė Slaptojo ir viešojo raktų ilgių bei bendrųjų atminties reikalavimų priklausomybė nuo parametro n

p	n	m	r	Raktų ilgiai bitais		Atminties reikalavimai bitais
				Slaptasis raktas	Viešasis raktas	
5	15	41	4	3444	10086	23928
7	21	32	6	3168	8192	20428
11	33	25	10	2600	6520	18000
23	69	19	22	1900	4332	26800
47	141	15	46	1440	3150	88792

Mes nagrinėjame pasiūlyto protokolo realizaciją 32 bitų mikroprocesoriuje. Kadangi visos aritmetinės operacijos yra atliekamos naudojant paiešką iš anksto sudarytose lentelėse, tai mes laikome šias operacijas elementariosiomis. Nagrinėkime dvi parametro p reikšmes: $p = 11$ ir $p = 47$. Pirmoji iš šių reikšmių buvo pasirinkta pagal bendruosius atminties reikalavimus. Iš lentelės matome, kad, kai $p = 11$, duomenims saugoti reikia mažiausiai atminties. Antroji reikšmė buvo pasirinkta pagal raktų ilgius. Iš lentelės matome, kad, kai $p = 47$, slaptojo ir viešojo raktų ilgiai yra trumpiausi. Mes įvertinome viršutinę atliekamų elementarių operacijų ribą šifravimo raktui generuoti MLAŠ protokolo antrame žingsnyje, t.y. vertinome šiuos veiksmus:

- Daugianario $P_V(XZX^{-1})$ skaičiavimas;
- Matricos E kėlimas laipsniu XVX^{-1} iš kairės;
- Gautos matricos kėlimas laipsniu Y iš dešinės.

Kadangi veksmams atlikti naudojame paieškos lenteles, tai skaičiuojant dviejų matricų daugyba ir vienpusės MLF atliekamų elementarių operacijų skaičius yra vienodas ir lygus $m^2(2m - 1)$. Skaičiuojant $(m - 1)$-os eilės daugianarį reikia apskaičiuoti visus matricos Z laipsnius nuo Z^2 iki $Z^{(m-1)}$, visas matricas padauginti iš koeficientų ir sudėti. Taigi bendras atliekamų elementarių operacijų skaičius daugianariui apskaičiuoti yra

$$m^2(2m-1)(m-2) + m^3 + m^2(m-1) = 2m^4 - 3m^3 + m^2. \qquad (6.7)$$

Kadangi skaičiuojant raktą reikia apskaičiuoti vieną daugianarį ir du kartus pakelti matricą laipsniu, tai bendras atliekamų elementarių operacijų skaičius šifravimo raktui apskaičiuoti yra

$$O(m) = 2m^4 + m^3 - m^2 \qquad (6.8)$$

Rezultatai parodė, kad atliekamų elementarių operacijų riba neviršija 8.0×10^5 kai $p = 11$ ir 1.04×10^5 kai $p = 47$. Matome, kad lyginant šiuos du atvejus elementariųjų operacijų skaičius yra 8 kartus mažesnis didesnei p reikšmei. Taigi didesnei p reikšmei reikia mažiau skaičiavimo resursų, tačiau daugiau atminties resursų. Iš lentelės matome, kad didesnei p reikšmei reikia 5 kartus daugiau atminties resursų.

Savo protokolą taip pat palyginome ir su kitais žinomais algoritmais. Mes remiamės skaičiavimo sąnaudų sąvoką, kurią suprantame kaip elementariųjų operacijų, atliekamų vidutinių galimybių mikroprocesoriuje, skaičių. Mes palyginsime du minėtus mūsų protokolo atvejus su tokių klasikinių algoritmų kaip El-Gamal-2048 bei elipsinių kreivių ECC-521 asimetrinių šifravimų realizacijomis 32 bitų mikroprocesoriuje.

Lyginant mūsų protokolą su El-Gamal algoritmu mes remiamės tuo, kad vidutiniškai dviejų 2048 bitų sveikųjų skaičių daugybai atlikti reikia 8191 elementarių operacijų. Tiek pat operacijų reikia atlikti ir keliant tokį skaičių kvadratu. Todėl vidutinis bendrasis elementariųjų operacijų kiekis, kurį reikia atlikti vykdant duomenų asimetrinį užšifravimą, yra apie 23.5×10^6. Matome, kad mažiausias vidutinis elementariųjų operacijų kiekis El-Gamal-2048 atvėju 235 kartus viršija mūsų protokolo atvejį, kai $p = 47$.

Lyginant mūsų protokolą su elipsinių kreivių asimetrinio šifravimo algoritmu ECC-521 mes remiamės tuo, kad dviejų kreivės taškų sudėtis gali buti atlikta naudojant 9 daugybos ir 5 kėlimo kvadratu operacijas [57]. Todėl vidutinis bendras elementarių operacijų kiekis atliekant dviejų elipsinės kreivės taškų sudėtį yra 8078. Taško dvigubinimas atliekamas naudojant 4 daugybos ir 4 kėlimo kvadratu operacijas. Šis veiksmas gali būti įvykdytas atlikus 4616 elementarias operacijas. Taigi vidutiniškai reikia atlikti apie 6.9×10^6 elementariųjų operacijų vykdant savo duomenų asimetrinį užšifravimą. Matome, kad tai yra 69 kartus daugiau, negu mūsų protokolo atvejis, kai $p = 47$.

Objektyvūs palyginimo rezultatai pateikti 6.2 lentelėje [56].

6.2. lentelė Asimetrinio šifravimo protokolų skaičiavimo sąnaudų palyginimas

Protokolas	Skaičiavimo sąnaudos (elem. op.)
El-Gamal-2048	23.5×10^6
ECC-521	6.9×10^6
Mūsų protokolas, $p = 11$	8.0×10^5
Mūsų protokolas, $p = 47$	1.04×10^5

Gautus palyginimo rezultatus galima paaiškinti tuo, kad naudojant El-Gamal-2048 arba ECC-521 algoritmus yra atliekami aritmetiniai veiksmai su dideliais sveikaisiais skaičiais. Nepaisant to, kad naudojant elipsines kreives skaičiai yra 4 kartus trumpesni už EL-Gamal-2048 protokole naudojamus skaičius, pačios operacijos yra žymiai sudėtingesnes, o jų skaičiavimo sąnaudos yra didesnes.

6.3. Patobulinto MLAŠ protokolo saugumo parametrai ir jų įvertinimas

Kaip jau minėjome aukščiau buvo atlikti pradinės protokolo versijos patobulinimai norint išvengti DLA. Tam tikslui buvo pasirinkta kita multiplikacinė pusgrupė ir pridėtas papildomas jungtinumo apribojimas. Šiame skyrelyje nagrinėsime kokią įtaką šie pasikeitimai turi pagrindiniams saugumo parametrams.

Patobulinto MLAŠ protokolo atveju remiantis Sylovo teorema ieškome pirminio periodo p pogrupio multiplikacinėje grupėje $\boldsymbol{Z_n}^*$. Naudojant šį pogrupį ir pusgrupės $\boldsymbol{Z_n}$ idempotentą sudarome multiplikacinę pusgrupę $\Gamma^{\#}_{p,n}$. Kadangi pries vykdant protokolą mes sudarome aritmetinių veiksmų lenteles pasirinktai pusgrupei, tai parametras n praranda savo svarbą, nes kiekvieną pusgrupės $\Gamma^{\#}_{p,n}$ elementą galima užkoduoti naudojant $\lceil \log_2 p \rceil + 1$ bitą. Taigi daugybos ir kėlimo laipsniu lentelių dydžiai priklauso tik nuo parametro p reikšmės. Parametras n šiems dydžiams įtakos neturi.

Nagrinėkime Aldonos slaptą raktą $PrK_A = \{X, U\}$. Šiuo atvėju matrica U jau nekomutuoja su viešai paskelbtomis matricomis Z_1 ir Z_2 ir yra apskaičiuojama kaip funkcija nuo šių matricų. Taigi nustatant saugias parametrų p ir m reikšmes mes remsimės šiais faktais bei tuo, kad matricų, kurios tenkina jungtinumo lygtį (4.21), skaičius turi viršyti 2^L. Ši sąlyga turi būti tenkinama abiems matricomis Z_1 ir Z_2.

Kadangi viešai paskelbtos matricos Z_1 ir Z_2 yra nekomutuojančios, tai daro įtaką ir parametro m reikšmei. Tokiu atvėju abi matricos turi būti panašios į skirtingas Žordano matricas, t.y. jų Žordano matricos turi būti sudarytos iš kelių kvadratinių Žordano langelių, turinčių formą (4.23). Šių langelių skaičių pažymėkime l, o langelių dydžius m_1, m_2, ..., m_l. Taigi šiuo atvėju atsiranda naujas parametras l, o Žordano matricos turi tokį pavidalą:

$$J_{1,2} = \left(\begin{array}{cccc:c:cccc} \mu_1 & 1 & & 0 & & & & & \\ & \mu_1 & \ddots & & & & & 0 & \\ & & \ddots & 1 & & & & & \\ 0 & & & \mu_1 & & & & & \\ \hdashline & & & & \ddots & & & & \\ \hdashline & & & & & \mu_l & 1 & & 0 \\ & 0 & & & & & \mu_l & \ddots & \\ & & & & & & & \ddots & 1 \\ & & & & & 0 & & & \mu_l \end{array}\right) \tag{6.9}$$

Dvi matricos tarpusavyje nekomutuoja jeigu jų Žordano matricose skiriasi Žordano langelių kiekiai arba šių langelių dydžiai. Jeigu visos tikrinės reikšmės μ_1, μ_2, ..., μ_l yra skirtingos, tai visos matricos, kurios komutuoja su matricomis J_1 ir J_2 turi tokį pavidalą:

$$\left(\begin{array}{cccc:c:cccc} a_1 & a_2 & \ddots & a_{m_1} & & & & & \\ & a_1 & \ddots & \ddots & & & & 0 & \\ & & \ddots & a_2 & & & & & \\ 0 & & & a_1 & & & & & \\ \hdashline & & & & \ddots & & & & \\ \hdashline & & & & & b_1 & b_2 & \ddots & b_{m_l} \\ & & & & & & b_1 & \ddots & \ddots \\ & 0 & & & & & & \ddots & b_2 \\ & & & & & 0 & & & b_1 \end{array}\right) \tag{6.10}$$

Be to šiuo atveju visos komutuojančios su J_1 arba J_2 matricos yra gaunamos apskaičiuojant daugianarius nuo atitinkamos matricos. Turėdami omenyje šį faktą bei tą faktą, kad Aldonos slaptojo rakto dalis – matrica *X* turi būti neišsigimusi, gauname matricos eilės *m* priklausomybę nuo parametro *p*:

$$p^{(m-l)}\left(p-1\right)^{l} \geq 2^{L} \tag{6.11}$$

Išlogaritmavus abi nelygybės (6.11) puses gauname:

$$m \geq \frac{L\ln 2 - l\ln\left(\dfrac{p-1}{p}\right)}{\ln p} \tag{6.12}$$

Kadangi $l \ln\left(\frac{p-1}{p}\right) < 0$, tai mes, norėdami sumažinti matricų eilę m, siūlome naudoti mažesnes parametro l reikšmes. Tačiau, reikia pabrėžti, kad mes turime pasirinkti $l \geq 2$, kadangi matricos Z_1 ir Z_2 yra nekomutuojančios. Taip pat, kadangi reiškinio $\ln\left(\frac{p-1}{p}\right)$ reikšmė yra artima 0, tai mes galime panaikinti narį $l \ln\left(\frac{p-1}{p}\right)$ nelygybėje (6.12). Tada gauname tokį parametro m įvertį:

$$m > \frac{L}{\log_2 p} \tag{6.13}$$

Iš matricos (6.10) išraškos matome, kad ši matrica turi lygiai m laisvųjų parametrų. Tai reiškia, kad iš viso matricų, kurios komutuoja su Žordano matricomis J_1 ir J_2 yra lygiai p^m. Kadangi tiek pat yra ir $(m - 1)$-os eilės daugianarių, tai aukščiau aprašytas pagrindinių parametrų bei viešųjų matricų Z_1 ir Z_2 pasirinkimas užtikrina didžiausią galimą komutuojančių su šiomis matricomis matricų entropiją. Taigi vienas iš galimų slaptosios funkcijos f pasirinkimų yra dviejų daugianarių nuo matricų Z_1 ir Z_2 sandauga, t.y. $f(Z_1, Z_2) = \boldsymbol{P_1}(Z_1) \cdot \boldsymbol{P_2}(Z_2)$.

Lyginant dvi protokolo versijas matome, kad šiuo atvėju reikia saugoti dvi viešai paskelbtas matricas Z_1 ir Z_2 vietoj vienos matricos Z bei dar vieną papildomą jugtinumo apribojimą. Taip pat, kadangi matricai U apskaičiuoti yra naudojami du daugianariai, tai reikia saugoti papildomą koeficientų rinkinį. Taip pat yra mažesnis skirtumas tarp multiplikacinės pusgrupės ir skaitinio žiedo elementų kiekio.

Kaip ir pirmos protokolo versijos atveju, mes siūlome pasirinkti parametero p reikšmę vertinant bendruosius atminties reikalavimus bei skaičiavimo sąnaudas. Pateikiame lentelę, kurioje parodyta, kokią įtaką parametras p daro slaptojo ir viešojo raktų ilgiams, bendriesiems atminties reikalavimams ir skaičiavimo sąnaudams, kai saugumo lygis $L = 80$.

Iš lentelės matome, kad atminties reikalavimai yra mažiausi, kai $p = 13$. Be to didėjant parametro p reikšmei atminties reikalavimai didėja. Šį faktą galima paaiškinti tuo, kad pasirinkus mažą p reikšmę didėja matricų eilė m, o kartu auga atminties reikalavimai matricoms saugoti. Pasirinkus didelę p reikšmę atminties reikalavimai matricoms saugoti mažėja, tačiau šiuo atvėju didėja aritmetinių operacijų lentelės, o tai sudaro didelę įtaką atminties reikalavimams. Tačiau raktų ilgiai ir skaičiavimos sąnaudos mažėja, kai parametro p reikšmės didėja. Taip yra todėl, kad šiems parametrams didžiausią įtaką sudaro matricų eilė m. Remiantis gautais rezultatais mes siūlome naudoti parametro p reikšmes $p = 23$ arba $p = 29$ vykdant patobulintą MLAŠ protokolą, kadangi pasirinkus vieną iš šių reikšmių gauname pusiausvyrą tarp atminties reikalavimų ir skaičiavimų sąnaudų.

6.3. lentelė Raktų ilgių, atminties reikalavimų ir skaičiavimo sąnaudų priklausomybė nuo parametro p

p	m	Raktų ilgiai bitais		Atminties reikalavimai bitais	Skaičiavimo sąnaudos
		Slaptasis raktas	Viešasis raktas		
5	35	3780	12250	28790	5915525
7	29	2610	8410	20410	2779505
11	24	2400	7488	20379	1298880
13	22	2024	6292	18781	915244
17	20	2100	6400	23468	623600
19	19	1900	5776	24130	507205
23	18	1710	5184	27672	407916
29	17	1530	4624	35486	323969
31	17	1530	4624	38988	323969

Kadangi elipsinių kreivių asimetrinis šifravimas ECC-521 reikalauja mažiau skaičiavimo sąnaudų už El-Gamal-2048 algoritmą, tai patobulintą MLAŠ protokolą palyginsime su šiuo protokolu. Iš 3 lentelės matome, kad vidutinis elementariųjų operacijų kiekis vykdant ECC-521 protokolą yra 17 kartų didesnis už MLAŠ-23 protokolą ir 21 kartą didesnis už MLAŠ-29 protokolą (čia skaičius reiškia parametro p reikšmę). Šiuos rezultatus galima paaiškinti tuo, kad elipsinių krievių operacijos yra sudėtingos, o MLAŠ protokolo atvėju operacijos yra atliekamos naudojant iš anksto apskaičiuotas lenteles.

Lyginant dviejų protokolo versijų atminties reikalavimus matome, kad rezultatai yra artimi tiems atvėjams, kai $r = p - 1$. Taip yra todėl, kad nepaisant to, kad patobulintoje protokolo versijoje yra naudojama papildoma matrica, multiplikacinę pusgrupę $\Gamma^{\#}_{p,n}$ sudaro mažiau elementų, negu grupė $\boldsymbol{Z_n}^*$, ir šie elementai yra koduojami naudojant mažesnį bitų kiekį. Tačiau patobulintos versijos skaičiavimo sąnaudos yra didesnės, kadangi yra apskaičiuojamas papildomas daugianaris.

6.4. Išvados ir rezultatai

- Nustatyti pagrindiniai MLAŠ protokolo saugumo parametrai, kurie yra multiplikacinės pusgrupės parametras p ir kvadratinių matricų eilė m.
- Matricų Z_1 ir Z_2 reikšmės yra saugios, jeigu šios matricos yra panašios į tarpusavyje nekomutuojančias Žordano matricas J_1 ir J_2.
- Nustatyta, kad siekiant sumažinti parametro m reikšmę geriausia naudoti Žordano matricas, sudarytas iš dviejų Žordano langelių. Matricų J_1 ir J_2 Žordano langelių dydžiai turi būti skirtingi.
- Nustatyta matricų eilės m priklausomybė nuo pasirinkto saugumo lygio ir parametro p.
- Apibrėžtas protokolo saugumo lygis L.
- Nustatytos saugios pagrindinių parametrų reikšmės, kai $L = 80$.
- Įvertinti MLAŠ protokolo raktų ilgiai ir bendrieji atminties reikalavimai duomenims saugoti, kai $L = 80$.
- Palyginta MLAŠ protokolo realizacijos greitaveika 32 bitų mikroprocesoriuje su klasikinių protokolų (El-Gamal-2048, ECC-521) protokolų realizacija elementariųjų operacijų atžvilgiu. Tyrimų rezultatai parodė, kad vykdant MLAŠ protokolą atliekama mažiau elementariųjų operacijų, negu vykdant klasikinius protokolus. Naudojant pirmąją protokolo versiją skaičiavimų sanaudos yra vidutiniškai 235 kartų mažesnės lyginant su El-Gamal-2048 protokolu ir 69 kartus mažesnės lyginant su ECC-521 protokolu. Naudojant patobulintą MLAŠ protokolo versiją skaičiavimų sanaudos yra 17 kartų mažesnės kai $p = 23$ ir 21 kartų mažesnės kai $p = 29$.
- Atlikti analogiški tyrimai patobulintam MLAŠ protokolui. Pagal gautus rezultatus pasiūlytos tokios parametro p reikšmės: $p = 23$ arba $p = 29$, kadangi naudojant šias reikšmes gauname pusiausvyrą tarp atminties reikalavimų ir skaičiavimų sąnaudų.

7. MLAŠ PROTOKOLO REALIZACIJA IR TYRIMAS

Praeitame skyriuje nustatėme pagrindinius MLAŠ protokolo parametrus bei suformulavome pagrindinius principus, pagal kuriuos yra generuojamos slaptos ir viešos matricos. Šiame skyriuje remiantis mūsų protokolu sudarysime MLAŠ agentą, t.y. programą, kuri galėtų aptarnauti sistemos klientus, t.y. leistų klientams saugiai užšifruoti ir iššifruoti duomenis. Naudojant sudarytus serverio ir klientų agentus atliksime eksperimentinius tyrimus. Gautus rezultatus palyginsime su kitų protokolų eksperimentinių tyrimų rezultatais.

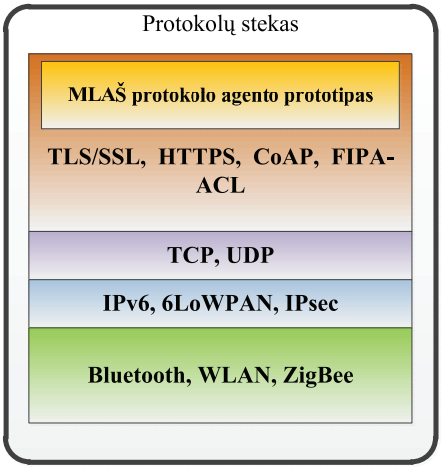
7.1. MLAŠ protokolo taikymas praktikoje

Protokolo realizacija yra naudojama projekte „Daiktų internetas“. Šio projekto tyrimų objektai yra fiziniai įrenginiai, kurie vadinami IoT daiktais. Kadangi daiktų internetas yra sudėtingas tinklas, kuriame yra naudojami įvairių rūšių komunikaciniai protokolai, tai pagrindiniai projekto uždaviniai apima tokius klausimus kaip:

- Komunikacinių protokolų tyrimai ir jų adaptacija išmaniosioms aplinkoms;
- Šių protokolų kriptografinės saugos metodų kūrimas ir tyrimas;
- Išmaniosios aplinkos daiktų integracijos steko prototipo sudarymas ir tyrimai.

Daiktų ir paslaugų interneto technologijos užtikrina išmaniųjų aplinkų sudarymą, o išmanioji aplinka turi intelekto elementų. Norint sukurti išmaniųjų aplinkų sprendimus, reikalingas daiktų internetas, kurio daiktai yra intelektualūs. Šiam tikslui pasiekti daiktuose turi būti įdiegti intelektualūs sprendimus priimantys ir tarpusavyje bendradarbiaujantys agentai.

Agentai žinių apsikeitimo procesui naudoja bendrą kalbą (žinių apsikeitimo protokolą), kuri ir nusako visas žinių apsikeitimui būtinas taisykles tarp skirtingų agentų. Svarbus išmaniųjų aplinkų daiktų interneto sistemų kūrimo aspektų yra šių sistemų sauga ir jos užtikrinimas. Kadangi išmaniųjų aplinkų daiktai neturi tokių atminties ir skaičiavimų resursų kaip dideli kompiuteriai, tai reikia vertinti ne tik komunikacinių protokolų saugą, bet ir tokius aspektus kaip energijos sąnaudos ir protokolo greitaveika. Remiantis teoriniais tyrimais, kurie buvo atlikti praeitame skyriuje galime teigti, kad MLAŠ protokolas turi potencialo būti panaudotas tokioms sistemos, kadangi tinkamai pasirinkus pagrindinius parametrus šis protokolas yra atsparus kriptografinėms atakoms ir naudoja mažiau elementariųjų operacijų, negu klasikiniai asimetrinio šifravimo algoritmai. Planuojama, kad MLAŠ protokolo agento prototipas bus panaudotas OSI modelio taikomajame sluoksnyje. Jo pagrindinė paskirtis yra saugiai persiusti duomenis apie įrenginio nustatymus, jo galimybes ir panašiai. Agento vieta yra parodyta paveiksle 7.1.



7.1 pav. Agento prototipas protokolų steke

Šiuo metu MLAŠ protokolo agento prototipo realizacijos darbai dar yra vykdomi. Agento diegimas į įrangą numatomas 2014 metų rudenį. Paprastumo dėlei buvo pasirinkta MLAŠ protokolo versija, kuri yra aprašyta straipsnyje [46]. Nuo patobulintos versijos, kuri yra aprašyta skyrelyje 5.5 ji skiriasi tuo, kad vietoj multiplikacinės Sylovo pusgrupės $\boldsymbol{\Gamma}_{p,n}^{\#}$ yra naudojama multiplikacinė pusgrupė $\boldsymbol{Z}_n^{\#}$. Šiuo metu planuojama, kad galutinė protokolo versija bus sudaryta naudojant patobulintą MLAŠ protokolą.

MLAŠ protokolo realizacija bei pagrindiniai algoritmai, kurie yra naudojami protokolui realizuoti yra aprašyti šio darbo priede.

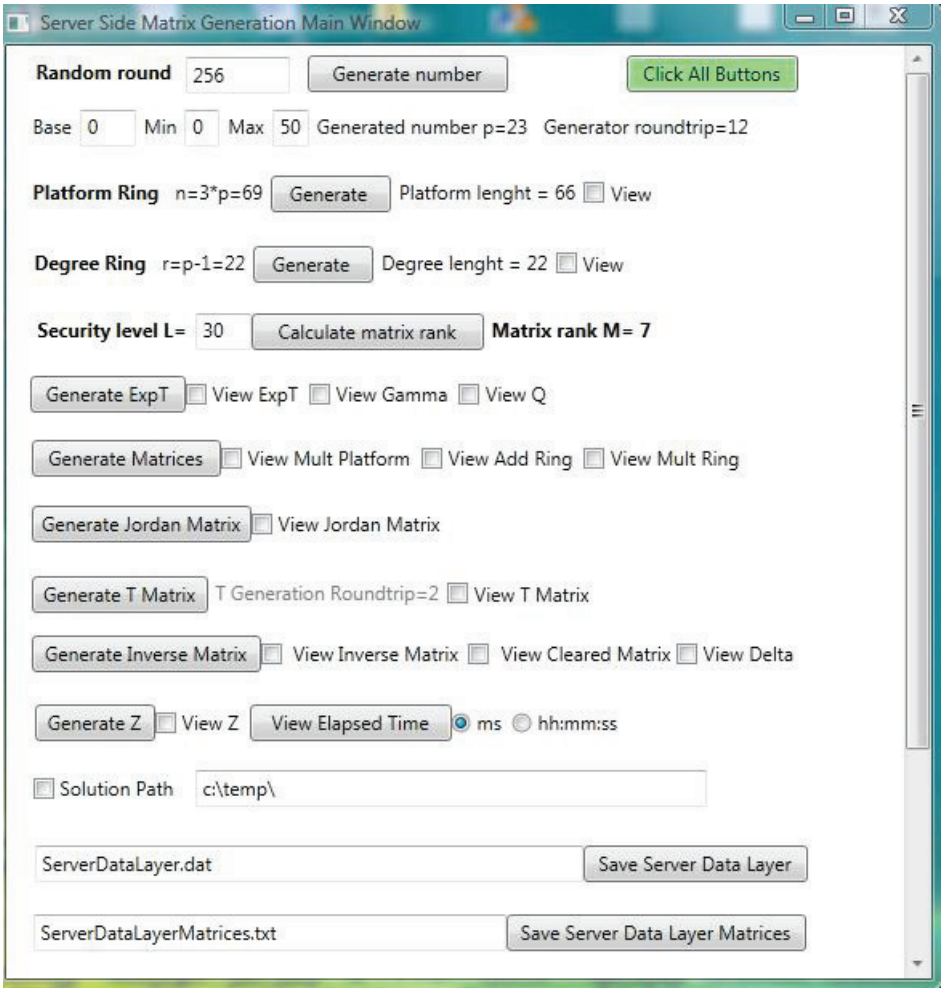
7.2. MLAŠ protokolo agentų aprašymas

Naudojant bendrą informaciją apie asimetrinių šifravimo sistemų realizaciją nutarta sukurti MLAŠ protokolo agento prototipe serverio ir klientų agentus. Toks pasirinkimas yra paremtas tuo, kad serverio pusėje atliekami vienkartiniai veiksmai, o klientas gauna jau paruoštą MLAŠ protokolui vykdyti reikalingą informaciją, kuri apima ne tik pagrindinius parametrus p ir m, saugumo lygį L bei viešąsias matricas Q, Z_1 ir Z_2, bet ir paieškos lenteles. Pagrindinė serverio agentų paskirtis yra sistemos parametrų, paieškos lentelių ir viešųjų matricų generavimas.

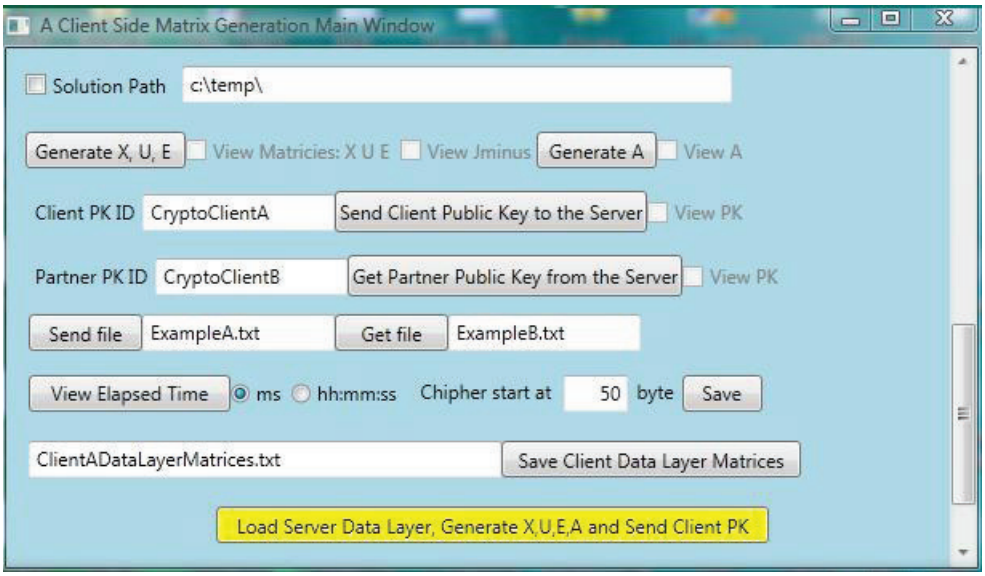
Pirmojo kliento agento naudojimo metu yra formuojamas vartotojo viešasis ir slaptasis raktai. Viešasis raktas yra saugojamas serverio duomenų bazėje ir pranešimo siuntėjas (šį kliento agentą vadinsime šifruotoju), naudodamas gavėjo identifikacijos duomenis, pasiima norimo agento (adresato) viešąjį raktą iš šios bazės. Kliento agentai dirba užšifravimo ir iššifravimo režimais ir yra atsakingi už slaptojo rakto saugojimą, informacijos siuntimą/gavimą ir jos užšifravimą/iššifravimą.

MLAŠ agento prototipo realizacijai buvo pasirinkta moderni Microsoft .NET Framework platforma, kuri pasižymi efektyviu resursų išnaudojimu. Ši platforma

puikiai tinka kurti taikomąsias programas, ribotus energijos išteklius ir skaičiavimų resursus turintiems įrenginiams naudojant mažesnes versijas .NET Compact Framework ir .NET Micro Framework. Naudojant šią platformą gauname suvienodintus programavimo bei resursu valdymo principus kaip serverio taip ir kliento pusėje. Protokolui realizuoti buvo pasirinkta C# programavimo kalba, kadangi ši kalba yra sukurta taip, kad būtų lengvai suprantama ir optimaliai eksploatuojama, atsižvelgiant į kompiuterinių resursų išteklius bei yra objektiškai orientuota, kas leidžia atlikti duomenų veiksmus objektų lygyje. Naudojant šią kalbą sukurti serverio ir du klientų agentai. Tyrimų tikslams yra padaryta grafinė sąsaja su vartotoju. Sukurtų programų langai yra pateikti žemiau:

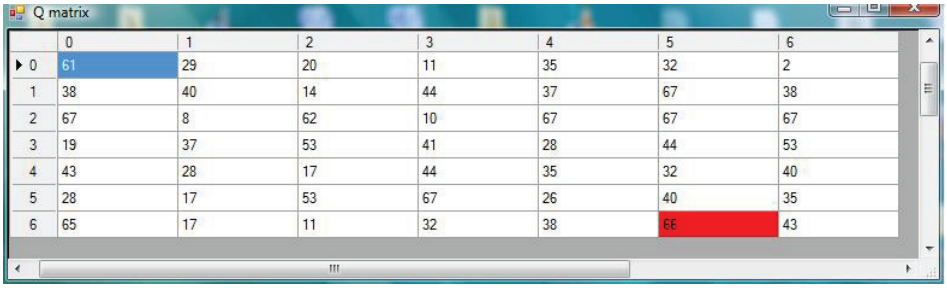


7.2 pav. Serverio agento langas.



7.3 pav. Kliento agento langas.

Naudojant serverio agentą yra generuojami pagrindiniai parametrai, paieškos lentelės ir matricos. Tam yra du būdai: generuoti duomenis pažingsniui spaudžiant atitinkamus mygtukus arba paspausti mygtuką Click All Buttons, kuri atlieka visus generavimus iš karto. Vartotojas gali pamatyti sugeneruotus duomenis pažymėjęs atitinkamą vėliavėlę (angl. checkbox). Matome, kad šiuo metu sugeneruoti parametrai $p = 23$, $n = 69$. Tam prireikė 12 bandymų. Pasirinktas saugumo lygis $L = 30$ yra pasiekiamas, kai yra naudojamos 7-os eilės matricos, t.y. $m = 7$. Pažymėjęs vėliavėlę View Gamma vartotojas gali pamatyti elementus, kurie yra naudojami generuojant matricą Q. Taip pat vartotojas gali pamatyti sugeneruotas matricas, pažymėjęs atitinkamą vėliavėlę. Čia pateiksime sugeneruotą matricą Q.

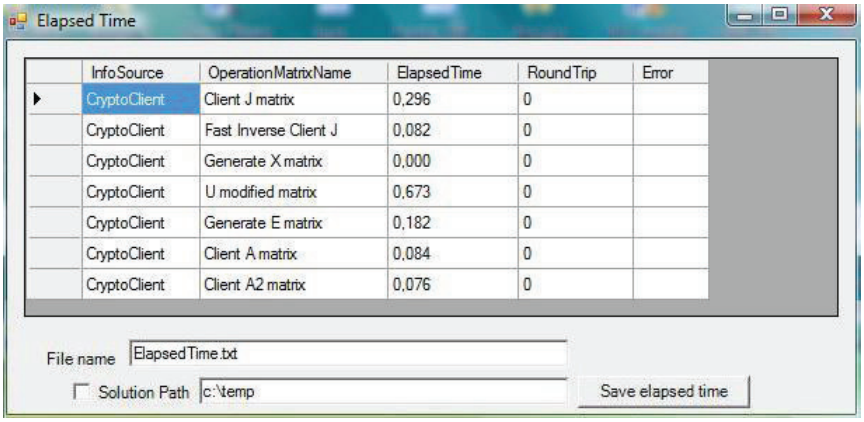


	0	1	2	3	4	5	6
0	61	29	20	11	35	32	2
1	38	40	14	44	37	67	38
2	67	8	62	10	67	67	67
3	19	37	53	41	28	44	53
4	43	28	17	44	35	32	40
5	28	17	53	67	26	40	35
6	65	17	11	32	38	66	43

7.4 pav. Sugeneruota platforminė matrica Q.

Paspaudęs mygtuką View Elapsed Time vartotojas mato kiek laiko užtruko duomenų generavimas. Nors atliekant bandymus nustatyta, kad patogiau sunaudotą laiką matuoti milisekundėmis, tačiau pagal vartotojo pageidavimą šiuos laikus galima

pateikti ir formatu valandos, minutės, sekundės. Tokiu atveju reikia pažymėti atitinkamą galimybę prieš generuojant duomenis. Šis langas yra pateiktas 7.5 paveiksle.

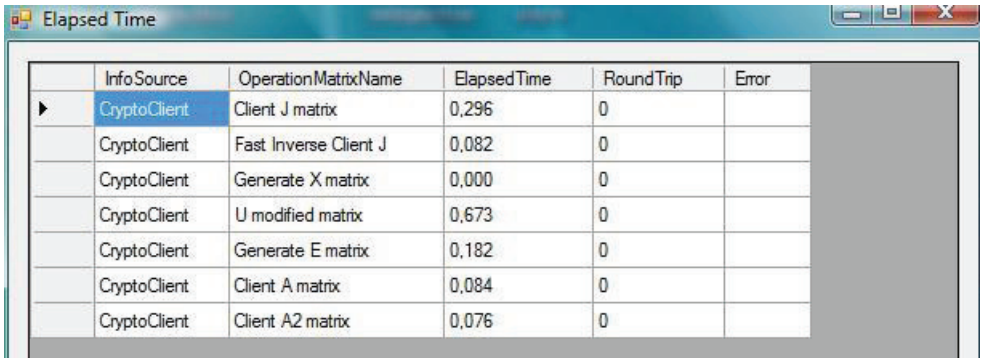


7.5 pav. Laiko sąnaudų vešiesiems duomenims generuoti suvestinė.

Iš pateiktos suvestinės matome, kad ilgiausiai užtruko kėlimo laipsniu lentelės formavimas. Tam buvo sunaudota 31,054 ms. Taip pat matome, kad generuojant neišsigimusią matricą *T* prireikė 5 bandymų. Bendras duomenų generavimo laikas yra 38,699 ms. Gautą informaciją apie laikus galima išsaugoti tekstiniame faile ElapsedTime.txt. Šis failas išsaugomas lange Solution Path nurodytoje direktorijoje. Failo struktūra yra tokia, kad jį galima būtų atidaryti naudojant Microsoft Excel programą.

Sukurti viešieji duomenys yra išsaugojami faile ServerDataLayer.dat paspaudus mygtuką Save Server Data Layer. Apie sėkmingą arba nesėkmingą duomenų išsaugojimą vartotoją informuoja atitinkamas pranešimas. Paspaudžius mygtuką Save Server Data Layer Matrices tekstiniame faile ServerDataLayerMatrices.txt išsaugojami visi sugeneruoti duomenys.

Kliento agentas pateikia užklausą serveriui ir gauna iš jo viešųjų duomenų failą. Paspaudęs geltoną mygtuką lango apačioje (žr. 7.3 pav.) vartotojas užkrauna šį duomenų failą bei formuoja savo raktus ir persiunčia sugeneruotą viešąjį raktą serveriui. Raktų generavimą ir viešojo rakto siuntimą galima atlikti ir pažingsniui spaudžiant atitinkamus mygtukus. Visas sugeneruotas matricas galima pamatyti pažymėjus atitinkamas vėliavėles. Taip pat yra galimybė stebėti laiko sąnaudas matricoms generuoti. Laiko sąnaudų suvestinė yra kviečiama mygtuku View Elapsed Time.



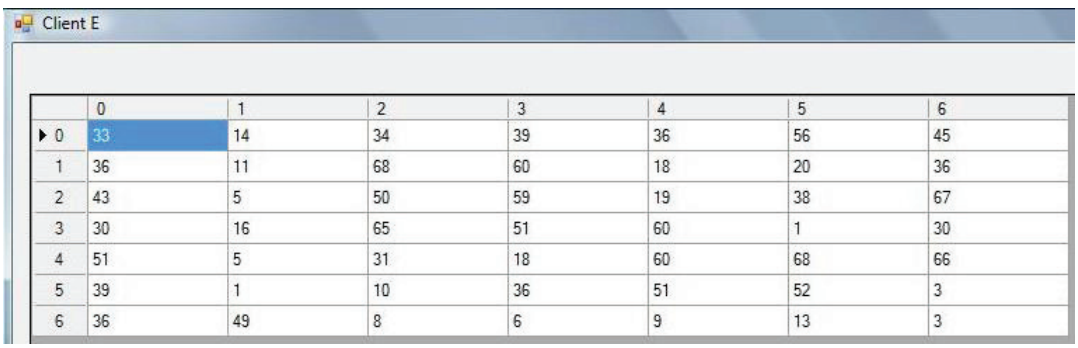
	InfoSource	OperationMatrixName	ElapsedTime	RoundTrip	Error
▶	CryptoClient	Client J matrix	0,296	0	
	CryptoClient	Fast Inverse Client J	0,082	0	
	CryptoClient	Generate X matrix	0,000	0	
	CryptoClient	U modified matrix	0,673	0	
	CryptoClient	Generate E matrix	0,182	0	
	CryptoClient	Client A matrix	0,084	0	
	CryptoClient	Client A2 matrix	0,076	0	

7.6 pav. Laiko sąnaudų generuojant kliento slaptąjį ir viešąjį raktus suvestinė.

Iš šios suvestinės matome, kad slaptasis raktas $PrK_A = \{X, U\}$ buvo suformuotas per 1,051 ms, kadangi šio rakto formavimo metu buvo atlikti šie veiksmai:

- Sugeneruota nauja Žordano matrica ir apskaičiuota jos atvirkštinė matrica. Šiems veiksmams sunaudoti laikai parodyti pirmoje ir antroje suvestinės eilutėse.
- Naudojant šias matricas ir serverio matricas T ir T^{-1} buvo sugeneruotos matricos X ir X^{-1}. Kadangi šios matricos buvo formuojamos tame pačiame metode, tai sunaudotas laikas šioms matricoms apskaičiuoti yra parodytas trečioje suvestinės eilutėje.
- Sugeneruoti du daugianarių koeficientų rinkiniai ir apskaičiuota matrica U. Sunaudotas šiems veiksmams laikas parodytas ketvirtoje suvestinės eilutėje.

Viešojo rakto apskaičiavimas užtruko 0,342 ms. Toks yra bendras matricoms A_1, A_2 ir E apskaičiuoti sunaudotas laikas. Sugeneruotas viešasis raktas $PuK_A = \{A_1, A_2, E\}$ įrašytas į failą CryptoClientA.key. Stebėkime ypatingą svarbą turinčią viešojo rakto matricą E ir slaptojo rakto matrica X ir U. Tą galima padaryti pažymėjus vėliavėlę View Matrices: X U E. Čia pateiksime tik matricą E.



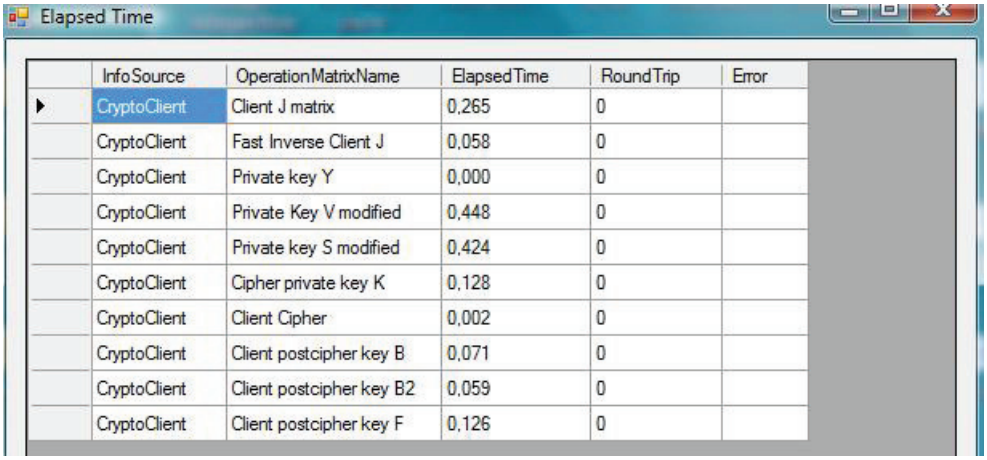
	0	1	2	3	4	5	6
▶ 0	33	14	34	39	36	56	45
1	36	11	68	60	18	20	36
2	43	5	50	59	19	38	67
3	30	16	65	51	60	1	30
4	51	5	31	18	60	68	66
5	39	1	10	36	51	52	3
6	36	49	8	6	9	13	3

7.7 pav. Sugeneruota kliento viešojo rakto matrica E.

Iš 7.7 paveikslo matome, kad ne visi sugeneruotos matricos E elementai priklauso multiplikacinės pusgrupės $\boldsymbol{Z}_{69}^{**}$ idealui. Šio fakto priežastis yra ta, kad sugeneruotos matricos X ir U turi nulinių elementų. Kadangi aprašinėdami algoritmus mes susitarėme, kad sąlyga $a^0 = 1$ galioja visiems pusgrupės $\boldsymbol{Z}_n^{\#}$ elementas, tai šios sąlygos naudojimas yra vienintelis būdas „išeiti“ iš idealo. Mūsų atveju taip atsitiko dėl to, kad matricos X elementas $x_{26} = 0$, kadangi vienintelis matricos Q idealo elementas yra $q_{65} = 66$. Pagal kairiosios MLF apibrėžimą šis elementas turėjo būti pakeltas nuliniu laipsniu.

Gautas rezultatas yra svarbus tuo, kad suteikia papildomą apsaugą nuo diskretinio logaritmavimo atakos, kadangi negalima surasti matricos E diskretinio logaritmo.

Tarkime, kad klientas A atlieka šifruotojo, o klientas B – adresato vaidmenį. Vartotojas užkrauna adresato viešąjį raktą (aišku, kad šis raktas turi buti suformuotas prieš tai) įrašęs kliento B viešojo rakto failo pavadinimą CryptoClientB.key Partner PK ID laukelyje ir paspaudęs mygtuką Get Partner Public Key from the Server kliento A lange. Šifruojamo pranešimo failo pavadinimas yra įrašomas į laukelį, esantį mygtuko Send File kairėje. Šiam tyrimui pasirinksime frazę „The quick brown fox jumps over a lazy dog“, kurią išsaugosime faile ExampleA.txt. Gauto failo dydis yra 42 baitai, o tai reiškia, kad šis pranešimas gali būti užšifruotas šifravimo raktu K, kurio ilgis šiuo atveju yra 49 baitai. Duomenų užšifravimas ir persiuntimas adresatui vykdomas paspaudus mygtuką Send File. Vartotojas gali stebėti užšifravimo proceso laiko sąnaudas. Mūsų atveju gauti rezultatai atrodo taip:



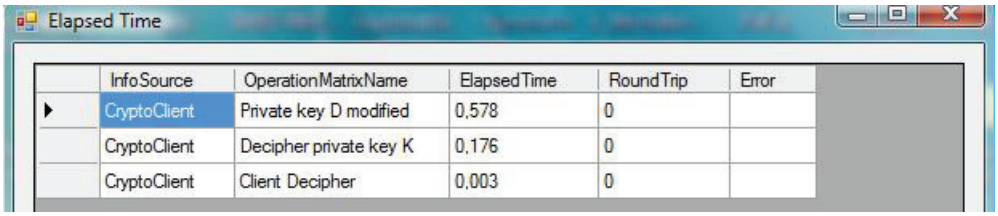
InfoSource	OperationMatrixName	ElapsedTime	RoundTrip	Error
CryptoClient	Client J matrix	0,265	0	
CryptoClient	Fast Inverse Client J	0,058	0	
CryptoClient	Private key Y	0,000	0	
CryptoClient	Private Key V modified	0,448	0	
CryptoClient	Private key S modified	0,424	0	
CryptoClient	Cipher private key K	0,128	0	
CryptoClient	Client Cipher	0,002	0	
CryptoClient	Client postcipher key B	0,071	0	
CryptoClient	Client postcipher key B2	0,059	0	
CryptoClient	Client postcipher key F	0,126	0	

7.8 pav. Laiko sąnaudų užšifruojant failą ExampleA.txt suvestinė

Suvestinės pirmos šešios eilutės parodo laiko sąnaudas generuojant užšifravimo raktą K. Bendras sunaudotas šiems veiksmams laikas yra 1,323 ms. Suvestinės septintoji eilutė yra paties užšifravimo proceso laiko sąnaudos. Jos sudaro 0,002 ms.

Paskutinės trys eilutės parodo dekriptoriaus $\varepsilon = \{B_1, B_2, F\}$ skaičiavimo laiko sąnaudas. Matome, kad laiko sąnaudos dekriptoriaus matricoms apskaičiuoti iš viso sudaro 0,256 ms. Gauta šifrograma ir papildomos matricos yra saugojamos faile ExampleAEncrypted.

Pranešimo adresatas užkrauna užšifruotą failą savo agente ir jį iššifruoja įrašydamas laukelyje, esančiame mygtuko Get File kairėje, pradinio failo pavadinimą (mūsų atveju ExampleA.txt) ir paspausdamas minėtą mygtuką. Vartotojas gali stebėti užšifravimo proceso laiko sąnaudas. Mūsų atveju gauti rezultatai atrodo taip:



7.9 pav. Laiko sąnaudų iššifruojant failą ExampleA.txt suvestinė

Iš gautos suvestinės matome, kad laiko sąnaudos iššifravimo raktui gauti yra 0,833 ms, o pats iššifravimo procesas užtruko 0,025 ms. Iššifruotas pranešimas yra saugojamas faile ExampleADecrypted.txt. Jo tekstas visiškai sutampa su pradiniu tekstu. Taip pat galime matyti, kad užšifravimo režimo metu reikia apskaičiuoti dvi matricas V ir $S = XVX^{-1}$ naudojant daugianarius, tačiau iššifravimo režimo metu tokia matrica yra tik viena. Ši matrica yra $D = Y^{-1}UY$. Šis faktas lemia tai, kad naudojant MLAŠ protokolą duomenų užšifravimas yra beveik dvigubai ilgesnis už iššifravimą.

Šiame skyrelyje pateikėme MLAŠ protokolo agentų veikimo paprastą pavyzdį. Kitame skyrelyje testuosime sukurtą programą, tirsime viešųjų raktų atsparumą DLA ir stebėsime pagrindinių procesų laiko sąnaudas prie skirtingų sistemos parametrų.

7.3. MLAŠ protokolo agentų tyrimas

Programa testuojama ant kompiuterio su šiais sisteminiais parametrais:

- Procesorius: Intel Core 2 Duo T6400 2.00 GHz;
- Atmintis (RAM): 4.00 GB;
- 32 bitų Windows operacinė sistema.

Šį skyrelį pradėsime nuo greitaveikos tyrimo, t.y. stebėsime laiko sąnaudas aukščiau aprašytuose etapuose kai keičiama parametro p reikšmė ir saugumo lygio reikšmė. Mes tirsime du saugumo lygius: $L = 80$ ir $L = 112$. Saugumo lygio reikšmė $L = 112$ buvo pasirinkta remiantis NIST standartais, kuriuose pabrėžiama, kad nuo 2014 metų pradžios dauguma kriptografinių primityvų turėtų naudoti 112 bitų saugumo galią. Šiam tyrimui pasirinksime frazę „The quick brown fox jumps over a

lazy dog“, kurią išsaugosime faile ExampleB.txt. Gauto failo dydis yra 42 baitai, o tai reiškia, kad užšifruojant ir iššifruojant šį failą raktas K bus panaudotas lygiai vieną kartą nepriklausomai nuo parametrų p ir m reikšmių. Stebėsime visų protokolo etapų laiko sąnaudas (jas žymėsime t). Kadangi laiko sąnaudos yra atsitiktinis dydis, tai siekiant sumažinti atsitiktinumą mes remsimės didžiųjų skaičių dėsniu, t.y. apskaičiuosime įvertį

$$\hat{t} = \sum_{i=1}^{N} t_i \tag{7.1}$$

kuris praranda atsitiktinumo pobūdį, kai N neaprėžtai didėja [58]. Mes pasirinksime $N = 20$, t.y. kiekvieno etapo veiksmus kartosime 20 kartų. Šio tyrimo rezultatai pateikti 7.1 ir 7.2 lentelėse.

Iš gautų duomenų matome, kad duomenų generavimas gali užtrukti nepriklausomai nuo parametro p reikšmės. Kai šio parametro reikšmė yra didelė, viešieji duomenys generuojami ilgai kadangi šiuo atveju beveik visas sunaudotas laikas yra skirtas multiplikacinės grupės $\boldsymbol{Z}_n^{\#}$ kėlimo laipsniais lentelei apskaičiuoti. Šie skaičiavimai sudaro daugiau, negu 99% sunaudoto laiko.

Ilgas viešųjų duomenų generavimas, kai parametro p reikšmė yra maža yra susijęs su neišsigimusios matricos T generavimo problemomis. Kadangi ši matrica priklauso laipsniniam žiedui virš skaitinio žiedo, tai didelės eilės matricoms dažnai tenka generuoti šią matricą kelis kartus. Šis faktorius stipriai įtakoja bendrąsias laiko sąnaudas. Taip pat, kai parametro p reikšmė yra maža, tai klientai sunaudoja daug laiko generuodami viešąjį, slaptąjį ir šifravimo raktus. Pagrindinė laiko sąnaudų dalis šiuo atveju yra skirta daugianariams skaičiuoti. Kadangi daugianario koeficientų vektoriaus dydis sutampa su matricų eile m, tai matricos U skaičiavimas užima apie 90% visų šio etapo laiko sąnaudų.

Taip pat matome, kad laiko sąnaudos operacijai XOR mažai priklauso nuo parametrų p, m ir L. Šios sąnaudos vidutiniškai sudaro apie 0,003 ms. Nuo minėtų parametrų priklauso bendrojo rakto skaičiavimo sąnaudos bei papildomų matricų skaičiavimo sąnaudos.

7.1 lentelė Laiko (ms) sąnaudų priklausomybė nuo sistemos parametrų, kai saugumo lygis $L = 80$.

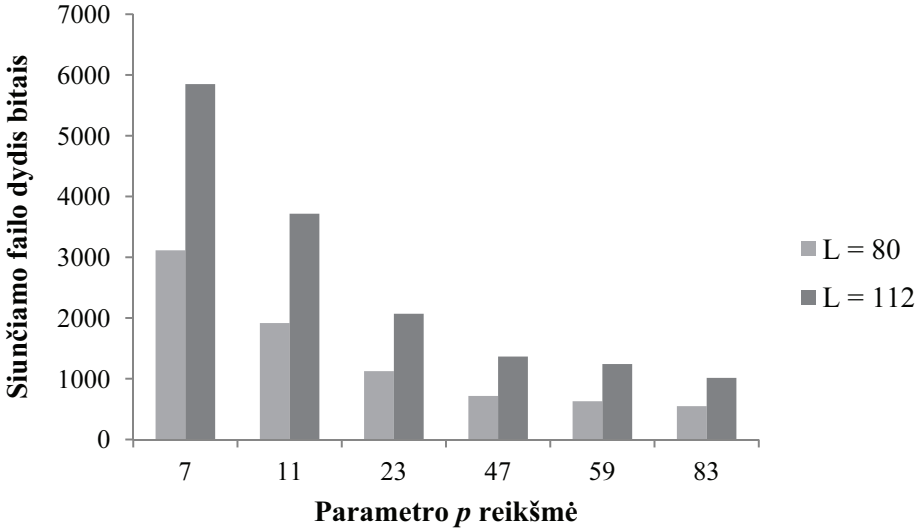
p	n	m	Viešieji duomenys	PrK ir PuK	Užšifravimas			Iššifravimas	
					K	XOR	PuK	K	XOR
7	21	32	21,489	211,869	290,110	0,002	16,948	159,042	0,002
11	33	25	93,570	81,353	122,597	0,002	9,355	76,936	0,003
23	69	19	49,994	31,340	52,238	0,003	5,312	28,292	0,003
47	141	15	105,069	14,045	22,458	0,003	2,781	12,536	0,003
59	177	14	182,839	9,442	18,295	0,003	2,526	9,554	0,003
83	249	13	304,661	8,448	14,916	0,003	2,121	7,797	0,003

7.2 lentelė Laiko (ms) sąnaudų priklausomybė nuo sistemos parametrų, kai saugumo lygis $L = 112$.

p	n	m	Viešieji duomenys	PrK ir PuK	Užšifravimas			Iššifravimas	
					K	XOR	ε	K	XOR
7	21	44	221,691	525,810	964,289	0,002	44,686	493,475	0,002
11	33	35	116,578	227,305	385,037	0,002	22,351	208,643	0,002
23	69	26	76,774	89,695	129,817	0,002	9,834	87,158	0,003
47	141	21	148,640	42,005	76,946	0,003	7,370	42,510	0,003
59	177	20	172,314	39,762	57,037	0,003	5,360	35,881	0,003
83	249	18	303,137	23,154	47,922	0,003	5,505	24,398	0,003

Kadangi serverio atliekami veiksmai yra vienkartiniai, o raktų generavimas atliekamas kiekvieną kartą, tai iš gautų rezultatų matome, kad vertinant laiko sąnaudas didesnės parametro p reikšmės turi pranašumą prieš mažesnes. Ypatingą svarbą čia turi papildomų matricų generavimas užšifravimo režimo metu, kadangi šie duomenys įtakoja siunčiamo failo dydį. Jau žinome, kad pradinio failo ExampleB.txt dydis yra 42 baitai. Toks yra ir iššifruoto failo ExampleBDecrypted.txt dydis. Siunčiamo failo

ExampleBEncrypted.txt dydžio priklausomybė nuo parametro *p* ir saugumo lygio pateikta grafiškai 7.10 paveiksle.



7.10 pav. Siunčiamo failo dydžio bitais priklausomybė nuo parametro *p* ir saugumo lygio *L*.

Matome, kad kai parametro *p* reikšmė yra maža, tai siunčiamas failas yra didelis. Šio fakto priežastis yra didelės eilės matricos. Kadangi visais atvėjais kiekvienas matricos elementas padidina siunčiamo failo dydį vienu baitu, tai beveik visą siunčiamą failą ExampleBEncrypted.txt sudaro būtent papildomos matricos.

Gautus rezultatus palyginsime su RSA asimetrinio šifravimo protokolo rezultatais. Laiko sąnaudos atskiriems protokolo etapams buvo gautos naudojant internetinę RSA testavimo sistemą [59]. RSA protokolo tyrimai buvo atlikti su plačiausiai praktikoje naudojamais 1024 ir 2048 bitų raktais bei su 3072 ir 4096 bitų raktais. Kiekvienu atveju šifruojamas didžiausios galimos apimties tekstas. Kiekvienas protokolo etapas buvo vykdomas 20 kartų.

7.3 lentelė RSA protokolo laiko (ms) sąnaudų priklausomybė nuo raktų ilgio.

Etapas	RSA-1024	RSA-2048	RSA-3072	RSA-4096
Raktų generavimas	465,250	3615,750	25506,600	54424,550
Užšifravimas	2,550	4,950	8,800	10,400
Iššifravimas	18,600	103,650	316,350	497,350

Reikia pastebėti, kad tiriamoji RSA šifravimo sistema [59] naudoja optimizaciją, kuri yra paremta kinų liekanų teorema. Tai leidžia sumažinti užšifravimo ir iššifravimo laiko sąnaudas, tačiau padidina sąnaudas raktams generuoti. Taip pat reikia paminėti, kad RSA protokolo specifika yra tokia, kad užšifravimo greitis yra žymiai mažesnis už iššifravimo greitį, jeigu yra pasirenkamas specialus eksponentės e pavidalas, t.y. $e = 2^k + 1$, kadangi šiuo atvėju tik pirmas ir paskutinis šio elemento bitai yra 1, o kiti bitai yra 0. Tai leidžia atlikti mažiau elementariųjų operacijų. Praktikoje dažnai naudojamos reikšmės $e = 3$ arba $e = 65537$ [59].

Kadangi MLAŠ protokolo greitaveika yra geriausia, kai $p = 83$, tai siekiant palyginti dvi asimetrinio šifravimo sistemas mes turime pasirinkti tokio dydžio matricas, kurios leistų mums šifruoti panašaus ilgio pranešimus naudojant šifravimo raktą tik vieną kartą. Dėl šios priežasties lyginant MLAŠ su RSA mes naudosime tokias matricas:

- RSA-1024 atitinka matricų eilė $m = 12$. Šifravimo rakto ilgis – 1152 bitai.
- RSA-2048 – $m = 16$. Šifravimo rakto ilgis – 2048 bitai.
- RSA-3072 – $m = 20$. Šifravimo rakto ilgis – 3200 bitų.
- RSA-4096 – $m = 23$. Šifravimo rakto ilgis – 4232 bitai.

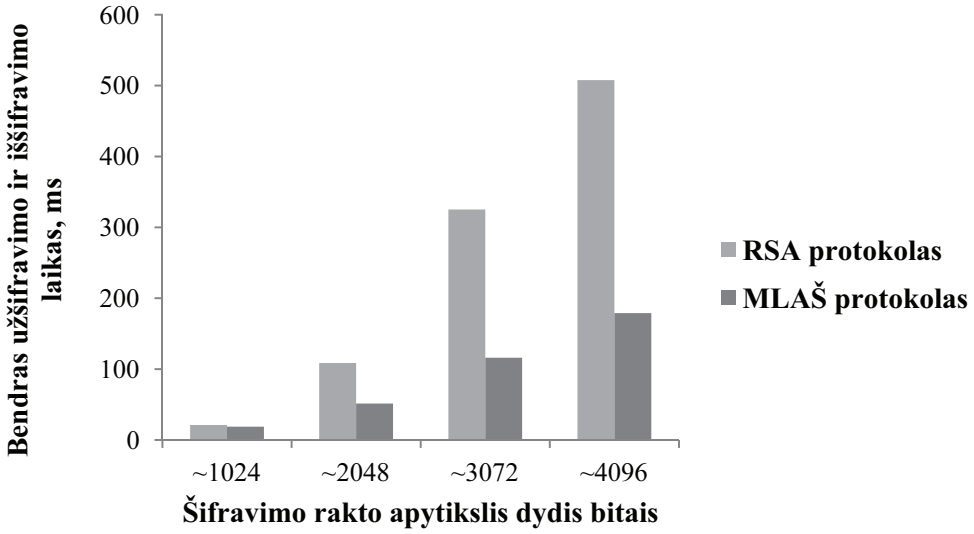
Naudojant RSA protokolo rakto ilgį atitinkančią matricų eilę m šifruosime tą patį pranešimą, kaip ir RSA atveju. Gauti rezultatai yra pateikti 7.4 lentelėje:

7.4 lentelė MLAŠ protokolo laiko sąnaudų priklausomybė nuo matricų dydžio.

Etapas	MLAŠ, $m = 12$	MLAŠ, $m = 16$	MLAŠ, $m = 20$	MLAŠ, $m = 23$
Raktų generavimas	8,634	18,663	34,438	52,392
Užšifravimas	12,911	35,597	77,614	114,814
Iššifravimas	5,885	15,832	38,401	64,206

Iš gautų rezultatų matome, kad, lyginant su RSA protokolu, MLAŠ protokolas naudoja daugiau laiko užšifruojant pranešimą, tačiau žymiai mažiau laiko iššifruojant šifrogramą. Šio fakto priežastis yra ta, kad papildomų dekriptoriaus matricų apskaičiavimas užšifravimo režime užtrunka beveik dvigubai daugiau negu raktų generavimo etapas, kadangi naudojant daugianarius reikia apskaičiuoti ne tik matricą V, bet ir matricą XVX^{-1}. Iššifruojant šifrogramą reikia apskaičiuoti tik iššifravimo raktą.

Grafiškai pateiksime bendro užšifravimo ir iššifravimo laiko sąnaudų palyginimo rezultatus:



7.11 pav. Bendrų užšifravimo ir iššifravimo laiko sanaudų palyginimas

Iš pateikto grafiko matome, kad bendras užšifravimo ir iššifravimo laikas yra mažesnis MLAŠ protokolo atveju. Šis skirtumas ryškėja ilgesniems raktams ir siekia 2,8 karto, kai lyginame RSA-4096 ir MLAŠ protokolą, kai $p = 83$ ir $m = 23$.

MLAŠ protokolas leidžia naudoti ir didesnes pagrindinių parametrų reikšmes, negu nagrinėtos aukščiau. Tačiau, kaip jau buvo minėta skyrelyje 5.3 šie parametrai turi būti pasirikti taip, kad MQ lygčių sistemą sudarytų virš 80 lygčių. Kai saugumo lygis $L = 80$, tai didžiausia parametro p reikšmė, su kuria ši sąlyga yra tenkinama yra 1019. Ši reikšmė yra viršutinė parametro p riba, kai $L = 80$. Siekiant, kad MQ lygčių sistemą sudarytų virš $L = 112$ lygčių reikia pasirinkti $m = 11$. Tokiu atveju viršutinė parametro p riba yra 2207. Nors šios reikšmės leistų maksimaliai sumažinti MLAŠ protokolo atskirų etapų laiko sąnaudas, tačiau naudoti tokias reikšmes praktikoje būtų neracionalu, kadangi dėl didelių aritmetinių veiksmų lentelių atminties reikalavimai yra per dideli. Taip pat reikia pabrėžti, kad, nors tokiu atveju multiplikacinės pusgrupės elementams vaizduoti jau neužtektų vieno baito, kas neigiamai įtakotų operacijos XOR greitaveiką, tačiau, dėl ypač mažų laiko sąnaudų šiam procesui atlikti, ši neigiama įtaka būtų beveik nepastebima.

Kadangi MLAŠ protokolo metu yra sudaromas bendras šifravimo raktas K, tai šis protokolą galima palyginti ir su žinomais raktų apsikeitimo protokolais. Šiam tikslui mes panaudosime italų mokslininkų straipsniu [24], kuriame tarpusavyje lyginami elipsinių kreivių, Difio-Helmano ir STR raktų apsikeitimo protokolai. Savo straipsnyje italų mokslininkai nagrinėja šių protokolų realizaciją mobiliajame telefone Nokia N70, kuris turi 220 MHz procesorių, 55 MB RAM atminties. Straipsnio

autoriai STR protokolui realizuoti panaudojo 31 bito skaičius. Matricų komutatyvumas yra užtikrinamas naudojant poaibį generuojančią matricą *S*.

Kadangi MLAŠ protokolas taip pat naudoja matricas, tai šį protokolą galima palyginti su STR protokolu skaičiavimo sąnaudų prasme. Šiam palyginimui pasirinksime MLAŠ protokolo parametrą $p = 83$ ir pasirinksime keisime parametro m reikšmę taip, kad gauto šifravimo rakto K ilgis atitiktų STR protokolo bendrojo rakto ilgį. Kadangi skaičiavimo sąnaudos raktui K apskaičiuoti yra skirtingos užšifravimo ir iššifravimo režimuose, tai įvertinsime abu šiuos atvejus. Vertinant STR protokolo elementariųjų operacijų kiekį mes orientuojamės į atliekamų operacijų vidurkį. Tyrimo rezultatai pateikti 7.5 lentelėje.

7.5 lentelė MLAŠ ir STR protokolų skaičiavimo sąnaudų bendrajam raktui K apskaičiuoti palyginimas.

STR			MLAŠ			
					Skaičiavimų sąnaudos	
m	Rakto K ilgis bitais	Skaičiavimų sąnaudos	m	Rakto K ilgis bitais	Užšifravimo režimas	Iššifravimo režimas
3	279	6570	6	288	9504	5148
4	496	16352	8	512	30720	16320
5	775	32850	10	800	76000	39900
6	1116	57816	12	1152	158976	82800
7	1519	93002	14	1568	296352	153468
8	1984	140160	16	2048	507904	261888
9	2511	201042	18	2952	816480	419580
10	3100	277400	20	3200	1248000	639600
11	3751	370986	22	3872	1831456	936540
12	4464	483552	24	4608	2598912	1326528

Matome, kad MLAŠ protokolas reikalauja daugiau elementariųjų operacijų, negu STR protokolas. Naudojant gautus rezultatus mes galime palyginti MLAŠ protokolo realizacijos Nokia N70 telefone laiko sąnaudas su kitais protokolais. MLAŠ protokolo laiko sąnaudas šifravimo raktui gauti įvertinsime padauginę STR protokolo laiko sąnaudas bendrajam raktui gauti iš santykio tarp elementariųjų operacijų, kurios reikalingos panašaus ilgio raktams gauti, kiekių. Gauti rezultatai parodyti 7.6 lentelėje.

7.6 lentelė MLAŠ ir STR protokolų laiko sąnaudų bendrajam raktui *K* apskaičiuoti palyginimas.

STR			MLAŠ			
m	Rakto *K* ilgis bitais	Laiko sąnaudos, ms	*m*	Rakto *K* ilgis bitais	Laiko sąnaudos, ms	
					Užšifravimo režimas	Iššifravimo režimas
3	279	3,13	6	288	4,529	2,454
4	496	5,94	8	512	11,161	5,928
5	775	10,31	10	800	23,857	12,527
6	1116	15,47	12	1152	42,543	22,153
7	1519	22,96	14	1568	73,174	37,884
8	1984	31,41	16	2048	113,830	58,674
9	2511	44,85	18	2952	182,136	93,602
10	3100	56,87	20	3200	255,858	131,142
11	3751	71,26	22	3872	351,811	179,860
12	4464	89,53	24	4608	481,224	245,581

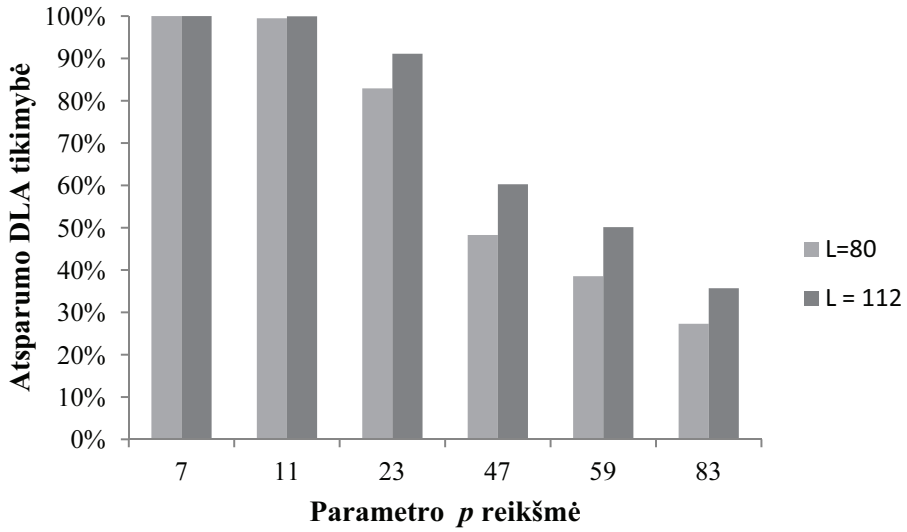
Iš gautų rezultatų matome, kad MLAŠ protokolo laiko sąnaudos yra didesnės, negu STR protokolo. Šiuos rezultatus galime palyginti su klasikiniais elipsinių kreivių ir Difio-Helmano raktų apsikeitimo protokolų laiko sąnaudomis. Palyginę mūsų gautus rezultatus su straipsnyje [24] pateiktais duomenimis matome, kad lyginant DH-

1024 protokolą su panašaus šifravimo rakto ilgio MLAŠ protokolu gauname pagreitėjimą iki 12,7 kartų užšifravimo ir 24,4 kartų iššifravimo atveju, o lyginant ECDH-571 protokolą su panašaus šifravimo rakto ilgio MLAŠ protokolu pagreitėjimas siekia 120 kartų užšifravimo ir 228 kartus iššifravimo atveju. Tai reiškia, kad mūsų pasiūlytas protokolas žymiai greitesnis už abu nagrinėjamus klasikinius protokolus, t.y. laiko sąnaudos panašaus ilgio raktui apskaičiuoti yra mažesnės.

Siekiant nustatyti geriausias šio parametro reikšmes mes nagrinėsime MLAŠ protokolo atsparumą DLA. Skyrelyje 7.2 buvo nustatyta, kad ML eksponentė E yra atspari minėtai atakai jeigu matrica X arba matrica U turi nulinių elementų. Pažymėję vienintelio matricos Q idealo elemento vietą (i_0, j_0), čia i_0, j_0 yra atitinkamai eilutės ir stulpelio indeksai, turime, kad matrica E yra atspari minėtai atakai, jeigu bent vienas matricos X i_0-nio stulpelio arba matricos U j_0-nės eilutės elementas yra lygus 0. Aišku, kad matricos E atsparumas priklauso nuo skaitinio žiedo parametro r ir matricų eilės m. Naudojant priešingo įvykio tikimybės skaičiavimo formulę turime, kad tikimybė, jog matrica E yra atspari DLA yra lygi:

$$prob(r,m) = 1 - \left(\frac{r-1}{r}\right)^{2m} \tag{7.2}$$

Kadangi skaitinio žiedo parametras $r = p - 1$, o matricų eilė priklauso nuo parametrų p ir L, tai 7.12 paveiksle pateiksime matricos E atsparumo DLA priklausomybę nuo šių parametrų. Tikimybę matuosime procentais.



7.12 pav. Matricos E atsparumo DLA tikimybės priklausomybė nuo p ir L.

Matome, kad kai parametro p reikšmė yra maža, tai matrica E beveik visada yra atspari DLA, kadangi matricos yra didelės, o skaitinio žiedo parametras r yra mažas. Taigi šiuo aspektu mažesnės parametro p reikšmės turi privalumą prieš didesnes.

Iš gautų šio skyrelio rezultatų matome, kad mažesnėms parametro p reikšmėms užšifravimo ir iššifravimo etapai trunka ilgiau, tačiau viešojo rakto matrica E yra atsparesnė DLA. Didesnėms parametro p reikšmėms minėti etapai trunka mažiau, bet serverio veiksmai yra atliekami lėčiau. Be to atminties reikalavimai mažesnėms p reikšmėms yra mažesni.

7.4. Išvados ir rezultatai

- Aprašyti pagrindiniai algoritmai, kurie naudojami realizuojant MLAŠ protokolą.
- Naudojant Microsoft .NET Framework platformą MLAŠ protokolas buvo realizuotas C# programavimo kalba. Sukurtas serverio agentas, kuriame yra generuojami protokolo viešieji duomenys ir du klientų agentai, kurie naudojant šiuos duomenis bendrauja tarpusavyje.
- Eksperimentiniu būdu nustatyta, kad MLAŠ yra atsparus DLA, jeigu bent vienas iš tam tikrų laipsninių matricų elementų yra lygus 0. Įvertintos atsparumo šiai atakai tikimybės kai keičiamos pagrindinių saugumo parametrų reikšmės.
- Naudojant sukurtus agentus ištirtos visų MLAŠ protokolo etapų laiko sąnaudos. Gauti rezultatai palyginti su RSA protokolo laiko sąnaudų eksperimentiniais rezultatais. Atlikto tyrimo rezultatai parodė, kad RSA užšifravimas yra greitesnis už MLAŠ užšifravimą, tačiau iššifravimas atliekamas greičiau, kai naudojamas MLAŠ protokolas. Bendros užšifravimo ir iššifravimo laiko sąnaudos yra mažesnės MLAŠ protokolo atveju. Šis skirtumas riškėja didinant šifravimo rakto ilgį.
- Remiantis [24] straipsniu MLAŠ protokolo šifravimo rakto generavimo laiko sąnaudos palygintos su STR, Difio-Helmano ir elipsinių kreivių raktų apsikeitimo protokolų laiko sąnaudomis. Gauti rezultatai parodė kad lyginant DH-1024 protokolą su panašaus šifravimo rakto ilgio MLAŠ protokolu gauname pagreitėjimą iki 12,7 kartų užšifravimo ir 24,4 kartų iššifravimo atveju, o lyginant ECDH-571 protokolą su panašaus šifravimo rakto ilgio MLAŠ protokolu pagreitėjimas siekia 120 kartų užšifravimo ir 228 kartus iššifravimo atveju. Tačiau lyginant MLAŠ ir STR protokolus matome, kad pastarasis protokolas yra greitesnis.

8. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

1. Ištirtos MLF algebrinės savybės ir pasiūlyta nauja platforminė algebrinė sistema paremta Sylovo teoremos pagrindu. Įrodyta, kad naudojant šią algebrinę sistemą, MLF turinti papildomus jungtinumo apribojimus yra saugi statistinės kriptoanalizės atžvilgiu.
2. Nagrinėjant MLF paremtą siųloma algebrine sistema su papildomais jungtinumo apribojimais nustatyta, kad šios funkcijos kriptografinis saugumas remiasi laipsninių lygčių sistemos sprendimo sudėtingumu, kuris yra panašus į MQ lygčių sistemos sudėtingumą. Kadangi MQ lygčių sistemos sprendimo uždavinys priklauso NP-pilnųjų uždavinių klasei, tai galime daryti prielaidą, jog mūsų darbe nagrinėjamos laipsninių lygčių sistemos sudėtingumas tenkina kriptografijai keliamus sudėtingumo reikalavimus, ir tuo pačiu MLF yra atspari algebrinei atakai.
3. Naudojant MLF su papildomais jungtinumo apribojimais sukurtas originalus asimetrinio šifravimo protokolas, kurio kriptografinis saugumas remiasi MLF apgręžiamumo uždavinio sudėtingumu. Remiantis atlikta analize, šis protokolas yra atsparus statistinei ir algebrinei kriptoanalizei.
4. Nagrinėjant MLAŠ protokolo atsparumą algebrinei kriptoanalizei pasiūlyta panaudoti diskretinio logaritmo funkciją matricų pusgrupėje. Remiantis šią funkciją pasiūlyta ataka prieš pirmąją protokolo versiją, kuri leidžia palengvinti MLF uždavinio analizę keičiant jį atitinkamu MMQ uždaviniu. Siekiant išvengti diskretinio logaritmo funkcijos panaudojimo kriptoanalizei, sukurta patobulinta MLAŠ protokolo versija.
5. Eksperimentiniu būdu nustatyta, kad MLAŠ yra atsparus DLA, jeigu bent vienas iš tam tikrų laipsninių matricų elementų yra lygus 0. Įvertintos atsparumo šiai atakai tikimybės kai keičiamos pagrindinių saugumo parametrų reikšmės.
6. Nustatyti pagrindiniai MLAŠ protokolo saugumo parametrai: parametras p, kuris nusako multiplikacinės pusgrupės eilę, naudojamų kvadratinių matricų eilė m ir saugumo lygis L. Taip pat nustatyta parametro m priklausomybė nuo kitų pagrindinių saugumo parametrų. Šis parametras tiesiogiai proporcingas saugumo lygiui L ir atvirkščiai proporcingas parametrui p.
7. Kadangi pagrindinė ataka prieš MLAŠ protokolą yra pilnas laipsninių matricų perrinkimas, pasiūlytas protokolo saugumo lygio L susiejimas su $(m - 1)$-os eilės daugianarių virš skaitinio žiedo $\boldsymbol{Z}_r$ aibės galia.
8. Atliktas MLAŠ protokolo realizacijos 32 bitų mikroprocesoriuje teorinis palyginimas su klasikinių protokolų (El-Gamal-2048, ECC-521) realizacija elementariųjų operacijų atžvilgiu. Tyrimų rezultatai parodė, kad vykdant MLAŠ protokolą atliekama mažiau elementariųjų operacijų, negu vykdant klasikinius protokolus. Naudojant pirmąją protokolo versiją skaičiavimo sanaudos yra vidutiniškai 235 kartų mažesnės lyginant su El-Gamal-2048 protokolu ir 69 kartus mažesnės lyginant su ECC-521 protokolu.

9. Atliktas eksperimentinis tyrimas parodė, jog bendros užšifravimo ir iššifravimo laiko sąnaudos yra mažesnės MLAŠ protokolo atveju lyginant šį protokolą su RSA asimetriniu šifravimu. Šis skirtumas dar labiau didėja didinant šifravimo rakto ilgį. Tai leidžia lanksčiai pritaikyti šifravimo rakto ilgį, o tuo pačiu ir šifruojamos informacijos kiekį vartotojo poreikiams.

LITERATŪRA

1. Katz, J., & Lindell, Y. (2008). *Introduction to modern cryptography.* Chapman & Hall/CRC.
2. Schneier, B. (2007). *Applied cryptography: protocols, algorithms, and source code in C*. john wiley & sons.
3. Lloyd, S. (2013). Quantum enigma machines. *arXiv preprint arXiv:1307.0380*.
4. Kruh, L., & Deavours, C. (2002). The commercial enigma: beginnings of machine cryptography. *Cryptologia*, *26*(1), pp. 1-16.
5. Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, 22*(6), pp. 644-654.
6. Виноградов, И. М. (1972). *Основы теории чисел*. Москва: Наука
7. Carmichael, R. D. (1912) On Composite Numbers P Which Satisfy the Fermat Congruence. *The American Mathematical Monthly 19*(2), pp. 363–385
8. Kammüller, F., & Paulson, L. C. (1999). A Formal Proof of Sylow's Theorem. *Journal of Automated Reasoning*, *23*(3), pp. 235-264.
9. Garey, M., & Johnson, D. (1979). *Computers and Intractability. A Guide to the Theory of NP-Completeness.* San Francisco: Freemann.
10. Goldreich, O. (2003). *Foundations of cryptography, vol. 1.* Cambridge University Press.
11. Hastad, J., Impagliazzo, R., Levin, L., & Luby, M. (1999). A pseudorandom generator from any one-way function. *Siam Journal on Computation, 28*(4), pp. 1364–1396.
12. Yao, A.: Theory and Applications of Trapdoor functions, Proceedings of the 23rd FOCS, IEEE, pp. 80-91 (1982)
13. Sakalauskas, E. (2008). Kriptografinės sistemos: mokomoji knyga. Kauno technologijos universitetas. *Vitae Litera*.
14. Stanek, M. (2011). Extending Baby-step Giant-step algorithm for FACTOR problem. *IACR Cryptology ePrint Archive*, *2011*, 59.
15. Rivest, R., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM, 21*(2), pp. 120-126.
16. ElGamal, T. (1985). A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory, 31*(4), pp. 469–472.
17. Saravanan, N., Mahendiran, A., Subramanian, N. V., & Sairam, N. (2012). An Implementation of RSA Algorithm in Google Cloud using Cloud SQL. *Research Journal of Applied Sciences, Engineering and Technology*, *4*(19), pp. 3574-3579.
18. Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, *26*(5), pp. 1484-1509.
19. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, *48*(177), pp. 203-209.
20. NIST. (2013). *Digital Signature Standard (DSS). FIPS 186-4.* National Institute of Standards and Technology.
21. Федеральное агенство по техническому регулированию и метрологии. (2013) *Инфромационная технология. Криптографическая защита информации. Процессы формирования и проверки цифровой подписи.* Москва: Стандартинформ.
22. Monico, C. (2002). *Semirings and Semigroup actions in Public–Key Cryptography. PhD. thesis.* University of Notre Dame.

23. Jacobs, K. (2011). A survey of modern mathematical cryptology.
24. Ottaviani, V., Zanoni, A., & Regoli, M. (2010). Conjugation as public key agreement protocol in mobile cryptography. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on* pp. 1-6. IEEE.
25. Sracic, M. (2011) Quantum Circuits for Matrix Multiplication.
26. Wagner, N. R., & Magyarik, M. R. (1985). A public-key cryptosystem based on the word problem. In *Advances in Cryptology* . pp. 19-36. Springer Berlin Heidelberg.
27. Myasnikov, A., Shpilrain, V., & Ushakov, A. (2008). *Group-based cryptography*. Springer.
28. Ko, K. H., Lee, S. J., Cheon, J. H., Han, J. W., Kang, J. S., & Park, C. (2000). New public-key cryptosystem using braid groups. In *Advances in Cryptology—CRYPTO 2000*, pp. 166-183. Springer Berlin Heidelberg.
29. Anshel, I., Anshel, M., & Goldfeld, D. (1999). An algebraic method for public-key cryptography. *Mathematical Research Letters*, *6*, pp. 287-292.
30. Shpilrain, V., & Ushakov, A. (2006). The conjugacy search problem in public key cryptography: unnecessary and insufficient. *Applicable Algebra in Engineering, Communication and Computing*, *17*(3-4), pp. 285-289.
31. Hughes, J. (2002). A linear algebraic attack on the AAFG1 braid group cryptosystem. In *Information Security and Privacy* (pp. 176-189). Springer Berlin Heidelberg.
32. Dehornoy, P. (2004). Braid-based cryptography. *Contemp. Math*, *360*, pp. 5-33.
33. Myasnikov, A. D., & Ushakov, A. (2007). Length based attack and braid groups: cryptanalysis of Anshel-Anshel-Goldfeld key exchange protocol. In *Public Key Cryptography–PKC 2007*, pp. 76-88. Springer Berlin Heidelberg.
34. Hughes, J., & Tannenbaum, A. (2003). Length-based attacks for certain group based encryption rewriting systems. *arXiv preprint cs/0306032*.
35. Maze, G., Monico, C., & Rosenthal, J. (2005). Public key cryptography based on semigroup actions. *arXiv preprint cs/0501017*.
36. Sakalauskas, E., Tvarijonas, P., & Raulinaitis, A. (2007). Key Agreement Protocol (KAP) Using Conjugacy and Discrete Logarithm Problems in Group Representation Level. *Informatica, 18*(1), pp. 115-124.
37. Eftekhari, M. (2012). A Diffie–Hellman key exchange protocol using matrices over noncommutative rings. *Groups-Complexity-Cryptology*, *4*(1), pp. 167-176.
38. Sakalauskas, E., & Luksys, K. (2007). Matrix Power S-Box Construction. *IACR Cryptology ePrint Archive*, *2007*, 214.
39. Sakalauskas, E., Listopadskis, N., & Tvarijonas, P. (2008). Key Agreement Protocol (KAP) Based on Matrix Power Function. *"Information Science And Computing", book 4 „Advanced Studies in Software and Knowledge Engineering"*, pp. 92–96.
40. Lukšys, K. (2013). *Matricinio laipsnio šifras ir jo analizė. Daktaro disertacija.* Kaunas: KTU.
41. Luksys, K., Sakalauskas, E., & Venckauskas, A. (2012). Implementation Analysis of Matrix Power Cipher in Embedded Systems. *Electronics and Electrical Engineering*, 2(118), pp. 95-98.
42. Vitkus, P., Sakalauskas, E., Listopadskis, N., & Vitkiene, R. (2012). Microprocessor realization of key agreement protocol (KAP) based on Matrix power function. *Electronics and electrical engineering* 1(117), pp. 33-36.
43. Luksys, K.. & Nefas, P. (2008). Matric Power S-Box Analysis. *"Information Science And Computing", book 4 „Advanced Studies in Software and Knowledge Engineering"*, pp. 97–102.
44. Luksys, K. & Sakalauskas, E. (2012) *Matrix power cipher. Information technology and control.* Kaunas: Technologija.

45. Michalkovič, A. (2010). *Netiesinės algebrinės lygčių sistemos sprendinių skaičiaus analizė. Magistro darbas.* Kaunas: KTU.
46. Hall, M. (1976). *The theory of groups* (Vol. 288). American Mathematical Soc.
47. Sakalauskas, E., Mihalkovich, A. (2012) Candidate One-Way Function Based on Matrix Power Function with Conjugation Constraints, In: *Bulgarian cryptography days 2012*. Conference proceedings, pp. 29-37
48. Гантмахер, Ф. Р. (1966). *Теория матриц*. Москва: Наука.
49. Mihalkovich, A., Sakalauskas, E. (2012) Asymmetric cipher based on MPF and its security parameters evaluation. In: Proc. of the Lithuanian Mathematical Society, Ser. A., *Lietuvos Matematikos Rinkinys*, Vol. 53, pp. 72-77.
50. Sakalauskas, E., Mihalkovich, A. (2014). New Asymmetric Cipher of Non-commuting Cryptography Class Based on Matrix Power Function. *Informatica*. 25(2) pp. 283-298.
51. Patarin, J., & Goubin, L. (1997). Trapdoor One-Way Permutations and Multivariate Polynomials. *Proceedings of ICICS'97, LNCS 1334,* pp. 356–368. Springer.
52. Buchberger, B. (1985). Gröbner-bases: an algorithmic method in polynomial ideal theory. *Recent Trends in Multidimensional Systems Theory*, pp. 184-232. Reidel Publishing Company.
53. Faugere, J. (2003). *Algebraic cryptanalysis of HFE using Gröbner bases.* Tech. rep., Nuskaityta iš: ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4738.pdf .
54. Thomae, E., & Wolf, C. (2012). Solving underdetermined systems of multivariate quadratic equations revisited. In *Public Key Cryptography–PKC 2012*, pp. 156-171. Springer Berlin Heidelberg.
55. Hashimoto, Y. (2009). *Algorithms to solve massively under-defined systems of multivariate quadratic equations.* Cryptology ePrint Archive: Report, no. 154 (2009). Nuskaityta iš http://eprint.iacr.org/2009/154.pdf
56. Mihalkovich, A., Sakalauskas, E., & Venckauskas, A. (2013). New Asymmetric Cipher Based On Matrix Power Function and Its Implementation in Microprocessors Efficiency Investigation. *Electronics & Electrical Engineering*, *19*(10), pp 119-122.
57. Gura, N., Patel, A., Wander, A., Eberle, H., & Shantz, S. C. (2004). Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. *Cryptographic Hardware and Embedded Systems, LNCS 3156*, pp. 119-132. Springer.
58. Aksomaitis, A. (2000). Tikimybių teorija ir statistika. *Kaunas: Technologija*.
59. Crypt-online (2014). RSA šifravimas: http://www.crypt-online.narod.ru/crypts/rsa/

MOKSLINIŲ PUBLIKACIJŲ DISERTACIJOS TEMA SĄRAŠAS

STRAIPSNIAI

Tarptautinėse duomenų bazėse esančiuose mokslo leidiniuose paskelbti straipsniai

Mokslinės informacijos instituto duomenų bazės „ISI Web of Science“ leidiniuose, turinčiuose citavimo indeksą

1. Michalkovič, Aleksejus; Sakalauskas, Eligijus; Venčkauskas, Algimantas. New Asymmetric Cipher Based On Matrix Power Function and Its Implementation in Microprocessors Efficiency Investigation. // Elektronika ir elektrotechnika. ISSN 1392-1215. 2013, No. 19(10), p. 119–122 [Science Citation Index Expanded (Web of Science); INSPEC; Computers & Applied Sciences Complete; Central & Eastern European Academic Source].
2. Sakalauskas, Eligijus; Michalkovič, Aleksejus. New Asymmetric Cipher of Non-commuting Cryptography Class Based on Matrix Power Function.// Informatica 25(2) pp. 283-298 / Vilniaus universitetas. Vilnius : Matematikos ir Informatikos institutas. ISSN 0868-4952. [Science Citation Index Expanded (Web of Science); INSPEC].

Kituose recenzuojamuose mokslo leidiniuose paskelbti straipsniai

Konferencijų pranešimų medžiagoje paskelbti straipsniai

1. Sakalauskas, Eligijus; Michalkovič, Aleksejus. Candidate one-way function based on matrix power function with conjugation constraints // BulCrypt 2012 : Bulgarian Cryptography Days 2012 : 1st International Conference on Bulgarian and Balkans Cryptography, 20-21, September, 2012, Sofia, Bulgaria : proceedings. Sofia : Bulgarian Science Fund, 2012. ISBN 9789542946229. p. 29-35. [0,500]

Periodiniuose leidiniuose ir vienkartiniuose straipsnių rinkiniuose ir kt. paskelbti straipsniai

1. Michalkovič, Aleksejus; Sakalauskas, Eligijus. Asymmetric cipher based on MPF and its security parameters evaluation // Lietuvos matematikos rinkinys : Lietuvos matematikų draugijos darbai. Serija B / Lietuvos matematikų draugija, Vilniaus universitetas. Vilnius : Vilniaus universitetas. ISSN 0132-2818. 2012, t. 53, p. 72-77. [0,500]

PRIEDAS

Šiame priede aprašysime keturis MLAŠ protokolo vykdymo etapus:

- Viešųjų parametrų generavimas;
- Kliento slaptojo ir viešojo raktų generavimas;
- Duomenų užšifravimas;
- Duomenų iššifravimas.

Viešųjų MLAŠ protokolo duomenų generavimas

Iš skyrelyje 5.5 pateikto aprašymo ir skyrelyje 6.2 pateiktų rezultatų matome, kad prieš vykdant MLAŠ protokolą serverio agente reikia suformuoti šiuos viešuosius duomenis:

- Multiplikacinė pusgrupė $\boldsymbol{Z_n^{\#}}$;
- Skaitinis žiedas $\boldsymbol{Z_r}$;
- Daugybos ir kėlimo laipsniu lentelės, kurios yra naudojamos atliekant skaičiavimus multiplikacinėje pusgrupėje;
- Sudėties ir daugybos lentelės, kurios yra naudojamos atliekant skaičiavimus skaitiniame žiede;
- Viešai paskelbta matrica Q, kuri priklauso platforminei pusgrupei;
- Viešai paskelbtos matricos Z_1 ir Z_2, kurios priklauso laipsniniam žiedui.

Kaip jau buvo minėta viešieji duomenys yra formuojami vieną kartą ir visi sistemos klientai naudoja juos vykdant protokolą, t.y šie duomenys yra persiunčiami visiems sistemos klientams. Formuodamas duomenis serveris atlieka veiksmus šešiais žingsniais, kurie yra aprašyti žemiau.

Pirmasis žingsnis. Serveris pasirenka multiplikacinę pusgrupę $\boldsymbol{Z_n^{\#}}$. Nuo šio pasirinkimo priklauso matricų eilė ir skaitinio žiedo $\boldsymbol{Z_r}$ pasirinkimas. Serveris pasirenka šiuos duomenis vertinant bendruosius atminties reikalavimus, raktų ilgius elementariųjų operacijų kiekį ir saugumo lygį. Šiame žingsnyje serverio verksmai yra tokie:

1 algoritmas. Atsitiktinio nelyginio pirminio skaičiaus *p* intervale [*min*; *max*] generavimas.

(Įvestis: generavimo bazė (angl. seed), intervalo rėžiai *min* ir *max*.)

1. Iš pirminių skaičių sąrašo pagal indeksą pasirenkamas skaičius p.
2. Jei skaičius p nėra iš intervalo [*min*; *max*] – generuoti naują indeksą ir grįžti prie 1. žingsnio. Priešingu atveju pereiti prie 3. žingsnio.
3. Jeigu skaičius $\frac{p-1}{2}$ yra pirminis – išeiti iš algoritmo ir grąžinti p. Priešingu atveju – generuoti naują indeksą ir grįžti prie 1. žingsnio.
4. Grąžinti p.

(Išvestis: parametras p.)

Mes rekomenduojame naudoti $max < 100$, kadangi kai $p = 83$, tai $n = 249$ yra didžiausias skaičius, tenkinantis MLAŠ protokolo reikalavimus, kurį galima pavaizduoti vienu baitu.

2 algoritmas. Multiplikacinės pusgrupės $Z_n^{\#}$ generavimas

(Įvestis: multiplikacinės pusgrupės parametras $n = 3p$.)

1. Sukuriamas darbinis kintamasis *ind*. Jam priskiriama reikšmė 0.
2. Su kiekvienu i nuo 0 iki $n - 1$ vykdyti:
   2.1. Jeigu $i = 0$ arba $i = p$ arba $i = 2p$ – praleisti skaičių. Priešingu atveju vykdyti:
      2.1.1. *MultSemigroup*[*ind*] = i;
      2.1.2. *ind* = *ind* + 1;
3. Grąžinti *MultSemigroup*.

(Išvestis: multiplikacinė pusgrupė $Z_n^{\#}$.)

3 algoritmas. Matricų eilės *m* skaičiavimas

(Įvestis: saugos lygis L, pagrindinis sistemos paramertas p)

1. $$m = \left\lceil \frac{(L+1)\ln 2 + \ln(p-1) - \ln(p-3)}{\ln(p-1)} \right\rceil$$
2. Grąžinti m.

(Išvestis: matricų eilė m)

Antrasis žingsnis. Serveryje yra formuojamos matematinių veiksmų lentelės atskirai multiplikacinei pusgrupei ir skaitiniam žiedui. Multiplikacinei pusgrupei sudaromos daugybos ir kėlimo laipsniu lenteles, o skaitiniam žiedui – sudėties ir daugybos lenteles. Iš kėlimo laipsniu lentelės išrenkami grupės ir idealo generatoriai. Formuojami du masyvai: pogrupio Z_n^* generatoriai $\mathbf{\Gamma_1}$ ir idealo $Id_q(Z_n)$ generatoriai $\mathbf{\Gamma_2}$. Šių veiksmų algoritmai atrodo taip:

4 algoritmas. Multiplikacinės pusgrupės $Z_n^{\#}$ daugybos lentelės generavimas

(Įvestis: masyvas *MultSemigroup*, skaičiavimų modulis – multiplikacinės pusgrupės parametras n.)

1. Su kiekvienu i nuo 0 iki $n - 4$ vykdyti:
   1.1. Su kiekvienu j nuo i iki $n - 4$ vykdyti:
      1.1.1. *PlatformMultTable*[i][j] = (*MultSemigroup*[i] * *MultSemigroup*[j]) % n;
      1.1.2. *PlatformMultTable*[j][i] = *PlatformMultTable*[i][j];
2. Grąžinti *PlatformMultTable*.

(Išvestis: multiplikacinės pusgrupės $Z_n^{\#}$ daugybos lentelė)

Pastaba. Algoritmo aprašyme * reiškia įprastą daugybą tarp dviejų masyvo elementų, o % – dalybos iš n liekaną.

Prieš aprašant kėlimo laipsniu lentelės generavimo algoritmą pateiksime reiškinio a^b mod n, t.y. elemento a kėlimo laipsniu b baigtinėje pusgrupėje, skaičiavimo algoritmą. Šis algoritmas vadinamas kvadratinimo ir daugybos algoritmu ir yra paremtas lygybe

$$a^b = a^{b_0+2b_1+\ldots+2^k b_k} = a^{b_0} a^{2b_1} \ldots a^{2^k b_k},$$

čia skaičius $b = b_k \ldots b_1 b_0$ yra išreiktas bitais.

5 algoritmas. Kėlimas laipsniu kvadratinimo ir daugybos metodu

(Įvestis: elementas *a*, laipsnis *b*, skaičiavimų modulis *n*.)

1. Laipsnis *b* išreiškiamas bitais. Rezultatas: „0“ ir „1“masyvas *B*. Šio masyvo ilgis yra *lenB*, o elementai $B[i] = b_{lenB-1-i}$
2. Sukuriamas darbinis masyvas *ElemPowers*. *ElemPowers*[0] = *a*.
3. Su kiekvienu *i* nuo 1 iki lenB – 1 vykdyti:
   3.1. *ElemPowers*[*i*] = (*ElemPowers*[*i* – 1] ^ 2) % *n*.
4. Sukuriamas kintamasis *rez* daugybos rezultatui kaupti. Šiam kintamajam priskiriamas skaičius 1.
5. Su kiekvienu *i* nuo 0 iki *lenB* – 1 vykdyti:
   5.1. Jeigu *B*[*i*] = 1, tai *rez* = (*rez* * *ElemPowers*[*lenB* – 1 – *i*]) % *n*.
6. Grąžinti *rez*.

(Išvestis: a^b dalybos iš *n* liekana)

Pastaba. Algoritmo aprašyme ^ reiškia įprastą elemento *a* kėlimą kvadratu.

Šio algoritmo privalumas yra tai, kad jis leidžia skaičiuoti dalybos liekaną ir tuo atveju, kai skaičius a^b yra pakankamai didelis. Taip yra dėl to, kad atliekant kėlimą kvadratu algoritmo žingsnyje 3.1 ir daugybą žingsnyje 5.1 iš karto yra skaičiuojama liekana moduliu *n*. Taip išvengiama darbo su dideliais skaičiais. Šis algoritmas yra naudojamas visur, kai reikia kelti elementą laipsniu. Taip pat jis naudojamas ir kėlimo laipsniu lentelei sudaryti.

6 algoritmas. Multiplikacinės pusgrupės $Z_n^{\#}$ kėlimo laipsniu lentelės generavimas

(Įvestis: masyvas *MultSemigroup*, Karmaiklo funkcijos reikšmė *r*, skaičiavimų modulis *n*.)

1. Su kiekvienu *i* nuo 0 iki *n* – 4 vykdyti:
   1.1. Su kiekvienu *j* nuo 0 iki *r* – 1 vykdyti:
      1.1.1. *PlatformExpTable*[*i*][*j*] = Power(*MultSemigroup*[*i*], *MultSemigroup*[*j*], *n*);
2. Grąžinti *PlatformExpTable*.

(Išvestis: multiplikacinės pusgrupės $Z_n^{\#}$ kėlimo laipsniu lentelė)

Pastaba. Algoritmo aprašyme funkcija Power(a, b, n) yra kėlimas laipsniu naudojant 7.6 algoritmą.

Naudojant šią lentelę išrenkame pogrupio $\boldsymbol{Z}_n^*$ ir idealo $Id_q(\boldsymbol{Z}_n)$ generatorius.

7 algoritmas. Pogrupio ir idealo generatorių paieška

(Įvestis: matrica *PlatformExpTable*, skaitinio žiedo parametras r, multiplikacinės pusgrupės parametras n)

1. Sukuriami du darbiniai kintamieji *indgr* ir *indid* indeksams kaupti. Jų pradinės reikšmės yra 0. Taip pat apskaičiuojamas papildomas kintamasis $s = \frac{p-1}{2}$.
2. Su kiekvienu i nuo 0 iki $n - 4$ vykdyti:
   2.1. Jeigu *PlatformExpTable*[i][3] = *PlatformExpTable*[i][0] arba *PlatformExpTable*[i][s + 1] = *PlatformExpTable*[i][0], tai pereiti prie kitos i reikšmės. Priešingu atvėju vykdyti:
   2.1.1.Jeigu gcd(i, n) = 1, tai vykdyti:
   2.1.1.1. *Gamma*1[*indgr*] = i;
   2.1.1.2. *indgr* = *indgr* + 1;
   2.1.2.Priešingu atveju vykdyti:
   2.1.2.1. *Gamma*2[*indid*] = i;
   2.1.2.2. *indid* = *indid* + 1;
3. Grąžinti *Gamma*1 ir *Gamma*2.

(Išvestis: du generatorių masyvai $\boldsymbol{\Gamma_1}$ ir $\boldsymbol{\Gamma_2}$)

Sudėties ir daugybos lentelių generavimo algoritmų čia nepateikiame, kadangi šie algoritmai yra analogiški algoritmui 7.5.

Trečiasis žingsnis. Naudojant pseudoatsitiktinių skaičių generatorių (PASG) serveryje generuojama matrica Q, kurios elementai yra generatoriai (vienas – idealo, likusieji – pogrupio).

8 algoritmas. Matricos *Q* generavimas

(Įvestis: masyvai *Gamma*1 ir *Gamma*2, matricos eilė m)

1. Sukuriami du darbiniai kintamieji *indi* ir *indj* indeksams atsiminti. Jų reikšmės yra pasirenkamos atsitiktinai iš intervalo [0, m – 1]. Taip pat nustatomi masyvų *Gamma*1 ir *Gamma*2 ilgiai *lenG*1 ir *lenG*2.
2. Su kiekvienu i nuo 0 iki m – 1 vykdyti:
   2.1. Su kiekvienu j nuo 0 iki m – 1 vykdyti:
   2.1.1.Jeigu i = *indi* ir j = *indj*, tai vykdyti:
   2.1.1.1. Generuojamas indeksas *ind* iš intervalo [0, *lenG*2];
   2.1.1.2. *Q*[i][j] = *Gamma*2[*ind*];
   2.1.2.Priešingu atveju vykdyti:
   2.1.2.1. Generuojamas indeksas *ind* iš intervalo [0, *lenG*1];
   2.1.2.2. *Q*[i][j] = *Gamma*1[*ind*];

3. Grąžinti Q.

(Išvestis: platforminė matrica Q)

Ketvirtasis žingsnis. Naudojant PASG serveryje generuojamos Žordano matricos J_1 ir J_2. Kiekvieną šių matricų, sudaro 2 Žordano langeliai su skirtingomis tikrinėmis reikšmėmis. Šios matricos yra naudojamos atitinkamai matricoms Z_1 ir Z_2 apskaičiuoti. Matricos J_1 generavimo algoritmas atrodo taip:

9 algoritmas. Matricos *J* generavimas

(Įvestis: skaitinio žiedo parametras r, matricos eilė m)

1. Atsitiktinai pasirenkamas viršutinio Žordano langelio dydis m_1.
2. Generuojamos dvi skirtingos tikrinės reikšmės $mu1$ ir $mu2$ iš intervalo $[1, r]$.
3. Su kiekvienu i nuo 0 iki $m_1 - 1$ vykdyti:
   3.1. $J[i][i] = mu1$;
   3.2. Jeigu $i \neq m_1 - 1$, tai $J[i][i+1] = 1$.
4. Su kiekvienu i nuo m_1 iki $m - 1$ vykdyti:
   4.1. $J[i][i] = mu2$;
   4.2. Jeigu $i \neq m - 1$, tai $J[i][i+1] = 1$.
5. Grąžinti J.

(Išvestis: Žordano matrica J)

Generuojant matricą J_1 Žordano langelio dydis yra įrašomas į atmintį. Kai generuojama matrica J_2 naudojamas algoritmas modifikuojamas taip, kad naujai sugeneruotas parametras m_1 nesutaptų su pirmos matricos atitinkamu parametru. Tuo atveju, kai gaunama šių parametrų lygybė, matricai J_2 generuojama nauja m_1 reikšmė. Atlikus matricų generavimą informacija apie Žordano langelio dydį gali būti pašalinta iš atminties. Šio etapo rezultatas yra dvi tarpusavyje nekomutuojančios Žordano matricos J_1 ir J_2, kurių struktūra yra tokia:

$$J = \left(\begin{array}{ccccc|ccccc} \mu_1 & 1 & & & & & & & & \\ & \mu_1 & \ddots & & & & & & & \\ & & \ddots & 1 & & & & & & \\ & & & \mu_1 & & & & & & \\ \hline & & & & & \mu_2 & 1 & & & \\ & & & & & & \mu_2 & \ddots & & \\ & & & & & & & \ddots & 1 & \\ & & & & & & & & \mu_2 & \end{array}\right)$$

Penktasis žingsnis. Naudojant PASG serveryje generuojama neišsigimusi matrica T, kurios elementai atsitiktinai pasirenkami iš skaitinio žiedo $\boldsymbol{Z_r}$, ir

apskaičiuojama atvirkštinė matrica T^{-1}. Atvirkštinės matricos skaičiavimas atliekamas naudojant šią teiginio 4.10 išvadą apie atvirkštinę matricą:

Išvada. Jeigu egzistuoja matricų T_p ir T_q atvirkštinės matricos T_p^{-1} ir T_q^{-1}, tai matricos T atvirkštinė matrica

$$T^{-1} = T_p^{-1}\mathbf{1}_p + T_q^{-1}\mathbf{1}_q \tag{7.3}$$

Kadangi pagal mūsų reikalavimus $r = 2s$, čia s yra pirminis skaičius, tai skaičiuojant atvirkštinę matricą T^{-1} reikia apskaičiuoti matricų $T_2 = T \bmod 2$ ir $T_s = T \bmod s$ atvirkštines matricas atitinkamai laukuose $\boldsymbol{Z_2}$ ir $\boldsymbol{Z_s}$, o tada panaudoti lygybę (7.3). Mūsų atvėju $\mathbf{1_2} = s$, o $\mathbf{1_s} = s + 1$. Skaičiuojant matricų T_2 ir T_s atvirkštines matricas panaudosime Gauso metodo modifikaciją, skirtą baigtiniams laukams. Kadangi naudojant Gauso metodą susidūriame su elementų dalyba, tai turime šį veiksmą pakeisti daugyba iš atvirkštinio elemento. Taip pat, jei bent viena iš matricų T_p^{-1} arba T_q^{-1} neegzistuoja, tai matricą T laikysime išsigimusia. Šio etapo algoritmai atrodo taip:

10 algoritmas. Atvirkštinės matricos skaičiavimas baigtiniame lauke Z_s naudojant modifikuotą Gauso metodą

(Įvestis: matrica T ir jos eilė m, skaičiavimų modulis – pirminis skaičius s)

1. Sukuriama vidinė darbinė matrica B. Pradinė reikšmė – vienetinė m-tos eilės matrica. Ieškomą atvirkštinę matricą žymėsime *Tinv*.
2. Su kiekvienu k nuo 0 iki $m - 2$ vykdyti:
   2.1. Rasti k-tojo stulpelio didžiausią elementą ir šio elemento stulpelio indeksą l. Jeigu gautas elementas lygus 0, tai išeiti iš algoritmo ir grąžinti *false*.
   2.2. Jeigu $k \neq l$, tai sukeisti k-tąją ir l-tąją eilutes vietomis. Ši operacija atliekama ir matricos T ir darbinės matricos B atitinkamų eilučių elementams.
   2.3. Su kiekvienu i nuo k iki $m - 1$ vykdyti:
      2.3.1. $tinv$ = Power($T[k-1][k-1], s-2, s$);
      2.3.2. $x = (T[i][k-1] * tinv) \% s$;
   2.4. Su kiekvienu j nuo k iki $m - 1$ vykdyti:
      2.4.1. $T[i][j] = (T[i][j] - x * T[k-1][j]) \% s$;
   2.5. Su kiekvienu j nuo 0 iki $m - 1$ vykdyti:
      2.5.1. $B[i][j] = (B[i][j] - x * B[k-1][j]) \% s$;
3. Jeigu $T[m-1][m-1] = 0$, tai išeiti iš algoritmo ir grąžinti *false*.
4. Su kiekvienu j nuo 0 iki $m - 1$ vykdyti
   4.1. $tinvelem$ = Power($T[m-1][m-1], s-2, s$);
   4.2. $Tinv[m-1][j] = (B[m-1][j] * tinvelem) \% s$;
   4.3. Su kiekvienu i nuo $m - 2$ iki 0 vykdyti:
      4.3.1. $x = 0$;
      4.3.2. Su kiekvienu k nuo i iki $m - 1$ vykdyti:
         4.3.2.1. $x = (x + T[i][k] * X[k][j]) \% s$;
         4.3.2.2. $tinvelem$ = Power($T[i][i], s-2, s$);

4.3.2.3. $Tinv[i][j] = ((B[i][j] - x) * tinvelem) \% s$;

5. Grąžinti *Tinv*.

(Išvestis: Atvirkštinė matrica baigtiniame lauke T^{-1} arba *false*)

Pastaba. Kadangi laipsnis $s - 2$ gali būti lygus ir 0, tai tokiu atvėju metodas Power(a, 0, s) grąžina 1, t.y $a^0 = 1$. Ši salyga taip pat galioja ir idealo elementams.

Matome, kad elemento a atvirkštinis elementas a^{-1} yra skaičiuojamas keliant šį elementą $(s - 2)$-uoju laipsniu. Toks skaičiavimo būdas yra paremtas Oilerio teoremos išvada, t.y.

$$a^{-1} = a^{s-2} \bmod s . \tag{7.4}$$

Naudojant 10 algoritmą galime apskaičiuoti atvirkštinę matricą baigtiniame lauke, bet ne žiede. Norint apskaičiuoti atvirkštinę matricą baigtiniame žiede turime panaudoti kinų liekanų teoremą, o tada naudojant algoritmą 7.11 apskaičiuoti atvirkštines matricas kiekviename lauke atskirai. Plačiau šis algoritmas yra aprašytas žemiau.

11 algoritmas. Atvirkštinės matricos skaičiavimas baigtiniame žiede Z_r

(Įvestis: matrica *T* ir jos eilė *m*, skaičiavimų modulis – sudėtinis skaičius $r = 2s$)

1. Sukriamos dvi darbinės matricos $T2 = T \% 2$ ir $Ts = T \% s$, t.y. kiekvienas matricos *T* elementas redukuojamas moduliu 2 ir moduliu *s*.
2. *T*2*inv* = GaussInverse(*T*, *m*, 2). Jeigu skaičiavimo rezultatas yra *false*, tai išeiti iš algoritmo ir grąžinti *false*;
3. *Tsinv* = GaussInverse(*T*, *m*, *s*). Jeigu skaičiavimo rezultatas yra *false*, tai grąžinti *false*;
4. *Tinv* = (*T*2*inv* * *s* + *Tsinv* * (s + 1)) % *r*.
5. Grąžinti *Tinv*.

(Išvestis: Neišsigimusios matricos *T* atvirkštinė matrica T^{-1} arba *false*)

Pastaba. Algoritmo aprašyme metodas GaussInverse(*T*, *m*, *s*) reiškia atvirkštinės matricos skaičiavimą Gauso metodu, o atvirkštinės matricos laukuose ir žiede žymimos atitinkamai *T*2*inv*, *Tsinv* ir *Tinv*.

Tada neišsigimusios matricos *T* generavimo algoritmas atrodo taip:

12 algoritmas. Matricos *T* generavimas

(Įvestis: skaitinio žiedo parametras *r*, matricos eilė *m*)

1. Atsitiktinai sugeneruoti matricos *T* elementus iš intervalo [0; *r*].
2. *Tinv* = Inverse(*T*, *m*, *r*). Jeigu rezultatas yra *false*, grįžti prie 1. žingsnio.
3. Grąžinti *T* ir *Tinv*.

(Išvestis: Neišsigimusi matrica *T* ir jos atvirkštinė matrica T^{-1})

Pastaba. Algoritmo aprašyme metodas Inverse(*T*, *m*, *r*) reiškia atvirkštinės matricos skaičiavimą pagal algoritmą 7.12.

Šeštasis žingsnis. Naudojant skaitinio žiedo aritmetinių veiksmų lenteles serveryje apskaičiuojamos matricos Z_1 ir Z_2. Kadangi šiame etape naudojama matricų daugyba, tai pirmasis šio etapo algoritmas atrodo taip:

13 algoritmas. Matricų daugyba naudojant paieškos lenteles

(Įvestis: kvadratinės matricos A ir B, šių matricų eilė m)

1. Su kiekvienu i nuo 0 iki $m - 1$ vykdyti:
   1.1. Su kiekvienu j nuo 0 iki $m - 1$ vykdyti:
      1.1.1. $C[i][j] = 0$;
      1.1.2. Su kiekvienu k nuo 0 iki $m - 1$ vykdyti:
         1.1.2.1. $tmp = MultTable[A[i][k]][B[k][j]]$;
         1.1.2.2. $C[i][j] = AddTable[C[i][j]][tmp]$.
2. Grąžinti C.

(Išvestis: Matrica $C = AB \bmod r$)

Pastaba. Algoritmo aprašyme matricos *AddTable* ir *MultTable* yra atitinkamai skaitinio žiedo sudėties ir daugybos lentelės.

Reikia pastebėti, kad naudojant paieškos lenteles turime įsitikinti tuo, kad iš paieškos lentelės pasiimame reikiamą elementą. Tačiau, kadangi skaitinio žiedo elemento indeksas masyve *NumRing* sutampa su pačiu elementu, tai vietoj elemento indekso naudojame patį elementą.

Kadangi $Z_1 = T^{-1}J_1T$ ir $Z_2 = T^{-1}J_2T$, tai algoritmas matricoms apskaičiuoti yra tas pats. Šis algoritmas atrodo taip:

14 algoritmas. Matricos *Z* skaičiavimas

(Įvestis: kvadratinės matricos *Tinv*, *J* ir *T*, šių matricų eilė *m* ir skaitinio žiedo aritmetinių veiksmų lentelės)

1. Z = MatrixMultiplication(*Tinv*, *J*, *m*)
2. Z = MatrixMultiplication(Z, *T*, *m*)
3. Grąžinti Z.

(Išvestis: viešasis MLAŠ protokolo parametras – matrica Z)

Pastaba. Algoritmo aprašyme metodas MatrixMultiplication(*Tinv*, *J*, *m*) reiškia matricų daugybą pagal algoritmą 7.14.

Visa informacija, kuri buvo gauta sėkmingai įvykdžius visus šešis etapus yra saugojama faile ServerDataLayer.dat. Sistemos kliento agentas prieš generuojant savo raktus parsisiunčia šį failą iš serverio ir užkrauna jį. Tokiu būdu klientas gauna iš serverio suformuotas lenteles ir viešąsias MLAŠ protokolo matricas Q, Z_1 ir Z_2.

Kliento slaptojo ir viešojo raktų generavimas

Pirmojo kliento agento naudojimo metu yra sukuriamas vartotojo slaptasis ir viešasis raktai. Šio etapo žingsniai atrodo taip:

Pirmasis žingsnis. Klientas generuoja slaptą raktą – matricų $\{X, U\}$ porą. Generuojant šiuos duomenis klientas naudoja informaciją, kurią gavo iš serverio, t.y sugeneruotus parametrus p, n, r, m ir algebrines struktūras, paieškos lenteles ir viešąsias matricas. Šio žingsnio veiksmai yra aprašyti žemiau.

Naudojant algoritmą 7.10 klientas generuoja Žordano matricą J, kurios elementai priklauso skaitiniam žiedui $\boldsymbol{Z}_r$. Šio algoritmo antrajame žingsnyje klientas papildomai patikrina ar sugeneruotos tikrinės reikšmės ir skaitinio žiedo parametras r yra tarpusavyje pirminiai skaičiai. Jeigu ši sąlyga vienai iš sugeneruotų tikrinių reikšmių netenkinama, tai generuojama kita reikšmė. Tokiu būdu yra užtikrinama, kad sugeneruota Žordano matrica turi atvirkštinę matricą J^{-1}. Šios matricos skaičiavimas gali būti atliktas greičiau, jeigu vietoj Gauso metodo panaudosime teorinę šios matricos išraišką, kuri atrodo taip:

$$J^{-1} = \left(\begin{array}{cccc|ccccc} \mu_1^{-1} & -\mu_1^{-2} & \ddots & (-1)^{m_1}\mu_1^{m_1} & & & & & \\ & \mu_1^{-1} & \ddots & \ddots & & & & & \\ & & \ddots & -\mu_1^{-2} & & & & & \\ & & & \mu_1^{-1} & & & & & \\ \hline & & & & \mu_2^{-1} & -\mu_2^{-2} & \ddots & (-1)^{m_2}\mu_2^{m_2} & \\ & & & & & \mu_2^{-1} & \ddots & \ddots & \\ & & & & & & \ddots & -\mu_2^{-2} & \\ & & & & & & & \mu_2^{-1} & \end{array}\right) \qquad (7.5)$$

Šios matricos skaičiavimo algoritmas atrodo taip:

15 algoritmas. Atvirkštinės Žordano matricos J^{-1} generavimas

(Įvestis: Žordano matrica J ir šios matricos eilė m, viršutinio Žordano langelio dydis m_1, skaitinio žiedo parametras – sudėtinis skaičius $r = 2s$)

1. Sukuriamas darbinis kintamasis *lambda* = *J*[0][0].
2. Sukuriami darbiniai kintamieji *temp*, *invlambda*, *invlambda*1, *minus*. Pradinės reikšmės: *temp* =1, *invlambda* = 1, *invlambda*1 = 0, *minus* = 1.
3. Jeigu didžiausias bendras daliklis gcd(*lambda*, *r*) = 1, tai apskaičiuojamas *invlambda* = Power(*lambda*, *s* – 2, *r*) ir pereinama prie ketvirto žingsnio. Priešingu atveju grąžinamas *false*.
4. *invlambda*1 = *invlambda*;
5. Sukuriamas darbinis kintamasis *indexj*, kurio paskirtis yra kaupti stulpelio indekso postūmį. Pradinė šio kintamojo reikšmė: *indexj* = 1.
6. Su kiekvienu *i* nuo 0 iki $m_1 - 1$ vykdyti:
   6.1. *Jinv*[*i*][*i*] = *invlambda*1;
   6.2. *minus* = *minus* * (–1);
   6.3. *temp* = *MultTable*[*invlambda*1][*invlambda*];
   6.4. *invlambda* = *temp*;

6.5. *temp* = (*r* + *temp* * *minus*)% *r*;
6.6. Su kiekvienu *j* nuo 0 iki m_1 – *indexj* vykdyti:
6.6.1. *Jinv*[*j*][*j* + *indexj*] = *temp*;
6.7. *indexj* = *indexj* + 1;
7. Imama kita tikrinė reikšmė, t.y. lambda = *J*[m_1][m_1]. Toliau atliekami žingsniai 2-6 su šia reikšme. 6 žingsnyje pakeičiami rėžiai: indeksas *i* kinta nuo m_1 iki *m* – 1, o indeksas *j* – nuo m_1 iki *m* – *indexj*.
8. Grąžinti *Jinv*.

(Išvestis: Žordano matricos atvirkštinė matrica J^{-1})

Naudojant savo sugeneruotas matricas *J* ir J^{-1} ir matricas *T* ir T^{-1}, kurias klientas gavo iš serverio, klientas, naudodamas algoritmą 7.15 apskaičiuoja matricas *X* ir X^{-1}, kadangi $X = T^{-1}JT$ ir $X^{-1} = T^{-1} J^{-1}T$. Matricos *T* ir T^{-1} neprivalo sutapti su tomis matricomis, kurios buvo naudojamos matricoms Z_1 ir Z_2 apskaičiuoti ir gali būti generuojamos serveryje kiekvienam klientui atskirai.

Generuodamas antrąją slaptojo rakto dalį – matricą *U*, klientas generuoja du koeficientų rinkinius – t.y. du vektorius, kuriuos sudaro *m* atsitiktinių skaitinio žiedo $\boldsymbol{Z_r}$ elementų. Šie vektoriai yra naudojami funkcijai $f_U(Z_1, Z_2)$ apskaičiuoti. Mes pasirinkome šią funkciją apskaičiuoti dauginat du daugianarius $P_1(Z_1)$ ir $P_2(Z_2)$, kurių koeficientai yra sugeneruotų vektorių elementai, t.y. $f_U(Z_1, Z_2) = P_1(Z_1)P_2(Z_2)$. Tuomet generavimo algoritmas atrodo taip:

16 algoritmas. Matricos *U* generavimas

(Įvestis: kvadratinės matricos Z_1 ir Z_2, kurios buvo sugeneruotos serveryje, šių matricų eilė *m*)

1. Generuojami du vektoriai *vector*1 ir *vektor*2, kuriuos sudaro *m* elementų. Šie vektoriai yra saugojami atmintyje.
2. Sukuriamos keturios darbinės matricos *Tmp*1, *Tmp*2 ir *U*1, *U*2. Pradinė matricų *Tmp*1 ir *Tmp*2 reikšmė – *m*-tos eilės vienetinė matrica, o matricų *U*1 ir *U*2 pradinė reikšmė yra *m*-tos eilės nulinė matrica.
3. Su kiekvienu *i* nuo 0 iki m – 1 vykdyti:
   3.1. *Tmp*3 = MultMatrixByElement(*vector*1[*i*], *Tmp*1, *m*);
   3.2. *Tmp*4 = MultMatrixByElement(*vector*2[*i*], *Tmp*2, *m*);
   3.3. *U*1 = AddMatrices(*U*1, *Tmp*3, *m*);
   3.4. *U*1 = AddMatrices(*U*2, *Tmp*4, *m*);
   3.5. *Tmp*1 = MatrixMultiplication(*Tmp*1, *Z*1, *m*);
   3.6. *Tmp*2 = MatrixMultiplication(*Tmp*2, *Z*2, *m*);
4. *U* = MatrixMultiplication(*U*1, *U*2, *m*);
5. Grąžinti *U*.

(Išvestis: Matrica $U = f_U(Z_1, Z_2)$)

Pastaba. Algoritmo aprašyme matome du metodus, kurių nepateikėme anksčiau: AddMatrices(*A*, *B*, *m*) ir MultMatrixByElement(*a*, *A*, *m*). Pirmasis jų sudeda matricas *A* ir *B* naudojant sudėties lentelę, o antrasis – padaugina matricos *A*

elementus iš a naudojant daugybos lentelę. Algoritmų aprašymo nepateikėme, kadangi jie yra panašūs į algoritmą 7.14.

Informacija, kuri gauta po šio žingsnio yra saugojama įrenginyje ir neturi būti pasiekiama kitiems klientams. Taip pat, kaip jau buvo minėta šeštame skyriuje matricos U galima nesaugoti įrenginio atmintyje, t.y. šią matricą galima ištrinti, kai bus baigtas antrasis žingsnis, kurio veiksmai aprašyti žemiau. Esant poreikiui šią matricą galima atkurti pagal algoritmą 7.17 praleidus pirmąjį šio algoritmo žingsnį, o sugeneruotus daugianarių vektorius paduodant per metodo parametrus.

Antrasis žingsnis. Klientas generuoja viešą raktą – matricų $\{A_1, A_2, E\}$ trejetą. Generuojant šiuos duomenis klientas naudoja serverio viešąsias matricas ir savo slaptąjį raktą. Kadangi matricos A_1 ir A_2 yra apskaičiuojamos pagal lygybę (4.21), tai šios matricos yra gaunamos naudojant algoritmą 7.15. Norint apskaičiuoti matricą E, turime panaudoti MLF lygybes (4.2), (4.4) ir savybę (4.11). Taigi pirmiausia turime aprašyti vienpusių MLF skaičiavimo algoritmus. Jie atrodo taip:

17 algoritmas. Kairiosios MLF skaičiavimas naudojant paieškos lenteles

(Įvestis: kairysis matricinis laipsnis – matrica X, kuri yra kliento slaptojo rakto dalis, platforminė matrica Q – viešoji matrica, kurią klientas gauna iš serverio ir šių matricų eilė m)

1. Su kiekvienu i nuo 0 iki m – 1 vykdyti:
    1.1. Su kiekvienu j nuo 0 iki m – 1 vykdyti:
        1.1.1. $C[i][j] = 1$;
        1.1.2. Su kiekvienu k nuo 0 iki m – 1 vykdyti:
            1.1.2.1. $qindex$ = GetIndexByElement($Q[k][j]$, *MultSemigroup*);
            1.1.2.2. $tmp = PlatformExpTable[qindex][X[i][k]]$;
            1.1.2.3. $qindex$ = GetIndexByElement(tmp, *MultSemigroup*);
            1.1.2.4. $C[i][j] = PlatformMultTable\ [C[i][j]][qindex]$;
2. Grąžinti C.

(Išvestis: Matrica $C = {}^{X}Q$)

18 algoritmas. Dešiniosios MLF skaičiavimas naudojant paieškos lenteles

(Įvestis: platforminė matrica Q – viešoji matrica, kurią klientas gauna iš serverio, dešinysis matricinis laipsnis – matrica U, kuri yra kliento slaptojo rakto dalis ir šių matricų eilė m)

1. Su kiekvienu i nuo 0 iki m – 1 vykdyti:
    1.1. Su kiekvienu j nuo 0 iki m – 1 vykdyti:
        1.1.1. $D[i][j] = 1$;
        1.1.2. Su kiekvienu k nuo 0 iki m – 1 vykdyti:
            1.1.2.1. $qindex$ = GetIndexByElement($Q[i][k]$);
            1.1.2.2. $tmp = PlatformExpTable[qindex][U[k][j]]$;
            1.1.2.3. $qindex$ = GetIndexByElement(tmp);
            1.1.2.4. $C[i][j] = PlatformMultTable\ [C[i][j]][qindex]$;

2. Grąžinti D.

(Išvestis: Matrica $D = Q^U$)

Iš 17 ir 18 algoritmų matome, kad prieš imant elementą iš paieškos lentelės reikia papildomai rasti matricos Q elemento q_{ik} indeksą masyve *MultSemigroup*. Taip yra dėl to, kad šis indeksas nesutampa su pačiu elementu. Šiam tikslui naudojamas metodas GetIndexByElement(a), kuris atrodo taip:

19 algoritmas. Elemento indekso multiplikacinėje pusgrupėje skaičiavimas

(Įvestis: elementas a)

1. Jeigu a priklauso intervalui $(0, p)$, grąžinti $a - 1$;
2. Jeigu a priklauso intervalui $(p, 2p)$, grąžinti $a - 2$;
3. Jeigu a priklauso intervalui $(2p, 3p)$, grąžinti $a - 3$;

(Išvestis: elemento a indeksas masyve *MultSemigroup*)

Naudojant algoritmus 7.18 ir 7.19 klientas apskaičiuoja matricą E.

20 algoritmas. Matricos *E* skaičiavimas

(Įvestis: kvadratinės matricos *Tinv*, J ir T, šių matricų eilė m ir skaitinio žiedo aritmetinių veiksmų lentelės)

1. E = LeftMatrixPowerFunction(X, Q, m)
2. E = RightMatrixPowerFunction (E, U, m)
3. Grąžinti E.

(Išvestis: matrica $E = {}^X Q^U$)

Antrame žingsnyje gautus rezultatus klientas įrašo į failą CryptoClient.key ir persiunčia serveriui. Šio failo struktūra yra tokia

1. lentelė Failo CryptoClient.key struktūra.

Matrica A_1	Matrica A_2	Matrica E

Duomenų užšifravimas ir iššifravimas

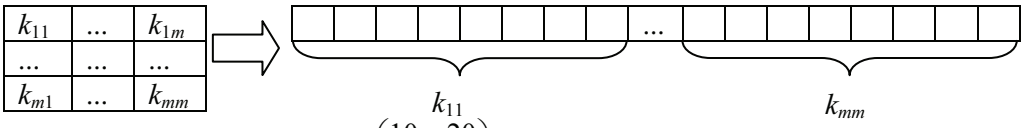
Siusdamas savo pranešimą adresatui šifruotojas pirmiausia pateikia užklausą serveriui ir gauna failą CryptoClient_ReceiverID.key, kuriame įrašytas adresato viešasis raktas. Šifruotojas nuskaito serverio atsiustą failą ir taip nustato matricas A_1, A_2 ir E. Naudojant viešuosius duomenis ir adresato raktą šifruotojas užšifruoja pranešimą atlikdamas keturis užšifravimo režimo žingsnius, kurie atrodo taip:

***Pirmasis žingsnis*.** Naudodamas algoritmus 7.15 – 7.17 šifruotojas generuoja vienkartinį slaptąjį raktą – matricų (Y, V) porą.

***Antrasis žingsnis*.** Naudodamas adresato viešąjį raktą šifruotojas apskaičiuoja užšifravimo raktą. Šiame žingsnyje šifruotojas apskaičiuoja matricą XVX^{-1} pagal 7.17 algoritmą vietoj matricų Z_1 ir Z_2 naudodamas adresato viešojo rakto matricas A_1 ir A_2. Užšifravimo raktui K gauti šifruotojas naudoja algoritmą 7.21, kai parametrai yra tokie:

- Kairysis matricinis laipsnis – matrica XVX^{-1};
- Platforminė matrica – Aldonos viešojo rakto matrica E;
- Dešinysis matricinis laipsnis – vienkartinio slaptojo rakto matrica Y.

***Trečiasis žingsnis*.** Šiame žingsnyje vyksta duomenų užšifravimas. Pirmiausiai yra pertvarkomas užšifravimo raktas K – šios matricos elementai yra surašomi į vieną eilutę dvejetainiu pavidalu ir tokiu būdu yra gaunamas vektorius, sudarytas iš „0“ ir „1“. Šiuo metu kiekvienas matricos K elementas yra vaizduojamas vienu baitu. Tada matricos K vaizdavimas atrodo taip:



Pavyzdžiui, matrica $K = \begin{pmatrix} 10 & 20 \\ 30 & 40 \end{pmatrix}$ po šio pertvarkymo atrodytų taip:

00001010000010100000111100010100 0,

čia pirmieji 8 bitai (00001010) yra skaičius 10, bitai nuo devinto iki šešiolikto sudaro skaičių 20 ir t.t.

Kadangi kiekvienas pranešimo simbolis taip pat gali būti pavaizduotas šešioliktainiu pavidalu naudojant vieną iš koduočių (pvz. UTF-8), tai juos galima transformuoti ir į dvejetainį pavidalą bei pavaizduoti juos vieno baito pavidalu.

Užšifravimo procesas yra XOR operacijos taikymas. Programa yra parašyta taip, kad galėtų užšifruoti ir iššifruoti tik tokius pranešimus, kurių ilgis yra nedidesnis už naudojamo šifravimo rakto K ilgį, kadangi būtent taip yra taikomi visi asimetriniai šifrai. Tuo atveju, kai šifruojamas pranešimas yra trumpesnis už raktą užšifravimui naudojama tik ta rakto dalis, kuri atitinka pranešimo ilgį. Priešingu atveju vartotojas yra informuojamas apie tai, kad šifruojamas pranešimas yra per ilgas.

***Ketvirtasis žingsnis*.** Šiame žingsnyje šifruotojas apskaičiuoja papildomus duomenis (dekriptorių), kuriuos jis siunčia adresatui kartu su šifrograma C. Dekriptorių sudaro tris matricos: $B_1 = Y^{-1}Z_1Y$, $B_2 = Y^{-1}Z_2Y$ ir $F = {}^{V}Q^{Y}$. Šioms matricoms apskaičiuoti yra naudojami algoritmai 7.15 ir 7.21. Gauta informacija įrašoma į failą šia tvarka

2 lentelė Užšifruoto pranešimo struktūra.

Matrica B_1	Matrica B_2	Matrica F	Šifrograma C

Suformuotas failas yra siunčiamas adresatui, kuris iššifruodamas jam skirtą pranešimą naudoja savo slaptąjį raktą $PrK_A = \{X, U\}$ ir gautą iš šifruotojo failą. Adresato vykdomi žingsniai yra šie:

Pirmasis žingsnis. Adresatas nuskaito duomenis iš gauto failo ir tokiu būdu sužino dekryptoriaus matricas B_1, B_2 ir F bei šifrogramą C.

Antrasis žingsnis. Šiame žingsnyje adresatas apskaičiuoja iššifravimo raktą K. Iš pradžių jis, naudodamas savo sugeneruotus daugianarių koeficientus, apskaičiuoja matricą $Y^{-1}UY$ pagal algoritmą 7.17 (šiame algoritme praleidžiamas koeficientų generavimas), o tada kelia matricą F kairiuoju matriciniu laipsniu X ir dešiniuoju matriciniu laipsniu $Y^{-1}UY$ naudodamas algoritmą 7.21.

Trečiasis žingsnis. Adresatas iššifruoja serverio pranešimą M naudodamas gautą pirmajame iššifravimo režimo žingsnyje raktą K. Kadangi šis raktas sutampa su serverio šifravimo raktu adresatas gali perskaityti jam skirtą pranešimą. Visi šiame žingsnyje duomenų pertvarkymai yra analogiški užšifravimo metu atliekamiems pertvarkymams, o iššifravimo procesas yra analogiškas užšifravimo procesui.