VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

McEliece viešojo rakto kriptografinės sistemos saugumo tyrimas

Study of the Security of McEliece Public-Key Cryptosystem

Bakalauro baigiamasis darbas

Atliko: Paulius Jachimovičius (parašas)

Darbo vadovas: dr. Gintaras Skersys (parašas)

Darbo recenzentė: dr. Audronė Lupeikienė (parašas)

Vilnius – 2021

Santrauka

Nagrinėjama problema – McEliece kriptografinės sistemos saugumo įvertinimas kvantinių kompiuterių vystymosi kontekste. McEliece kriptosistema pasižymi atsparumu P. Shor algoritmui, skirtam kvantiniams kompiuteriams, priešingai, nei dabartinės naudojamos kriptografinės sistemos, kaip, pavyzdžiui, RSA. Dėl šios priežasties reikia ieškoti alternatyvių kriptografinių sistemų, kaip, pavyzdžiui McEliece. McEliece kriptografinės sistemos saugumas buvo įvertintas įgyvendinus nekritinę (mažo svorio žodžio tiesiniame kode radimo) bei kritines (susijusio pranešimo bei pranešimo persiuntimo) atakas vykdant eksperimentus. Išanalizavus atakų algoritmus bei gautą statistiką buvo padaryta išvada, jog vertinant McEliece kriptosistemos saugumą vykdytų atakų požiūriu, pasinaudojus išvadose paminėtomis rekomendacijomis bei naudojant kriptosistemą su rekomenduojamais parametrais, McEliece kriptografinė sistema yra viena iš galimų alternatyvų dabartinėms naudojamoms kriptografinėms sistemos.

Raktiniai žodžiai: McEliece, kriptosistema, kriptografinė sistema, saugumas, kriptografija.

Summary

The considered problem is the assessment of the McEliece cryptographic system in the context of the intense development of quantum computers. The McEliece cryptosystem is resistant to the P. Shor‘s algorithm for quantum computers in contrast to currently used cryptographic systems, such as RSA. For this reason, there is a need for alternative cryptosystems to be considered, such as McEliece. The security of the McEliece cryptographic system was assessed by implementing non-critical (related message and message resend) and critical (finding low-weight codeword) attacks as well as performing experiments using them. After analyzing attacks‘ algorithms as well as obtained statistics and considering the recommendations regarding the way of using McEliece cryptosystem with recommended specified parameters, the conclusion was made that the McEliece cryptographic system is one of the possible alternatives to currently used cryptosystems.

Keywords: McEliece, cryptosystem, cryptographic system, security, cryptography.

TURINYS

ĮVADAS 6

1. MCELIECE KRIPTOSISTEMA 9
1.1. Viešojo rakto kriptosistemos koncepcija 9
1.2. Tiesiniams kodams naudojama terminologija 10
1.3. Klaidas taisantys tiesiniai kodai 10
1.4. McEliece kriptosistemos aprašymas 11
1.4.1. Kriptosistemos inicijavimas 12
1.4.2. Užšifravimo algoritmas 12
1.4.3. Dešifravimo algoritmas 12

2. ATAKOS 13
2.1. Susijusio pranešimo ir pranešimo persiuntimo atakos 13
2.1.1. Kriptoanalitinis kontekstas 13
2.1.2. Pranešimo persiuntimo atakos sąlygos 13
2.1.3. Atakos būdas 14
2.1.4. Susijusio pranešimo atakos sąlygos 15
2.1.5. Konkretus atakos pavyzdys 15
2.2. Mažo svorio žodžio tiesiniame kode radimo ataka 16
2.2.1. Mažo svorio žodžio tiesiniame kode radimo atakos algoritmo žingsniai 17
2.2.2. Mažo svorio žodžio tiesiniame kode radimo atakos algoritmo darbo sąnaudos 18

3. ATAKŲ REALIZAVIMAS 20
3.1. Apžvalga 20
3.2. Bibliotekos 20
3.3. Projektavimo šablonai bei principai 20
3.4. Susijusio pranešimo ir pranešimo persiuntimo atakų įgyvendinimas 22
3.5. Mažo svorio žodžio tiesiniame kode radimo atakos įgyvendinimas 22
4. ATAKŲ REZULTATAI 25
4.1. Susijusio pranešimo bei pranešimo persiuntimo atakų rezultatai 25
4.2. Mažo svorio žodžio tiesiniame kode radimo atakos rezultatai 29
4.1.1. Mažo svorio žodžio tiesiniame kode radimo atakos efektyviausių parametrų apskaičiavimas 29
4.1.2. Mažo svorio žodžio tiesiniame kode radimo atakos eksperimentai 32

4.3. Mažo svorio žodžio tiesiniame kode radimo atakos duomenų ekstrapoliacija.....33

REZULTATAI IR IŠVADOS.....37

ŠALTINIAI.....40

SĄVOKŲ APIBRĖŽIMAI.....41

SANTRUMPOS.....42

ĮVADAS

Beieškant informacijos internete ar beskaitant pasaulio naujienas el. dienraštyje, kibernetinis saugumas nekelia susirūpinimo. Jo svarba bei dėmesio atkreipimas padidėja tuomet, kai vykdomi pavedimai el. bankinkystėje ar siunčiami konfidencialūs dokumentai, kurių paviešinimas sukeltų dideles problemas kelių žmonių, visuomenės ar net pasaulio mastu - tai tėra tik keli situacijų pavyzdžiai, reikalaujantys saugumo. Modernios vartotojo sąsajos naršyklėse bei internetiniuose tinklalapiuose palengvina naršymo patirtį, tačiau už jų slepiasi kur kas sudėtingesni procesai, nei puslapio apačioje mirguliuojančios reklamos bei naršyklės lango viršutiniuose kampuose prisitvirtinusios vaizdo leistuvės.

Naršyklės adresų juostoje suvedus internetinio tinklalapio adresą bei nuspaudus „Enter“ klavišą, vos už kelių sekundžių naršyklė atvaizduoja puslapį. Pasigilinus į tai, kas vyksta prisijungimo prie svetainės metu, išaiškėja įdomios detalės – tiek naršyklė, tiek serveris, kurį pastaroji bando pasiekti, apsikeičia informacija, kuri padeda sužinoti, ar internetinis puslapis, prie kurio vartotojas jungiasi, yra patikimas – t.y., ar jis turi sertifikatą, įrodantį, kad puslapis yra saugus. Toliau besigilinant į detales galima užduotį esminį klausimą – ar duomenys, kuriuos naršyklė gavo iš serverio, yra saugūs? Kas užtikrina, jog nesaugiu kanalu siųstas pranešimas pakeliui nebuvo perimtas, modifikuotas ir nusiųstas atgal į naršyklę?

Kriptografija, arba kitaip - slapto rašymo mokslas - tai nuo senovės iki naujausių laikų taikomas menas, kurio ankstyviausias panaudojimas buvo aprašytas apie 1900 m. Pr. Kr., kai egiptiečių raštininkas užraše panaudojo nestandartinius hieroglifus. Jis buvo panaudotas taikant įvairias programas - pradedant diplomatinėmis misijomis ir baigiant karo laiko mūšio planais. Nenuostabu, jog netrukus po plačiai pasauly pradėtos naudoti kompiuterinės komunikacijos atsirado naujos kriptografijos formos. Duomenų ir telekomunikacijų srityje kriptografija yra būtina bendraujant nesaugiais kanalais - internetu [Kes20]. Nors dabartinės kriptografinės sistemos, kaip, pavyzdžiui, RSA, užtikrina, jog ir didelio galingumo šiuolaikiniai kompiuteriai nesugebės jų įveikti, itin svarbu atkreipti dėmesį į vieną iš šiuo metu labiausiai vystomų technologijų – kvantinius kompiuterius.

Fiziniai kvantinės mechanikos dėsniai leidžia naudoti alternatyvų metodą, kaip šiuolaikiniai kompiuteriai apdoroja informaciją. Tradiciniai kompiuteriai naudoja dvejetainę skaičiavimo sistemą, kurios galimos reikšmės (0 arba 1) yra vadinamos bitais. Tuo tarpu kvantiniuose kompiuteriuose naudojami kvantiniai bitai arba *kubitai*, kurie tuo pačiu metu gali būti 0 ir 1 pozicijų derinys. Kvantinis skaičiavimas gali pasinaudoti galingesniais *kubitais* ir vienu metu

atlikti operacijas ne tik nustatytose vertėse (0 arba 1), bet ir visose galimose superpozicijose, ko pasekoje jis pasiekia efektyvumo pranašumą prieš dvejetainį skaičiavimą tam tikriems skaičiavimams atlikti [OR20].

Asimetrinis šifravimas yra kriptografinė sistema, naudojanti viešojo bei privataus raktų porą. Kaip pavyzdys „Rivest-Shamir-Adleman“ (RSA) algoritmas yra būtent tokio tipo sistema. Pasinaudojus viešaisiais bei privačiaisiais raktais, ji leidžia atitinkamai šifruoti bei dešifruoti pranešimus, todėl jų konfidencialumas bei autentiškumas lieka nepakitę. Esminė problema yra ta, jog pakankamai galingo kvantinio kompiuterio savininkai sukeltų pavojų viešojo rakto kriptografijos sistemoms kompiuteriui atliekant dešifravimą be išankstinių žinių apie privatųjį raktą. Tai gali būti, pavyzdžiui, skaitmeniniai parašai, būtini interneto protokolai, pvz., HTTPS (TLS), reikalingi saugiam naršymui, internetinei bankininkystei, apsipirkimui internete ir t.t. [OR20].

Pokvantinė kriptografija yra paremta algoritmais, kurių saugumui kvantiniai kompiuteriai nekelia grėsmės, kol nėra įrodyta kitaip. Tai pasiekiama naudojant labai skirtingus matematinius statybinius blokus, kurie apima matematines operacijas, kurių kvantiniai kompiuteriai negali įvykdyti efektyviau nei kiti kompiuteriai [OR20]. Pokvantinė kriptografija yra skirstoma į keletą kategorijų, kaip, pavyzdžiui, daugiamatė, kodu ar maišos funkcijomis pagrįsta kriptografija. Viena iš kriptosistemų, manoma, atsparių kvantiniams kompiuteriams, yra 1978 metais matematiko Robert McEliece pasiūlyta asimetrinio rakto kriptografinė sistema, besiremianti Goppa kodų šeima bei atsitiktiniais klaidų vektoriais. Tai tiesiniais kodais grįsta sistema, kurios negalima įveikti pasinaudojus kvantiniams kompiuteriams skirtu P. Shor algoritmu sveikiesiems skaičiams faktorizuoti. Būtent šis algoritmas ir kelia grėsmę tokioms kriptografinėms sistemoms, kaip RSA.

Baigiamojo darbo tikslas yra įgyvendinus atakas nustatyti saugius McEliece kriptografinės sistemos parametrus ir pateikti saugaus naudojimo rekomendacijas.

Siekiant įgyvendinti šį tikslą, yra iškeliami atitinkami uždaviniai:

1. Susipažinti su McEliece kriptografine sistema.
2. Susipažinti su susijusio pranešimo, pranešimo persiuntimo bei mažo svorio žodžio tiesiniame kode radimo atakomis prieš McEliece kriptografinę sistemą.
3. Realizuoti bei įvykdyti antrame punkte paminėtas atakas prieš McEliece kriptosistemą renkant statistinius duomenis.
4. Išanalizuoti statistinius duomenis.

5. Numatyti, kiek laiko užtruktų įvykdyti mažo svorio žodžio tiesiniame kode radimo ataką su šiuo metu rekomenduojamais McEliece kriptografinės sistemos parametrais, ekstrapoliuojant gautus atakos statistinius duomenis.
6. Palyginti atakoms naudoto kompiuterio bei superkompiuterio pajėgumų skirtumus ir numatyti, kiek laiko užtruktų 5 punkte minėta ataka, jei ją vykdytų superkompiuteris.
7. Įvertinus statistinius duomenis bei atakų algoritmus padaryti išvadas.

Visų pirma, siekiant susipažinti su viešojo rakto kriptografinės sistemos veikimo principu buvo pasinaudota [Neh20] šaltiniu, kuriame viešojo rakto kriptografijos principas buvo apibrėžtas aukštu abstrakcijos lygmeniu. Siekiant suprasti bei išanalizuoti McEliece kriptografinę sistemą buvo pasinaudota pirmąja publikacija, pristatančia McEliece kriptosistemą - [McE78]. Susijusio pranešimo bei pranešimo persiuntimo atakų analizei, pristatymui bei realizavimui buvo pasinaudota [Ber97] šaltiniu, kuriame apibrėžtos šios atakos. [CS98] šaltinis buvo panaudotas mažo svorio žodžio tiesiniame kode radimo atakos algoritmo apibrėžimui ir pritaikymui. Siekiant įvykdyti eksperimentus su susijusio pranešimo bei pranešimo persiuntimo atakomis buvo pasinaudota [NMB+12] šaltiniu, kuriame nurodyti rekomenduojami McEliece kriptosistemos parametrai remiantis numatomu kompiuterių techninių charakteristikų tobulėjimu. Apskaičiuoti efektyviausius mažo svorio žodžio tiesiniame kode radimo atakos parametrus buvo pasinaudota [KI03] šaltinyje apibrėžtomis formulėmis, leidžiančiomis apskaičiuoti atakai reikalingų dvejetainių operacijų skaičių. Atlikus eksperimentus su mažais McEliece kriptografinės sistemos parametrais ir apskaičiuotais efektyviausiais mažo svorio žodžio tiesiniame kode radimo atakos parametrais buvo vykdoma ekstrapoliacija siekiant numatyti, kiek laiko užtruktų ataka su didesniais kriptosistemos parametrais. Pasinaudojus numatytais laikais didesniems kriptosistemos parametrams bei techninėmis bandymams naudoto kompiuterio bei superkompiuterio charateristikomis buvo numatyta, kiek laiko teoriškai užtruktų superkompiuteris vykdydamas mažo svorio žodžio tiesiniame kode radimo ataką.

Apskaičiuoti matematines formules buvo naudojama *Microsoft Excel* programa, atakų algoritmų realizavimui bei statistikos rinkimui – *C#* programavimo kalba.

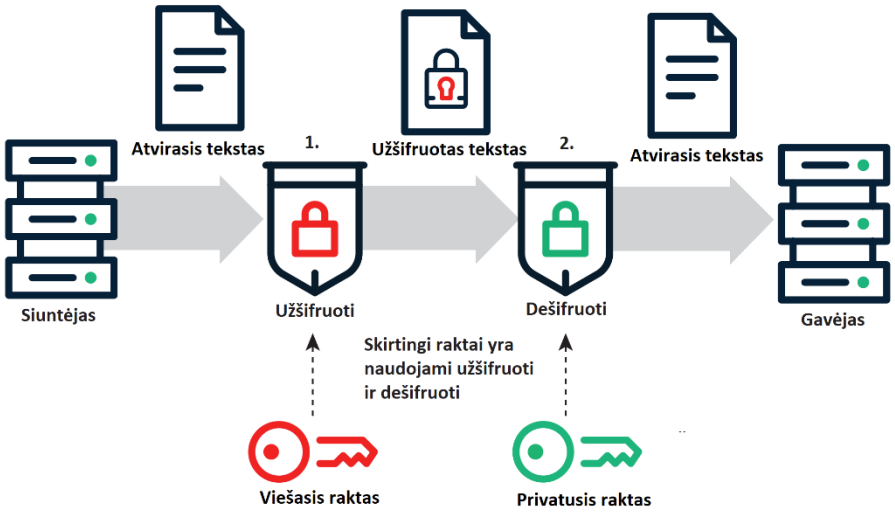
1. MCELIECE KRIPTOSISTEMA

Siekiant suprasti McEliece kriptografinę sistemą, svarbu suvokti viešojo rakto kriptosistemos koncepciją bei susipažinti su tiesiniams kodams naudojama terminologija.

1.1. Viešojo rakto kriptosistemos koncepcija

Bet kuriame viešojo rakto kriptografiniame algoritme yra penki elementai:

1. Atvirasis tekstas - tai yra skaitomas pranešimas, kuris pateikiamas kaip įvestis į šifravimo algoritmą. Viešojo rakto algoritme atvirasis tekstas šifruojamas blokais.
2. Užšifravimo algoritmas – jo vykdymo metu atliekamos viena ar keletas atvirojo teksto transformacijų pasinaudojant viešuoju raktu. Šio algoritmo vykdymas yra vadinamas šifravimu.
3. Viešieji ir privatieji raktai - tai raktų rinkinys, tarp kurių viešasis naudojamas šifravimui, o privatus - dešifravimui.
4. Užšifruotas tekstas - tai yra užšifravimo algoritmo išvestis. Jis visiškai priklauso nuo rakto, kuriuo buvo užšifruotas.
5. Dešifravimo algoritmas - jis priima užšifravimo algoritmo išvestį, t.y. užšifruotą tekstą, ir panaudoja privatųjį raktą atvirajam tekstui iš jo atkurti. Šio algoritmo vykdymas vadinamas dešifravimu [Neh20].



1 pav. Pranešimo perdavimo nesaugiu kanalu eigos diagrama

Apibrėžus algoritmo elementus galima apibrėžti veiksmus vykdant algoritmą:

1. Kiekvienas vartotojas turi sugeneruoti du raktus, iš kurių vienas bus naudojamas užšifravimo, o kitas - pranešimų dešifravimo algoritmams.

2. Kiekvienas vartotojas turi porą raktų, iš kurių viešasis bus atskleistas, o privatusis laikomas paslapty.
3. Jei vartotojas turi išsiųsti pranešimą nesaugiu kanalu, jis turi užšifruoti pranešimą (atvirąjį tekstą) naudodamas gavėjo viešąjį raktą, jog perėmus jį kanale įsilaužėlis nesuprastų žinutės. Tai iliustruota pirmuoju veiksmu 1-ame paveikslėlyje.
4. Gavęs užšifruotą tekstą, gavėjas turi dešifruoti pranešimą naudodamas savo privatųjį raktą. Tai iliustruota antruoju veiksmu 1-ame paveikslėlyje [Neh20].

1.2. Tiesiniams kodams naudojama terminologija

Šis poskyris yra skirtas apibrėžti pagrindines sąvokas, kurios bus naudojamos tolimesniuose skyriuose kalbant apie tiesinius kodus.

1. Baigtinis kūnas ($F(p^m)$, kur p – pirminis skaičius, $m \geq 1$) - tai baigtinė aibė jam priklausančių elementų ir rinkinys operacijų (sudėtis, atimtis, dalyba, daugyba), tenkinančių tam tikras taisykles. Operacijos atliekamos tarp baigtinės aibės apibrėžtų elementų.
2. Kodas – tai visų duotojo n ilgio vektorių aibės poaibis virš duotojo baigtinio kūno $F(p^m)$, kur n – kodo ilgis.
3. Tiesinis kodas – tai kodo vektorių tiesinė erdvė. Žymėjimas – $[n, k]$, kur n yra tiesinio kodo ilgis, k – tiesinio kodo dimensija, apibrėžta 7-ame punkte.
4. Hemingo atstumas tarp vektorių – tai skaičius pozicijų, kuriose skiriasi jų reikšmės.
5. Vektoriaus Hemingo svoris – tai nenulinių jo koordinačių skaičius.
6. Tiesinio kodo bazė – tai tiesiškai nepriklausomų vektorių aibė, kuria pasinaudojant galima gauti visus įmanomus pasirinkto tiesinio kodo žodžius. Iš visų kodo bazės vektorių sudaryta matrica yra vadinama generuojančia to kodo matrica.
7. Tiesinio kodo dimensija – tai jo bazės vektorių skaičius.
8. Klaidų vektorius – tai vektorius, kurio nenulinės reikšmės atitinkamose koordinatėse apibrėžia padarytas klaidas tiesinio kodo žodyje.
9. Aibės galia – tai aibės elementų skaičius.
10. Apverčiamoji matrica – turinti atvirkštinę.
11. Matricos A atvirkštinė matrica – tai matrica, kuriai galioja tokia lygybė: $AB = BA = Id_k$, kur Id yra vienetinė matrica, k – tiesinio kodo dimensija, A – apverčiamoji matrica, B – atvirkštinė matrica.

1.3. Klaidas taisantys tiesiniai kodai

Tiesiniai kodai yra klaidas taisantys kodai. Jų veikimo principą ir apsibrėžkime.

Visi tiesiniai kodai $[n, k]$ turi generuojančią matricą, kuri yra sudaryta iš visų tiesinio kodo bazės vektorių. Pasirinkus bet kokį k ilgio vektorių ir jį sudauginus su generuojančia tiesinio kodo matrica bus gaunamas to tiesinio kodo žodis, kurio ilgis yra n. Šis procesas yra vadinamas užkodavimu, nes iš pradinio k ilgio vektoriaus mes gauname kitą n ilgio vektorių, kuris priklauso naudotam tiesiniam kodui. Tuo tarpu dekodavimo proceso metu iš užkoduoto vektoriaus galima atkurti pradinį vektorių, kurio ilgis yra k. Praktikoje tiesiniai kodai neretai yra naudojami perduoti duomenis kanalais, kuriuose pasitaiko trikdžių. Trikdžiai iškraipo užkoduotus vektorius, t.y. kanale būna padaromos klaidos. Svarbu paminėti, jog dekodavimo procesas pasižymi savybe atkurti pradinį vektorių ir tuomet, kai užkoduotame vektoriuje pasitaikė klaidų, tačiau tik mažiau ar lygiai tiek, kiek tiesinis kodas gali jų ištaisyti. Ši tiesinių kodų savybė yra pritaikoma kriptografijoje.

Kriptografijoje yra pasinaudojama tiesinių kodų savybe ištaisyti kanale padarytas klaidas tokiu būdu – pats siuntėjas sugeneruoja klaidas tam, jog siunčiant vektorių nesaugiu kanalu ir jį perėmus įsilaužėliui jam nebūtų taip paprasta rasti pradinį vektorių naudojant algebros metodus (sprendžiant tiesinių lygčių sistemą), o reikėtų taikyti klaidas taisančių kodų metodus (dekodavimo procesas). Priklausomai nuo to, koks tiesinis kodas ir jo parametrai buvo naudojami, užkoduotą vektorių galima iškraipyti padarant tiek klaidų, kiek daugiausiai jų gali ištaisyti tiesinis kodas, jei jis jas iš viso gali taisyti. Iškraipymo procesas yra atliekamas sugeneruojant atsitiktinį n ilgio vektorių, kurio Hemingo svoris atitinka ar yra mažesnis už maksimalų klaidų skaičių, kurį gali ištaisyti tiesinis kodas. Turint klaidų vektorių jis yra pridedamas prie užkoduoto vektoriaus. Gavus tokį užkoduotą ir iškraipytą vektorių priklausomai nuo pasirinkto algoritmo yra vykdomas dekodavimo procesas siekiant atkurti pradinį k ilgio vektorių. Svarbu paminėti, jog jei klaidų buvo padaryta daugiau, nei tiesinis kodas gali ištaisyti, klaidų dekodavimo algoritmas nesugebės atkurti užkoduoto vektoriaus. Tokiu atveju gauti pradinį k ilgio vektorių yra praktiškai neįmanoma.

1.4. McEliece kriptosistemos aprašymas

McEliece kriptosistema yra grindžiama Goppa šeimos kodų egzistavimu. Kiekvienam pirminiam t laipsnio polinomui virš baigtinio kūno $F(2^m)$ egzistuoja dvejetainis, $n = 2^m$ ilgio Goppa kodas, kurio dimensija $k \geq n - tm$, galintis atkurti vektorių, kuriame buvo padaryta t ar mažiau klaidų, kur t ir m yra sveikieji teigiami skaičiai, o n – Goppa kodo ilgis. Taip pat egzistuoja ir greitas Patterson dekodavimo algoritmas, skirtas šio kodo dekodavimui. Jo vykdymo laikas apibrėžtas $O(nt)$ [McE78].

1.4.1. Kriptosistemos inicijavimas

Tarkime, kad sistemos projektuotojas pasirenka norimas m ir t reikšmes bei atsitiktinai pasirenka pirminį t laipsnio polinomą virš baigtinio kūno $F(2^m)$, kur t ir m yra sveikieji teigiami skaičiai, o $n = 2^m$. Atsitiktinis polinomo pasirinkimas yra sąlyginai lengvas procesas, nes tikimybė, jog jis yra pirminis, yra apie $1/t$ įskaitant faktą, jog yra ir greitas algoritmas, skirtas patikrinti, ar polinomą galima išskaidyti į mažesnio laipsnio polinomų sandaugą. Kitame žingsnyje sistemos projektuotojas sukuria Goppa kodo, kurį atitinka pasirinktas polinomas, $k \times n$ generuojančią matricą G, kur k yra Goppa kodo dimensija. Sugeneravęs matricą G, sistemos projektuotojas pasirenka atsitiktinę invertuojamą $k \times k$ matricą S bei atsitiktinę $n \times n$ perstatų matricą P. Tada jis apskaičiuoja dar vieną generuojančią matricą $G' = SGP$, kuri apibrėš tiesinį kodą, o šis praktiškai nesiskirs nuo atsitiktinai parinkto tiesinio kodo. G' generuojanti matrica kartu su klaidų vektoriaus svoriu t sudarys šios kriptosistemos viešąjį raktą, o rinkinys G, S ir P matricų – privatųjį raktą. Paskutiniame žingsnyje sistemos projektuotojas paskelbs sistemos viešąjį raktą, kurį galės naudoti tie, kurie norės nusiųsti slaptą pranešimą gavėjui (šiuo atveju projektuotojui) [McE78].

1.4.2. Užšifravimo algoritmas

Šifruojamus duomenis reikia suskirstyti į k ilgio bitų blokus, kur k yra vienas iš McEliece kriptosistemos parametrų – viešosios matricos G' eilučių skaičius, tiksliau – Goppa kodo dimensija. Jeigu u yra toks blokas, tai nesaugiu kanalu bus perduodamas $x = uG' + z$ vektorius, kur nurodytoje formulėje G' yra viešoji generuojanti matrica, o z yra lokaliai sukurtas atsitiktinis n (viešosios matricos G' stulpelių skaičius) ilgio ir t svorio klaidų vektorius (n ir t yra McEliece kriptosistemos parametrai). Gavęs x sistemos projektuotojas, kaip gavėjas, gali efektyviai atkurti u vykdydamas dešifravimo algoritmą [McE78].

1.4.3. Dešifravimo algoritmas

Dešifravimui reikia apskaičiuoti $x' = xP^{-1}$, kur P^{-1} yra atvirkštinė perstatų matrica P, o x' bus anksčiau pasirinkto Goppa kodo žodis. Kitame žingsnyje pasinaudojus Patterson algoritmu, galima apskaičiuoti $uS = u'$, kur u yra siųstas pranešimas, o S yra atsitiktinė invertuojama matrica (dalis privačiojo rakto). Galiausiai u apskaičiuojamas $u = u'S^{-1}$, kas ir yra siųstas pranešimas [McE78].

2. ATAKOS

Skyriuje aprašomos kritinės ir nekritinė atakos. Kritinės atakos pasižymi tuo, jog tam, kad jos būtų įgyvendintos, yra reikalinga papildoma informacija. Tuo tarpu nekritinėms atakoms papildoma informacija nėra reikalinga.

2.1. Susijusio pranešimo ir pranešimo persiuntimo atakos

Šis poskyris skirtas pristatyti susijusio pranešimo ataką bei specialų jos atvejį, kai žinutės sutampa, t.y., kai ta pati žinutė siunčiama nesaugiu kanalu 2 kartus.

2.1.1. Kriptoanalitinis kontekstas

Prieš apžvelgiant galimą atakos scenarijų, svarbu apsibrėžti [NMB+12] šaltinyje nurodytus saugiausius McEliece kriptografinės sistemos parametrus pagal metus (šiuo atveju 2021): $n = 1983$, $t = 46$, $k = 1473$, kur n yra Goppa kodo ilgis, t – klaidų vektoriaus Hemingo svoris, o k – Goppa kodo dimensija. Atakos scenarijus gali būti toks: jei kriptoanalitikas galėtų atspėti k skaičiaus užšifruoto vektoriaus x koordinates, kurios nebuvo iškraipytos klaidų vektoriaus z, tuomet beliktų apsibrėžti bei išspręsti matricinę lygtį. Lygties matrica būtų sudaryta iš viešosios matricos stulpelių, atitinkančių atspėtas koordinates, o lygties nežinomieji – užkoduoto vektoriaus reikšmės tose pačiose atspėtose koordinatėse. Siekiant išspręsti lygtį, reikia rasti atvirkštinę matricą, kuri gali būti randama tik tuo atveju, jei lygtys yra tiesiškai nepriklausomos, t.y. atitinkamos matricos stulpeliai (ar eilutės) yra tiesiškai nepriklausomi. Konkrečiai šią matricinę lygtį reiktų išspręsti taip – transponuoti lygties matricą bei įvykdyti Gauso eliminaciją šiai lygčiai. Jei pasirinkti viešosios matricos stulpeliai buvo tiesiškai nepriklausomi, gautas lygties sprendinys bus u – dešifruotas atvirasis tekstas. Svarbu paminėti, kad tai yra vieno pranešimo ataka; privatus sistemos raktas kriptoanalitikui lieka nežinomas [Ber97].

Kokio dydžio skaičiavimus reikia atlikti šiai atakai? Kriptoanalitikas turi teisingai atspėti k skaičiaus klaidų vektoriaus neiškraipytus stulpelius iš galimų $v = n - t$, kur v yra klaidų vektoriaus neiškraipytų stulpelių skaičius. Taigi galime apskaičiuoti, jog tam reikės $\frac{\binom{n}{k}}{\binom{v}{k}}$ spėjimų.

2.1.2. Pranešimo persiuntimo atakos sąlygos

Tarkime, kad per neapsižiūrėjimą arba dėl kriptoanalitiko veiksmų tiek $x_1 = uSGP + z_1$, tiek $x_2 = uSGP + z_2$ užšifruoti vektoriai su sąlyga, jog $z_1 \neq z_2$, yra išsiųsti nesaugiu kanalu (tai vadinsime pranešimo persiuntimo sąlyga), kur x_1 ir x_2 yra užšifruoti pranešimai, u – atvirasis pranešimo tekstas, SGP – viešasis raktas, o z_1 ir z_2 – klaidų vektoriai. Svarbu pastebėti, jog $x_1 + x_2 = z_1 + z_2 \ (mod\ 2)$. Kriptoanalitikas gali lengvai nustatyti pranešimo persiuntimo sąlygą,

stebėdamas bet kurių dviejų užšifruotų pranešimų sumos Hemingo svorį. Kai pagrindiniai pranešimai skiriasi, numatomas sumos svoris yra apie $n/2$, o kai jie identiški, sumos svoris negali viršyti $2 \times t$ [Ber97].

2.1.3. Atakos būdas

Pradžioje reikia atitinkamais elementais užpildyti dvi aibes remiantis vektorių x_1 ir x_2 suma, kur x_1 ir x_2 yra užšifruoti pranešimai. Apsibrėžkime šias aibes.

1. L_0 aibėje saugosime visas sumos koordinates, kur $(x_1 + x_2)$ yra lygus 0. $L_0 = \{l \in \{1,2,\dots,n\}: x_1(l) + x_2(l) = z_1(l) + z_2(l) = 0\}$, kur z_1 ir z_2 – klaidų vektoriai, o l – indeksų aibės $\{1,2,\dots,n\}$ elementas (n yra sveikasis teigiamas skaičius), $x(l)$ - vektoriaus x elementas, esantis koordinatėje l.
2. L_1 aibėje saugosime visas sumos koordinates, kur $(x_1 + x_2)$ yra lygus 1. $L_1 = \{l \in \{1,2,\dots,n\}: x_1(l) + x_2(l) = z_1(l) + z_2(l) = 1\}$.

Mes siekiame pasinaudoti faktu, jog:

1. $l \in L_0$ → greičiausiai nei $x_1(l)$, nei $x_2(l)$ nebuvo iškraipyti klaidų vektorių;
2. $l \in L_1$ → būtinai vienas iš $x_1(l)$ arba $x_2(l)$ buvo iškraipytas klaidų vektorių.

Kiekvienas $l \in L_0$ reiškia, jog $z_1(l) = 0 = z_2(l)$ arba $z_1(l) = 1 = z_2(l)$. Darant prielaidą, jog z_1 ir z_2 vektoriai yra pasirenkami nepriklausomai, bet kuriam l

$$\Pr\big(z_1(l) = 1 = z_2(l)\big) = \left(\frac{t}{n}\right)^2.$$

Kitaip tariant, didžioji dalis $l \in L_0$ reiškia $z_1(l) = 0 = z_2(l)$. Dėl šios priežasties kriptoanalitikas turėtų bandyti atspėti k skaičiaus klaidų vektorių neiškraipytas stulpelių koordinates iš L_0 aibės [Ber97].

Įvertinkime šią strategiją. Tegul p_i yra tikimybė, jog tiksliai i koordinačių yra iškraipytos abiejų z_1 ir z_2 klaidų vektorių. Tuomet

$$p_i = Pr(|\{l: z_1(l) = 1\} \cap \{l: z_2(l) = 1\}| = i) = \frac{\binom{t}{i}\binom{k}{t-i}}{\binom{n}{t}},$$

nes, tarkime, klaidų vektorius z_2 privalo rinktis i klaidų koordinačių iš tų t z_1 klaidų vektoriaus bei likusių $t - i$, neiškraipytų klaidų vektoriaus z_1, sumoje gaunant $\binom{n}{t}$ galimus klaidų vektorius, kur i yra skaičius koordinačių, kuriose padarytos klaidos. Tad numatoma L_1 aibės galia yra $E(|L_1|) = \sum_{i=0}^{t}(2t - 2i)p_i$, kadangi kiekvienas l, kuriam galioja lygybė $z_1(l) = 1 = z_2(l)$, sumažina $|L_1|$ dviem [Ber97].

2.1.4. Susijusio pranešimo atakos sąlygos

Aprašius žinutės persiuntimo ataką, galima ją apibendrinti. Įsivaizduokime, jog turime $x_1 = u_1 SGP + z_1$ ir $x_2 = u_2 SGP + z_2$ užšifruotus pranešimus su sąlygomis, jog $u_1 \neq u_2$, $z_1 \neq z_2$, ir kriptoanalitikas žino tiesinį ryšį tarp pranešimų, pavyzdžiui, $u_1 + u_2$ (tai vadinsime susijusio pranešimo sąlyga), kur x_1 ir x_2 yra užšifruoti pranešimai, u_1 ir u_2 – atvirieji pranešimo tekstai, *SGP* – viešasis raktas, o z_1 ir z_2 – klaidų vektoriai. Šiuo atveju kriptoanalitikas gali išgauti u iš x įvykdydamas vieną skaičiavimą ir po to atlikdamas ataką, nurodytą 2.1.3 skyriuje. Aptarkime skaičiavimą iki nurodytos atakos [Ber97].

Sudėjus abu užšifruotus vektorius, gauname $x_1 + x_2 = u_1 SGP + u_2 SGP + z_1 + z_2$, kur x_1 ir x_2 yra užšifruoti pranešimai, u_1 ir u_2 – atvirieji pranešimo tekstai, *SGP* – viešasis raktas, o z_1 ir z_2 – klaidų vektoriai. Svarbu pastebėti, jog $u_1 SGP + u_2 SGP = (u_1 + u_2)SGP$ – tai reikšmė, kurią kriptoanalitikas gali apskaičiuoti pasinaudodamas susijusio pranešimo sąlyga iš jau žinomo sąryšio tarp pranešimų bei viešojo rakto. Kriptoanalitikas apskaičiuoja $x_1 + x_2 + (u_1 + u_2)SGP = z_1 + z_2$ ir toliau vykdo ataką, nurodytą 2.1.3 skyriuje panaudodamas $(x_1 + x_2 + (u_1 + u_2)SGP)$ vietoje $(x_1 + x_2)$. Svarbu paminėti, jog 2.1.3 skyriuje nurodyta ataka yra atskiras susijusio pranešimo atakos atvejis, kai $u_1 + u_2 = 0$ [Ber97].

2.1.5. Konkretus atakos pavyzdys

Pateikime konkretų pranešimo persiuntimo atakos pavyzdį. Prieš tai prisiminkime McEliece kriptosistemos parametrus bei priskirkime jiems vertes: $n = 1983$ – Goppa kodo ilgis, $k = 1473$ – Goppa kodo dimensija, o $t = 46$ – klaidų vektoriaus svoris. Tarkime, jog aibės $|L_0| = 1895$, kur $|L_0|$ yra L_0 aibės galia. Kai abu klaidų vektoriai iškraipo nesutampančias to paties užkoduoto vektoriaus koordinates (Hemingo atstumas tarp vektorių yra $2t$), neiškraipytų koordinačių skaičius užkoduotame vektoriuje yra $|L_0| = 1983 - 2 \times 46 = 1891$. Tai žinodami galime daryti išvadą, jog jei $|L_0| > 1891$, tuomet dalis klaidų vektorių koordinačių sutampa. Kadangi lyginame 2 nesaugiu kanalu perimtus užkoduotus bei iškraipytus vektorius, kaskart, kai tarp jų sutaps viena koordinatė, jie L_0 aibę papildys 2 elementais. Remiantis šia logika bei pasinaudojus tuo, jog $|L_0| = 1895$, apskaičiuojame, kad iš viso klaidų vektorių buvo iškraipytos $(1895 - 1891) \div 2 = 2$ sutampančios koordinatės abiejuose užkoduotuose vektoriuose. Kiek teorinių spėjimų reikėtų, kad būtų pasirinktos Goppa kodo dimensiją atitinkančios 1473 klaidų vektoriaus neiškraipytos užkoduoto vektoriaus koordinatės? Šiam skaičiui gauti pasinaudosime pirmąja formule.

Tikimybė iš pasirinkto užkoduoto vektoriaus atspėti 1473 klaidų vektoriaus neiškraipytas koordinates iš L_0 aibės yra apibrėžta pirmoje formulėje.

1 formulė. Teorinis spėjimų skaičius, reikalingas pasirinkti 1473 klaidų vektoriaus neiškraipytoms koordinatėms

$1 \div \frac{\binom{|L_0|-b}{1473}}{\binom{|L_0|}{1473}}$, kur b yra klaidų vektoriaus iškraipytų koordinačių užkoduotame vektoriuje skaičius, o$|L_0|$ - L_0 aibės galia.

Dalmeniu yra išreikšta įvykio tikimybė, jog iš užkoduotojo vektoriaus bus pasirinktos 1473 klaidų vektoriaus neiškraipytos koordinatės, o dalyba iš vieneto apibrėžia, kiek vidutiniškai spėjimų reikės, kol įvykis įvyks. Įrašę pavyzdyje numatytas reikšmes, gauname tokį rezultatą: $1 \div \frac{\binom{1895-2}{1473}}{\binom{1895}{1473}} \approx 20$. Tai reikš, jog iš pasirinkto užkoduoto vektoriaus atspėti 1473 klaidų vektoriaus neiškraipytas koordinates iš L_0 aibės teoriškai mums prireiks apie 20-ies bandymų.

2.2. Mažo svorio žodžio tiesiniame kode radimo ataka

Tarkime, jog turime tiesinį dvejetainį kodą C, kurio ilgis yra n, kodo dimensija k, klaidų vektoriaus svoris t, tenkinantis lygybę $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, kur d yra mažiausias tiesinio kodo C atstumas, ir viskas, ką žinome apie jį, yra jo generuojanti matrica G. Apsibrėžkime algoritmą, kuris padės surasti svorio w žodį, priklausantį tiesiniam dvejetainiui kodui C, kuris yra artimas šio kodo mažiausiam atstumui d. Jei pranešimas x buvo sudarytas iš kodo C žodžio, kuris buvo iškraipytas klaidų vektoriaus z, kurio Hemingo svoris tenkina lygybę $w \leq t$, pasinaudojant algoritmu galima atkurti klaidų vektorių z remiantis tuo, jog jis yra vienintelis minimalaus Hemingo svorio kodo žodis, priklausantis tiesiniam kodui C. Pasinaudojus lygybe bei atliktais pastebėjimais dekoduoti vektorių, priklausantį $[n, k]$ tiesiniam dvejetainiui kodui, tereikia rasti minimalaus Hemingo svorio žodį, priklausantį $[n, k+1]$ tiesiniam dvejetainiui kodui [CS98].

Tarkime, jog $N = \{1, \dots, n\}$ yra visų tiesinio dvejetainio kodo žodžio koordinačių aibė. Kiekvienam N poaibiui I formulė $G = (V, W)_I$ žymi generuojančios matricos G skaidinį I, kur $V = (G_i)_{i \in I}$, $W = (G_j)_{j \in N \setminus I}$, o G_i yra i-tasis generuojančios matricos stulpelis. Tuo tarpu n-bitų vektoriaus x ir koordinačių poaibio I susiejimas yra žymimas $x_{|I} = (x_i)_{i \in I}$. Taip pat tarkime, jog I yra k-elementų (k skaičius atitinka tiesinio kodo dimensiją) aibės N poaibis. Poaibis I yra vadinamas tiesinio dvejetainio kodo C informacine aibe tada ir tik tada, kai $G = (Id_k, Z)_I$ yra tiesinio dvejetainio kodo C generuojanti matrica, kur Id_k yra vienetinė matrica, o Z – likę $n-k$ matricos G stulpeliai. Svarbu paminėti, kad tokio pavidalo $G = (Id_k, Z)_I$ matrica yra vadinama sistemine. Algoritmas remiasi tikimybine euristika, kuri apibendrina gerai žinomą informacinės aibės dekodavimo metodą. Tačiau, užuot tyręs atsitiktinai parinktas sistemines generuojančias

matricas kiekvienos iteracijos metu vis iš naujo atliekant Gauso eliminaciją $(k \times n)$ matricai, kiekviename žingsnyje algoritmas parenka naują informacinę aibę pakeisdamas tik vieną elementą. Jei I yra informacinė aibė ir $G = (Id_k, Z)_I$ yra atitinkama sisteminė generuojanti matrica, tuomet $I' = (I \backslash \{\lambda\}) \cup \{\mu\}$ vis tiek yra tiesinio dvejetainio kodo informacinė aibė tada ir tik tada, kai koeficientas $Z_{\lambda,\mu}$ yra lygus 1, kur λ yra atsitiktinis informacinės aibės I elementas, o μ – atsitiktinė koordinatė n iš N, kuri nepriklauso I. Šio algoritmo atveju sisteminė generuojanti matrica, susieta su I' informacine aibe, yra gaunama iš prieš tai buvusios atliekant keitimo operaciją, kuri reikalauja tik $k(n-k)/2$ dvejetainių operacijų (vietoj Gauso eliminacijos atlikimo $(k \times n)$ matricai). Pasinaudodami šia logika apibrėžkime pilną algoritmo veiksmų seką [CS98].

2.2.1. Mažo svorio žodžio tiesiniame kode radimo atakos algoritmo žingsniai

Algoritmo pradžioje yra apibrėžiamas naujas tiesinis dvejetainis kodas $[n, k+1]$, kuris yra sudaromas prie McEliece kriptosistemos viešojo rakto matricos pridedant papildomą eilutę, kuri yra nesaugiame kanale perimtas vektorius x. Tuomet yra vykdomas šis veiksmas: atsitiktinai užpildoma informacinė aibė I ir atliekama Gauso eliminacija siekiant gauti sisteminę generuojančią matricą pavidalu $(Id_k, Z)_I$. Tolimesnė algoritmo eiga yra vykdoma iteratyviai, kol bus gaunamas w skaičiaus Hemingo svorio žodis, priklausantis tiesiniam dvejetainiui kodui $[n, k+1]$:

1. Informacinė aibė I yra atsitiktinai padaloma į tokius 2 poaibius I_1 ir I_2, kad $|I_1| = \lfloor k/2 \rfloor$ ir $|I_2| = \lceil k/2 \rceil$. Z matricos eilutės yra taip pat atsitiktinai padalomos į dvi lygias dalis Z_1 ir Z_2 tokiu pat principu, kaip ir I. Taip pat atsitiktinai parenkamas σ-elementų poaibis L iš aibės $J = N \backslash I$, kur σ yra vienas iš informacinės aibės dekodavimo atakos parametrų.
2. Apskaičiuojamos visos galimos p skaičiaus iš Z_1 (atitinkamai ir Z_2) matricos eilučių sudarytos tiesinės kombinacijos. Tuomet visos gautos tiesinės kombinacijos yra išsaugojamos į pirmąją arba antrąją maišos lentelę (priklausomai nuo to, kokios matricos eilutės buvo naudojamos (Z_1 arba Z_2)) tokiu būdu - pasinaudojus tiesine kombinacija bei remiantis L aibės nariais, kurie atitinka perimto pranešimo x koordinates, yra apskaičiuojamas tiesinę kombinaciją atitinkantis raktas, gaunamas sudarius naują vektorių. Naujasis vektorius yra sudaromas pasirenkant tik tuos gautos tiesinės kombinacijos elementus, kurie atitinka indeksus, apibrėžtus aibėje L. Tuomet pasinaudojant apskaičiuotu raktu į atitinkamą maišos lentelę yra pridedama kiekviena tiesinė kombinacija.

3. Naudojantis maišos lentelėmis yra pasirenkami visi sutampantys jų raktai. Inicijavus ciklą, kurio kiekvienoje iteracijoje yra panaudojamas sutampantis raktas, yra gaunamos dvi tą raktą atitinkančios tiesinių kombinacijų aibės, kurias galime apsibrėžti kaip Λ_1 ir Λ_2 indeksais nurodydami, kuria maišos lentele buvo pasinaudota. Tuomet naudojantis Λ_1 ir Λ_2 aibių elementais apskaičiuojame visas tiesines kombinacijas, gaunamas panaudojus po vieną elementą iš aibių Λ_1 ir Λ_2 ir kiekvienos gautos tiesinės kombinacijos, kurią galime apsibrėžti kaip h, žingsnyje patikriname, ar yra patenkinama tokia lygybė: $wt(h) = w - 2p$, kur $wt(x)$ yra x vektoriaus Hemingo svoris. Jei lygybė yra patenkinama, tai reiškia, jog gavome klaidų vektorių z, kuris atitinka apskaičiuotą tiesinę kombinaciją h ir algoritmas yra sustabdomas.
4. Atsitiktinai pasirenkamos tokios $\lambda \in I$ ir $\mu \in J$ reikšmės, jog jos tenkintų sąlygą $Z_{\lambda,\mu} = 1$ ir informacinė aibė I yra atnaujinama į $(I \backslash \{\lambda\}) \cup \{\mu\}$. Tuomet yra atliekama Gauso eliminacija generuojančioje tiesinio kodo matricoje su stulpeliu, kurio indeksas yra μ. Galiausiai yra atnaujinama Z matrica. [CS98].

Gautas algoritmo rezultatas yra svorio w tiesinio kodo C žodis, atitinkantis klaidų vektorių z. Atakos algoritmo vykdymo laikui sumažinti yra parenkami atitinkami p ir σ parametrai [CS98].

2.2.2. Mažo svorio žodžio tiesiniame kode radimo atakos algoritmo darbo sąnaudos

Darant prielaidą, jog kiekviena mažo svorio žodžio tiesiniame kode radimo atakos algoritmo iteracija yra nepriklausoma, bendram iteracijų skaičiui apskaičiuoti galima pritaikyti šią formulę:

$$T_{p,\sigma} = \frac{\binom{k+1}{\frac{k+1}{2}}\binom{n-k-1}{\sigma}}{\binom{k+1-2p}{\frac{k+1}{2}-p}\binom{2p}{p}\binom{n-k-1+2p-t}{\sigma}} \cdot \frac{\binom{n}{k+1}}{\binom{n-t}{k+1-2p}\binom{t}{2p}}.$$

Svarbu paminėti, jog ši formulė yra aproksimuota bei modifikuota – šaltinyje [KI03] galimai buvo padaryta klaida. Ji buvo ištaisyta sukeitus formulės vardiklį ir skaitiklį vietomis.

Pirmojo ir antrojo algoritmo žingsnių metu reikia apskaičiuoti visas galimas p skaičiaus iš Z_1 ir Z_2 matricos eilučių sudarytas tiesines kombinacijas, kurių yra $\binom{(k+1)/2}{p}$. Pirmajam ir antrajam algoritmo žingsniams atlikti reikalingas dvejetainių operacijų skaičius yra išreikštas šia formule: $\Omega_1(p,\sigma) = p \times \sigma \times \binom{(k+1)/2}{p}$. Trečiame žingsnyje maždaug $\frac{\binom{(k+1)}{2p}^2}{2^\sigma}$ tiesinių kombinacijų iš Λ_1 ir Λ_2 aibių atitinka tą patį raktą iš maišos lentelių. Kiekvienai tiesinei kombinacijai h, gaunamai panaudojus po vieną elementą iš aibių Λ_1 ir Λ_2, yra apskaičiuojama

tokia lygybė: $wt(h) = w - 2p$. Dėl šios priežasties trečiojo algoritmo žingsnio reikalingam dvejetainių operacijų skaičiui apskaičiuoti yra naudojama ši formulė: $\Omega_2(p,\sigma) = 2(t - 2p) \times p \times \frac{\binom{(k+1)/2}{p}^2}{2^\sigma}$. Galiausiai paskutinio algoritmo žingsnio veiksmų atlikimo (atnaujinti kodo generuojančią matricą) dvejetainėms operacijoms apskaičiuoti yra skirta ši formulė: $\Omega_3(p,\sigma) = \frac{k(n-k-1)}{4}$. Bendrą dvejetainių operacijų skaičių, reikalingą įgyvendinti algoritmui, galima išreikšti antrąja formule.

2 formulė. Dvejetainių operacijų skaičius, reikalingas atlikti siekiant įvykdyti mažo svorio žodžio tiesiniame kode radimo atakos algoritmo ciklą.

$$W_{p,\sigma} \approx (\Omega_1(p,\sigma) + \Omega_2(p,\sigma) + \Omega_3(p,\sigma)) \times T_{p,\sigma}.$$

3. ATAKŲ REALIZAVIMAS

Šiame skyriuje bus apžvelgtas susijusio pranešimo, pranešimo persiuntimo bei mažo svorio žodžio tiesiniame kode radimo atakų programinis kodas, pasiekiamas *Github* repozitorijoje adresu: https://github.com/pjach/mceliece-crypto-syst-security.

3.1. Apžvalga

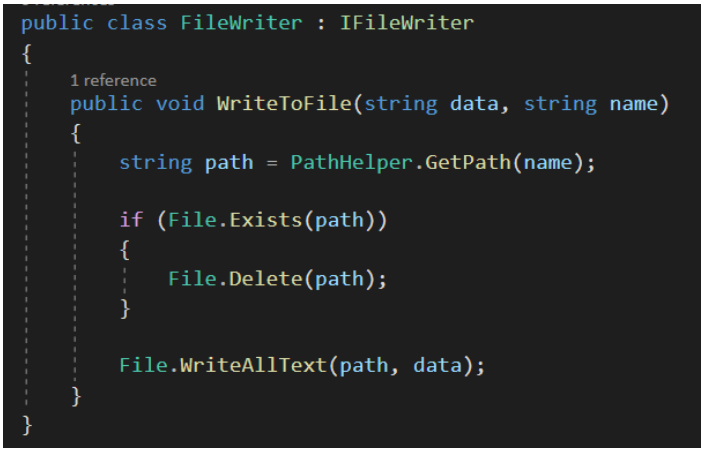
Konsolinė programa buvo parašyta *C#* objektine programavimo kalba naudojant *.NET Core* karkasą. Objektinė paradigma buvo pasirinkta dėl to, jog atakoms realizuoti reikalinga objektų (matricų, vektorių) bei metodų (atakų algoritmai) sąveika, kuria ir pasižymi nurodyta paradigma. *.NET Core* karkaso pasirinkimas grindžiamas tuo, jog šis karkasas suteikia skirtingų platformų suderinamumą, kas leidžia neprisirišti prie operacinės sistemos; taip pat šis karkasas turi didelę perspektyvą dėl *Microsoft* planuojamo jo palaikymo bei kuriamų naujesnių versijų ateityje. Konsolinės programos tipas buvo pasirinktas dėl grafinės vartotojo sąsajos nereikalingumo – programos esmė yra vykdyti numatytas atakas, kurioms nereikia vartotojo įsiterpimo.

3.2. Bibliotekos

Programiniame kode buvo panaudotos kelios atvirojo kodo bibliotekos - *Math.Net Numerics, Newtonsoft.Json* ir *Combinatorics.* Pirmoji biblioteka yra skirta atlikti veiksmus su vektoriais bei matricomis, kas yra labai aktualu vykdant atakos algoritmus. Esminis bibliotekos trūkumas yra mažas jos efektyvumas konkretiems atakos algoritmams. Visi vektorių bei matricų elementai gali įgyti tik 0 arba 1 vertes, o mažiausio dydžio kintamasis, kurį palaiko biblioteka, yra 4 baitų dydžio *Single* tipas. Nepaisant nurodyto trūkumo biblioteka buvo pasirinkta dėl to, jog geresnių bei nemokamų alternatyvų nebuvo rasta. *Newtonsoft.Json* biblioteka yra skirti dirbti su JSON objektais bei atlikti įvairias transformacijas, susijusias su šiuo formatu. Programoje ši biblioteka pasitarnavo kaip įrankis, leidžiantis serializuoti bei deserializuoti statistikos objektų sąrašą taip suteikiant galimybę manipuliuoti duomenimis ilgalaikėje atmintyje. *Combinatorics* biblioteka buvo naudojama skaičiuojant galimas kombinacijas.

3.3. Projektavimo šablonai bei principai

Programinis kodas buvo rašomas pritaikant SOLID objektiškai orientuoto programavimo projektavimo principus. Pirmasis pavyzdys antrame paveikslėlyje pateikiamas remiantis *Open-Closed* principu, teigiančiu, jog objektai arba esybės turėtų būti atviros praplėtimui, tačiau uždaros keitimui.



```
public class FileWriter : IFileWriter
{
    1 reference
    public void WriteToFile(string data, string name)
    {
        string path = PathHelper.GetPath(name);

        if (File.Exists(path))
        {
            File.Delete(path);
        }

        File.WriteAllText(path, data);
    }
}
```

2 pav. Įgyvendinama sąsaja, kuri apriboja keisti metodo parametrus, tačiau leidžia jį realizuoti

Antrasis pavyzdys yra susijęs su *Liskov Substitution* principu, kuris teigia, jog jei S yra T potipis, tuomet T tipo objektai gali būti pakeisti S tipo objektais. Principas įgyvendintas per sąsają trečiame paveikslėlyje.

```
IFileReader fileReader = new FileReader();

string relatedAttackData = fileReader.ReadFromFile(ConfigurationManager
                .AppSettings["related_statistics"]);
```

3 pav. *fileReader* objektas yra *IFileReader* tipo – kode galima pasiekti tik sąsajai priklausančius metodus

Programiniame kode taip pat buvo taikytas *Singleton* projektavimo šablonas, kuris užtikrina, jog visos programos metu bus sukurtas bei perpanaudotas tik vienas objektas. Pavyzdys ketvirtame paveikslėlyje.

```
public sealed class Randomizer : RandomNumberGenerator
{
    private readonly RandomNumberGenerator rng = new RNGCryptoServiceProvider();
    private static Randomizer instance = null;

    0 references
    static Randomizer()
    {
    }

    1 reference
    private Randomizer()
    {
    }

    1 reference
    public static Randomizer Instance
    {
        get
        {
            if (instance == null)
            {
                instance = new Randomizer();
            }
            return instance;
        }
    }
```

4 pav. *Singleton* projektavimo šablono pritaikymas atsitiktinių skaičių generavimui

3.4. Susijusio pranešimo ir pranešimo persiuntimo atakų įgyvendinimas

Šiame poskyryje apžvelgsime susijusio pranešimo bei pranešimo persiuntimo atakų algoritmo įgyvendinimą bei atvirojo pranešimo gavimą. Programa pradeda darbą nuo duomenų failo nuskaitymo – McEliece kriptografinės sistemos viešojo rakto matricos. Šie duomenys yra gaunami pasinaudojus programa [Sot19], kuri realizuoja McEliece kriptosistemą remiantis Goppa kodų šeima. Nuskaičius duomenis į *String* tipo kintamąjį, jis yra konvertuojamas į matricos atitikmens objektą. Kitas žingsnis yra paruošti duomenis atakai: priklausomai nuo jos, yra atsitiktinai sugeneruojamas pranešimo (arba pranešimų) vektorius bei sudauginamas su viešuoju raktu imituojant realų McEliece kriptosistemos užšifravimo algoritmą. Jis užbaigiamas atsitiktinai sugeneravus klaidų vektorių, kuris yra pridedamas prie atvirojo pranešimo bei viešojo rakto sandaugos. Užšifravimo algoritmas yra vykdomas „paruošiamajame“ metode, kuris, baigdamas darbą, iškviečia ataką vykdantį metodą. Ataką vykdantis metodas sukuria 2.1.3. bei 2.1.4. poskyriuose nurodytas aibes L_1 bei L_0 inicijuodamas du sąrašo tipo objektus. Tuomet metodas pradeda vykdyti begalinį ciklą, kuris yra baigiamas tik tuomet, kai ataka pavyksta, t.y. gaunamas vektorius, kurio Hemingo atstumas iki užkoduoto vektoriaus yra lygus t, kur t yra McEliece kriptosistemos parametras, nurodantis klaidų vektoriaus svorį. Ciklas prasideda atsitiktinių L_0 aibėje saugomų k skaičiaus (tiesinio kodo dimensija) koordinačių pasirinkimu. Pasinaudojant atsitiktinai pasirinktomis koordinatėmis programa sukuria matricos objektą, sudarytą iš šių koordinačių, atitinkančių transponuotus stulpelius, paskutiniajam papildomam matricos stulpeliui priskiriant užkoduoto vektoriaus reikšmes, atitinkančias programos pasirinktas atsitiktines koordinates. Kitas algoritmo žingsnis yra vykdyti Gauso eliminaciją, kuri parodo, ar matrica, kaip lygčių sistema, turi sprendinį, kuris yra kandidatas į atvirąjį pranešimą. Jei Gauso eliminacijos metu pavyko matricą suvesti į vienetinį pavidalą ir gauti sprendinį, *bool* tipo kintamasis įgauna *true* reikšmę (nesėkmės atveju reikšmė nustatoma į *false*), kurią kitame žingsnyje algoritmas patikrina. Nustačius, jog reikšmė yra *true*, programa paskutinįjį matricos stulpelį, kuris yra kandidatas į sprendinį, sudaugina su viešuoju raktu gaudama vektorių. Paskutinis ciklo žingsnis yra patikrinti, ar Hemingo atstumas tarp gautojo vektoriaus ir užkoduoto vektoriaus yra lygus klaidų vektoriaus svoriui t. Jei tai tiesa, programa sustabdo ciklą ir grąžina atsakymą, reiškiantį sėkmingą ataką. Priešingu atveju pradedama nauja iteracija.

3.5. Mažo svorio žodžio tiesiniame kode radimo atakos įgyvendinimas

Šiame poskyryje apžvelgsime mažo svorio žodžio tiesiniame kode radimo atakos įgyvendinimo algoritmą bei atvirojo pranešimo gavimą iš užkoduoto vektoriaus, kuris nebėra iškraipytas klaidų vektoriaus t. Programa vykdo lygiai tokius pačius žingsnius, kaip ir susijusio

pranešimo (ir pranešimo persiuntimo) atakos paruošiamasis metodas iki tol, kol yra gaunamas užkoduotas vektorius (paruošiamasis metodas imituoja realų McEliece kriptosistemos užšifravimą). Tuomet yra pradedamas vykdyti 2.2.1. poskyryje apibrėžtas mažo svorio žodžio tiesiniame kode radimo atakos algoritmas, kuris prasideda sukuriant tiesinį dvejetainį kodą $[n, k+1]$ prie viešojo rakto matricos pridedant papildomą paskutinę eilutę (užkoduotas vektorius x) ir gaunama modifikuota viešojo rakto matrica. Toliau yra sukuriamas informacinės aibės I atitikmens sąrašas atsitiktinai pasirenkant k skaičių atsitiktinių modifikuotos viešojo rakto matricos stulpelių indeksų. Tuomet, naudojantis informacinės aibės sąrašu, atitinkančiu informacinę aibę I, yra vykdoma Gauso eliminacija modifikuotoje viešojo rakto matricoje siekiant gauti sisteminę generuojančią matricą pavidalu $(Id_k, Z)_I$, kur Id_k yra k stulpelių skaičiaus vienetinė matrica, o Z yra likę $n-k$ stulpeliai. Jei kuris nors pasirinktas matricos stulpelis yra tiesiškai priklausomas kitų stulpelių, atitinkančių informacinės aibės sąrašo elementus, požiūriu, tuomet programa vietoj to stulpelio atsitiktinai pasirenka kitą (dar nepanaudotą) stulpelį, atnaujina informacinės aibės sąrašą ir toliau bando vykdyti Gauso eliminaciją. Kitas žingsnis įvykdžius Gauso eliminaciją yra J aibės sąrašo sukūrimas siekiant suformuoti Z matricą. Programa perbėga visus modifikuotos viešojo rakto matricos stulpelių indeksus ir suformuoja naują sąrašą tik iš tų stulpelių indeksų, kurių nėra informacinės aibės sąraše. Tuomet programa pradeda begalinį ciklą, kurio žingsniai yra apibrėžti bei sunumeruoti 2.2.1. poskyryje. Ciklo pradžioje programa vykdo pirmąjį ciklo žingsnį apskaičiuodama bei sukurdama L aibės sąrašą atsitiktinai pasirinkdama σ skaičių indeksų iš J aibės sąrašo. Tuomet programa sukuria naują objektą, kuriame yra saugoma suformuota Z matrica (ji gaunama sudarant modifikuotą matricą iš viešojo rakto matricos stulpelių, kurių indeksai yra saugomi J aibės sąraše). Tęsiant pirmąjį ciklo žingsnį informacinė aibė yra padaloma į du lygius (jei įmanoma) bei nepersidengiančius poaibius. Ši dalis yra realizuota taip: programa atsitiktinai sugeneruoja pusę informacinės aibės indeksų. Tuomet yra paleidžiamas pirmasis ciklas, kuris pagal tuos indeksus suformuoja pirmąjį informacinės aibės I poaibio sąrašą I_1. To paties ciklo metu yra sukuriamas ir Z matricos eilučių pirmojo poaibio Z_1 sąrašas (vietoj atskiros matricos taupant atmintį). Jis yra užpildomas tokiu būdu: pasinaudojant informacinės aibės I poaibio sąrašo I_1 elementais, atitinkančiais modifikuotos viešojo raktos matricos vienetinius stulpelius, yra pasirenkami tokie stulpelių (vektorių) elementų indeksai, kuriuose stulpelio elemento reikšmė yra „1“. Antrojo ciklo metu yra sukuriami Z_2 bei I_2 poaibių sąrašai su likusiomis informacinės aibės sąrašo I indeksų vertėmis taikant tą pačią logiką. Toliau programa vykdo antrąjį 2.2.1. poskyryje apibrėžtą ciklo žingsnį, kurio metu sukuria dvi maišos lenteles bei randa visas galimas p skaičiaus kombinacijas Z_1 ir Z_2 sąrašams atskirai. Tuomet programa inicijuoja po atskirą ciklą ir perbėga visas įmanomas tiesines kombinacijas iš Z_1 ir Z_2 poaibių sąrašų. Logika cikluose yra identiška: remiantis kombinacijomis yra sudedamos atitinkamos Z

matricos eilutės ir sukuriamas naujas objektas, kuriame yra išsaugomas gautas vektorius bei sudėtųjų eilučių indeksai. Tuomet šis objektas yra išsaugomas į maišos lenteles su tokiu raktu, kuris atitinka gauto vektoriaus indeksus, kurie nurodyti L aibės sąraše. Toliau programa vykdo 2.2.1. poskyryje apibrėžtą trečiąjį ciklo žingsnį, kurio metu yra vykdomas toks ciklas: pagal identiškus maišos funkcijos raktus yra gaunami du jų reikšmių sąrašai (objektas, sudarytas iš susumuotų vektorių pagal Z matricos eilutes bei panaudotų eilučių indeksų sąrašą), kurie algortime įvardinti kaip Λ_1 ir Λ_2. Šių objektų sumos vektoriai yra sudedami vienas su kitu (suma tik tarp skirtingų sąrašų Λ_1 ir Λ_2 vektorių) ir tikrinama, ar gauto sumos vektoriaus Hemingo svoris yra tas, kurio ieškome ($t - 2p$, kur t yra klaidų vektoriaus svoris, o p yra vienas iš atakos parametrų). Jei sąlyga nėra niekur patenkinama, tuomet programa pradeda vykdyti ketvirtąjį 2.2.1. poskyryje apibrėžtą ciklo žingsnį - atsitiktinai pasirenka nenulinį stulpelį iš modifikuotos viešojo rakto matricos ir pakeičia atsitiktinį informacinės aibės elementą naujuoju indeksu. Programa indeksus taip pat atnaujina ir J aibės sąraše. Tuomet programa įvykdo Gauso eliminaciją tik naujam indeksui modifikuotoje viešojo rakto matricoje (atnaujinant visą matricą) ir ciklą pradeda iš naujo. Jei gautas sumos vektoriaus Hemingo svoris patenkina sąlygą, tuomet, remiantis J eilučių indeksų sąrašu, yra sudedami visi vektoriai iš viešojo rakto matricos. Gautasis vektorius yra atspėtasis klaidų vektorius z. Svarbu paminėti, jog suradus klaidų vektorių z 2.2.1. poskyryje apibrėžtas algoritmas yra baigtas, tačiau dar liko surasti atvirąjį pranešimą. Turėdama atspėtąjį klaidų vektorių programa jį prideda prie gauto užkoduoto vektoriaus. Gautas sumos vektorius jau yra neiškraipytas užkoduotas vektorius, iš kurio norint išgauti atvirąjį pranešimą belieka išspręsti matricinę lygtį. Ji yra sprendžiama tokiu būdu: iš informacinės aibės yra pašalinamas pirmasis elementas (tai yra daroma dėl to, jog atakos pradžioje yra sukuriamas naujas tiesinis dvejetainis kodas $[n, k + 1]$, kurio dimensija yra vienetu didesnė nei kriptosistemos tiesinio kodo dimensija k). Tuomet viešojo rakto matricai yra vykdomas tas pats Gauso eliminacijos algoritmas, kaip ir mažo svorio žodžio tiesiniame kode radimo atakos algoritmo pradžioje. Pasinaudojus atnaujintais informacinės aibės elementais (tiesiškai nepriklausomais viešojo rakto matricos stulpeliais) yra sukonstruojama nauja matrica, prie kurios paskutinio stulpelio yra pridedamas papildomas stulpelis – naujas vektorius, gautas panaudojus užkoduoto, tačiau neiškraipyto vektoriaus elementus tuose indeksuose, kurie yra apibrėžti informacinės aibės sąraše. Galiausiai paskutinį kartą įvykdžius Gauso eliminaciją yra išsprendžiama matricinė lygtis, kurios atsakymas yra paskutinis matricos stulpelis. Šis stulpelis yra atvirasis pranešimas.

4. ATAKŲ REZULTATAI

Šis skyrius skirtas apžvelgti eksperimentus bei gautus statistinius duomenis, atlikti jų analizę, paaiškinti anomalijas.

4.1. Susijusio pranešimo bei pranešimo persiuntimo atakų rezultatai

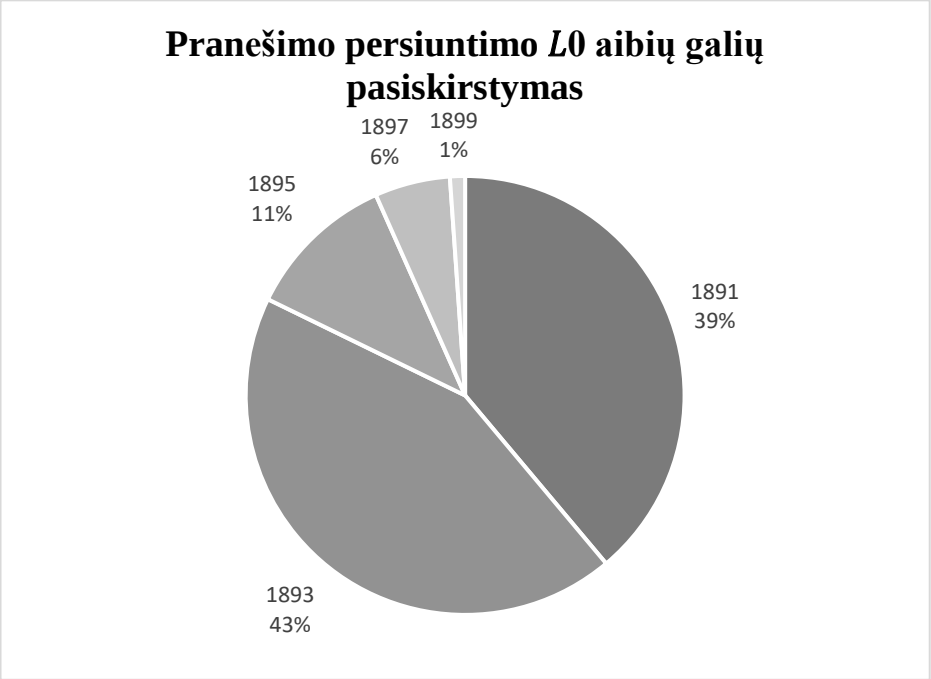
Programa statistiką renka atakos vykdymo metu. Susijusio pranešimo bei pranešimo persiuntimo atakos buvo įvykdytos po 100 kartų su tokiais McEliece kriptosistemos rekomenduojamais parametrais ([NMB+12]): $n = 1983$, $t = 46$, $k = 1473$, kur n yra Goppa kodo ilgis, t – klaidų vektoriaus Hemingo svoris, o k – Goppa kodo dimensija. Renkami statistiniai duomenys kiekvienam nurodytų atakų bandymui:

1. Bendras iteracijų skaičius;
2. Sėkmingų Gauso eliminacijų įvykdymo skaičius;
3. Bendras atakos vykdymo laikas sekundėmis;
4. L_0 bei L_1 aibių galios;
5. Aibė, kurios elementai yra kiekvienos atakos iteracijos spėjimų skaičiai, kurie buvo reikalingi pasirinkti klaidų vektoriaus neiškraipytus stulpelius.

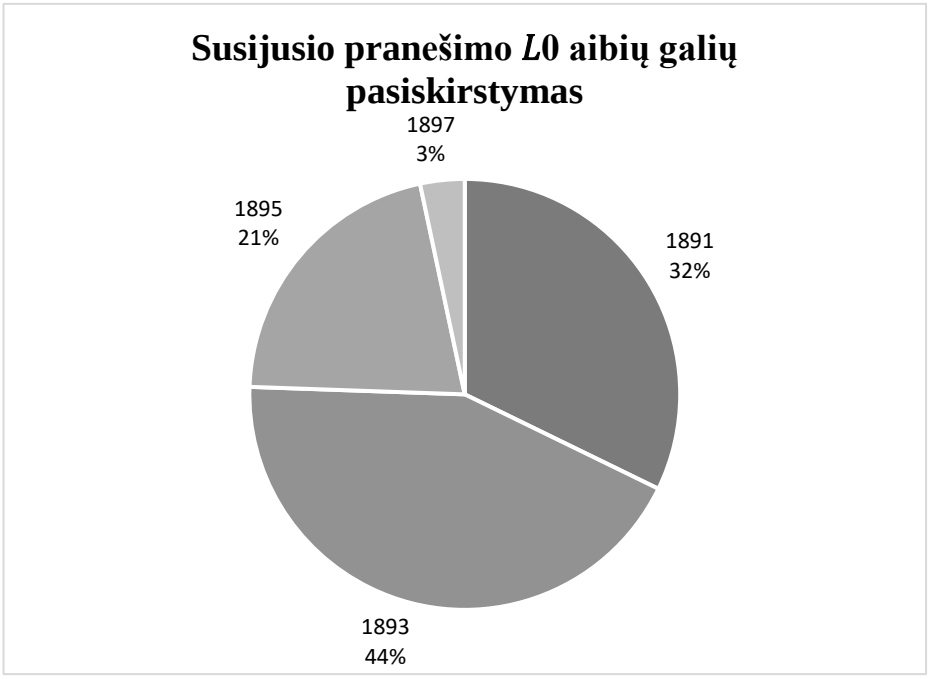
Kompiuterio, panaudoto atakoms vykdyti, techninės charakteristikos:

1. Procesorius – 2 fizinių (ir papildomų 2 virtualių) branduolių *Intel Core i5 7200U* su varijuojančiu taktiniu dažniu, galinčiu pasiekti 3.1 GHz bei turinčiu 3 megabaitus *cache* tipo atminties.
2. Operatyvioji atmintis – 8 gigabaitų LPDDR3 tipo atmintis.
3. Vaizdo plokštė – integruota *Intel HD 620* grafikos posistemė, galinti panaudoti iki 4 gigabaitų bendrinamos operatyviosios atminties.
4. Talpykla – SSD tipo 256 gigabaitų pastovioji atmintis.

Siekiant suvokti bendrų atakų vykdymų laikų vidurkių skirtumus tarp atakų bei tai lėmusias priežastis, 5-6 paveikslėliuose vaizduojami L_0 aibių galių skirstiniai kiekvienai atakai. Kuo didesnė L_0 aibės galia, tuo labiau didėja teorinis spėjimų skaičius, reikalingas pasirinkti klaidų vektoriaus neiškraipytiems stulpeliams, ilginantis pačią ataką. Svarbu paminėti, jog siekiant išlaikyti kuo mažesnį statistikos tarp abiejų atakų nuokrypį, buvo atliktas statistikos imties aritmetinio vidurkio 5% kirpimas. Tai reiškia, jog abiejų atakų imtys buvo išrikiuotos didėjančia tvarka pagal užtrukusį laiką atlikti vieną atakos bandymą ir 5% (po 5 imties vienetus) trumpiausių ir ilgiausių atakos bandymų buvo pašalinti iš statistinių imčių.



5 pav. Pranešimo persiuntimo atakos L_0 aibių galių skritulinė diagrama



6 pav. Susijusio pranešimo atakos L_0 aibių galių skritulinė diagrama

Remiantis 5 - 6 paveikslėliais galima pastebėti, jog pranešimo persiuntimo atakos vykdymo metu L_0 galios 1893 ir 1891 pasiskirstė beveik vienodai kartu sudarydamos 82% visos statistikos imties, kuomet susijusio pranešimo atakos vykdymo metu 1891 ir 1893 L_0 galios skyrėsi apie 25% kartu sudarydamos 76 % bendros statistikos imties.

Bendroji atakų statistika pateikiama pirmoje lentelėje, kurioje vaizduojami rinktos statistikos vidurkiai kiekvienai atakos statistikos imčiai atskirai bei kartu.

1 lentelė. Bendroji atakų statistika

Ataka	**Statistinis duomuo**	**Vertė**	**Bendras verčių vidurkis**
Praneš. persiun.	Vidutinis iteracijų skaičius	20,3	22,7
Susij. praneš.		25	
Praneš. persiun.	Vidutinis sėkmingų Gauso eliminacijų skaičius	5,8	6,6
Susij. praneš.		7,3	
Praneš. persiun.	Vidutinis atakos vykdymo laikas sekundėmis	2747,8	3093,6
Susij. praneš.		3439,3	
Praneš. persiun.	Vidutinė L_0 aibės galia	1892,7	1892,8
Susij. praneš.		1892,9	
Praneš. persiun.	Vidutinis spėjimų skaičius, reikalingas pasirinkti klaidų vektoriaus neiškraipytas koordinates	5,9	7,2
Susij. praneš.		8,5	

Pirmoje lentelėje matyti skirtumai tarp susijusio pranešimo bei pranešimo persiuntimo atakų statistinių duomenų, kurie nėra dideli. Šiuos skirtumus lemia atsitiktinumo faktorius. Visų pirma, kiekvienos atakos bandyme yra sugeneruojami atsitiktiniai klaidų vektoriai, nuo kurių visiškai priklauso L_0 aibės galia. Remiantis pirmąja formule galima daryti išvadą, jog kuo didesnė L_0 aibės galia, tuo daugiau reikia spėjimų, reikalingų pasirinkti klaidų vektoriaus neiškraipytas koordinates. Taip pat svarbu paminėti, jog kiekvienas toks spėjimas irgi yra atsitiktinis. Besiremiant pirmos lentelės duomenimis galima pastebėti, jog viena iteracija vidutiniškai užtrunka šiek tiek daugiau,

nei 2 minutės. Atlikus laiko išmatavimus buvo pastebėta, jog didžiausią laiko dalį atakos algoritme užima Gauso eliminacijos vykdymas (šiek tiek daugiau nei 2 minutes). Taip pat pirmoje lentelėje matyti, jog iš bendrų vidutiniškai 23 iteracijų apie 7 atvejais Gauso eliminacija buvo sėkminga, o likusiais 16 – nesėkminga, nes parinkti stulpeliai buvo tiesiškai priklausomi. Vidutiniškai pirmuosiais 6 atvejais, kai Gauso eliminacija buvo sėkminga, lygčių sistemos sprendinys vis tiek netiko, nes pasirinktose pozicijose būdavo klaidų pozicijų, ir vidutiniškai tikdavo tik 7-tąjį kartą. Toliau analizuojant pirmąją lentelę bei pasinaudojant pirmąja formule $\left(1 \div \frac{\binom{|L_0|-b}{1473}}{\binom{|L_0|}{1473}}\right)$ galima patikrinti, ar teorinis spėjimų skaičius, reikalingas pasirinkti klaidų vektoriaus neiškraipytas koordinates, atitinka praktinį. Remiantis pirmąja lentele suapvalinta vidutinė L_0 aibės galia yra 1893, o b galime apskaičiuoti vadovaujantis 2.1.5. poskyryje aprašytomis įžvalgomis: $\frac{1893-(1983-2t)}{2} \Rightarrow \frac{1893-(1983-92)}{2} = 1$. Įstačius reikšmes bei atlikus veiksmus yra gaunamas neatitikmuo: $1 \div \frac{\binom{1892}{1473}}{\binom{1893}{1473}} \approx 4{,}52 \neq 7{,}2$. Šiam neatitikimui paaiškinti yra kelios priežastys. Viena iš jų – statistikos imtis. 180 bendrų atakos bandymų yra sąlyginai mažas skaičius. Antrajai priežasčiai paaiškinti reikia detaliau pasigilinti į L_0 aibių galių pasiskirstymus bei pritaikyti pirmąją formulę kiekvienai aibių galiai atskirai.

2 lentelė. Teorinių bei praktinių vidutinių spėjimų, reikalingų pasirinkti klaidų vektoriaus neiškraipytas koordinates, skaičių palyginimas

Ataka	L_0 aibės galia	Eksperimentų su 2 stulpelyje pateikta L_0 aibės galia skaičius	Praktinis vidutinis atakos spėjimų skaičius	Atakų praktinių spėjimų vidurkis	Teorinis vidutinis spėjimų vidurkis
Praneš. persiun.	1891	35	1	1	1
Susij. praneš.		29	1		
Praneš. persiun.	1893	39	4,27	4,14	4,52
Susij. praneš.		39	4		
Praneš. persiun.	1895	10	19,79	21,56	20,4
Susij. praneš.		19	23,36		
Praneš. persiun.	1897	5	69,43	91,32	90,9
Susij. praneš.		3	113,2		
Praneš. persiun.	1899	1	21	21	500

Ataka	L_0 aibės galia	Eksperimentų su 2 stulpelyje pateikta L_0 aibės galia skaičius	Praktinis vidutinis atakos spėjimų skaičius	Atakų praktinių spėjimų vidurkis	Teorinis vidutinis spėjimų vidurkis
Susij. praneš.		-	-		

Remiantis antrosios lentelės duomenimis, vidutinis abiejų atakų spėjimų, reikalingų pasirinkti klaidų vektoriaus neiškraipytas koordinates, skaičius konkrečiai L_0 aibės galiai yra pakankamai arti teorinio vidurkio, atsižvelgiant į aibės pasikartojimų skaičių. Kaip ir minėta pirmosios lentelės apžvalgoje, didėjant L_0 aibės galiai išauga spėjimų skaičius, reikalingas atspėti klaidų vektoriaus neiškraipytas koordinates. Tęsiant 1 lentelės praktinių vidutinių spėjimų skaičiaus neatitikimą L_0 aibės galiai, antros lentelės duomenys rodo, jog galiai išaugus vos 2 (paskutinės 4 eilutės antroje lentelėje), teorinių spėjimų skaičius išauga drastiškai taip darydamas didelę įtaką (ženkliai didindamas) bendram visų galių vidurkių bei visų atakų spėjimų skaičiaus sąryšiui.

4.2. Mažo svorio žodžio tiesiniame kode radimo atakos rezultatai

Kaip ir susijusio pranešimo bei pranešimo persiuntimo atakų statistikai rinkti, mažo svorio žodžio tiesiniame kode radimo atakos vykdymo metu buvo renkama statistika. Ataka buvo vykdoma po 1000 kartų su skirtingais McEliece kriptografinės sistemos bei mažo svorio žodžio tiesiniame kode radimo atakos parametrais: $n = 2^m$ bei poromis $(m, t,\ p, \sigma)$: $(5, 4, 1, 2)$, $(6, 5, 1, 4)$, $(7, 8, 1, 6)$ ir $(8, 13, 1, 7)$. Taip pat buvo įvykdyta viena ataka su tokiais kriptosistemos ir atakos parametrais: $n = 2^m$, bei pora (m, t, p, σ) - $(9, 12, 2, 16)$. Tokie McEliece kriptografinės sistemos bei atakos parametrai buvo pasirinkti ne atsitiktinai.

4.1.1. Mažo svorio žodžio tiesiniame kode radimo atakos efektyviausių parametrų apskaičiavimas

Siekiant įsitikinti McEliece kriptografinės sistemos saugumu bei numatyti, kiek laiko užtruktų įvykdyti mažo svorio žodžio tiesiniame kode radimo ataką ne tik naudotam, tačiau ir galingiausiam pasauly superkompiuteriui, visų pirma reikia apsibrėžti saugiausius kriptografinės sistemos parametrus kiekvienam naudotam m skaičiui. Šių parametrų pasirinkimo kelias buvo toks: pasinaudojus antrąja formule su kiekvienomis m, t reikšmėmis surasti optimalius atakos parametrus p ir σ. Visų pirma, pasirinkus konkrečią m reikšmę, buvo pasirinktos visos galimos t reikšmės. Tuomet kiekvienai galimai t reikšmei buvo pritaikoma antroji formulė panaudojant skirtingus atakos parametrus: $p = 1,2,\ \sigma = 1,2,3, \ldots 20$. Kiekvienai t reikšmei buvo ieškomos

tokios p ir σ reikšmės, jog atakai įvykdyti reikalingas dvejetainių operacijų skaičius būtų mažiausias. Tuomet, turint mažiausias sąnaudas kiekvienai t reikšmei, pasirinkti tokį skaičių t, jog dvejetainių operacijų skaičius būtų didžiausias. Pritaikius tokią logiką yra pasirenkami efektyviausi atakos parametrai daugiausiai dvejetainių operacijų reikalaujantiems McEliece kriptosistemos parametrams. Tokie kriptosistemos parametrai būtų rekomenduojami kaip saugiausi, o jų saugumą būtų galima įvertinti panaudojus efektyviausius atakos parametrus. Iš viso buvo sukurta ir panaudota 160 lentelių, įvertinančių parametrus naudojantis *Microsoft Excel* programa. Vienos iš lentelių pavyzdys yra nurodomas septintame paveikslėlyje.

p = 1	SIGMA	W
m=5	1	431.52
n =32	2	477.6226
k = 2	3	560.6874
t = 6	4	680.5394
MIN:	5	842.1462
431.52	6	1055.348
	7	1335.596
	8	1705.734
	9	2199.14
	10	2864.91
	11	3776.378
	12	5045.348
	13	6846.474
	14	9460.408
	15	13353.09
	16	19327.95
	17	28833.44
	18	44623.13
	19	72279.63
	20	124062

7 paveikslėlis. Reikalingų dvejetainių operacijų, apskaičiuojamų remiantis antrąja formule, skaičius konkretiems McEliece kriptografinės sistemos bei mažo svorio žodžio tiesiniame kode radimo atakos parametrams rasti.

Konkrečios atakos parametrų ribos buvo pasirinktos dėl to, jog, kaip galima pastebėti, σ reikšmės artėjant link 20 pastoviai didėja, todėl nėra prasmės skaičiuoti jų toliau. Tuo tarpu reikšmės su p, kai $p = 1$, visuomet pateikdavo geresnius rezultatus. Tik tuomet, kai buvo apskaičiuojamas dvejetainių operacijų skaičius McEliece kriptosistemos parametrams, kuomet $m = 9$, buvo pastebėta, jog didesnėms t reikšmėms efektyviau yra naudoti $p = 2$. Svarbu paminėti

ir tai, jog dvejetainių operacijų skaičiaus paieškos buvo pilnai atliktos (su visais galimais t) tik tokiems McEliece kriptografinės sistemos parametrams, kai $m = 5,6,7,8$. Pilniems skaičiavimams atlikti, kai $m = 9$, nebuvo prasmės, nes tam būtų prireikę apskaičiuoti 112 lentelių, iš kurių gavus saugiausius parametrus ir panaudojus atakai, ji būtų užtrukusi pernelyg ilgai. Dėl šios priežasties buvo apskaičiuotos tik 28 lentelės, iš kurių gauti parametrai buvo panaudoti vykdant ataką tik vieną kartą. Šis bandymas užtruko apie 15 valandų, nors pasirinkti parametrai nebuvo saugiausi. Aštuntame paveikslėlyje yra pateikiami visi apibendrinti skaičiavimai.

W	**M,N**				
T	5,32	6,64	7,128	8,256	9,512
2	227.3333333	485.3333333	1051.034483	2292.892562	5011.096774
3	594.2536476	2053.153846	7705.154458	29351.11111	114493.6489
4	**661.0294118**	3627.181287	21344.53782	62958.47701	213693.9131
5	604.7818182	**4581.153043**	46780.75095	514300.4441	6098298.188
6	431.52	4527.293793	75038.89463	1420986.32	**29532392.74**
7		3726.158937	99868.64257	3220212.705	117776341.9
8		2682.251363	**111220.3309**	6178570.183	397284777.4
9		1757.744215	108969.722	10270390.27	1165172360
10		1206.961679	92252.10619	15050571.14	3008162342
11			71500.92289	19712012.66	6963501611
12			49392.93889	23330339.04	14492786178
13			32316.77208	**25181992.01**	26708637180
14			19396.76335	24979429.57	44922230822
15			11359.01927	22922873.67	71108195063
16			6267.701242	19572915.79	
17			3840.928648	15630242	
18			2773.899818	11727549.56	
19				8302851.383	
20				5527857.525	
21				3497824.818	
22				2112040.506	
23				1221784.359	
24				680062.8825	
25				366039.1114	
26				191441.7667	
27				96598.65648	
28				48355.01686	
29				24527.85627	
30				12688.74663	
31				7512.039203	

8 paveikslėlis. Mažiausios reikalingų dvejetainių operacijų, apskaičiuojamų antrąja formule, reikšmės skirtingiems McEliece kriptosistemos bei mažo svoro žodžio tiesiniame kode radimo atakos parametrams.

Oranžine spalva pažymėtos reikšmės atspindi didžiausias reikšmes kiekviename stulpelyje, pavyzdžiui, kai $m = 5$, didžiausias dvejetainių operacijų skaičius yra parametrui $t = 4$ ir t.t. Raudona spalva pažymėtas skaičius rodo, kokie McEliece kriptografinės sistemos parametrai, kai $m = 9$, jau reikalauja didesnio skaičiaus dvejetainių operacijų, reikalingų atakai įvykdyti, nei kriptosistemos parametrai, kai $m = 8$. Tuo tarpu mėlyna spalva pavaizduota reikšmė parodo, kokie kriptografinės sistemos parametrai buvo panaudoti vykdant ataką vienintelį kartą, užtrukusį apie 15 valandų.

4.1.2. Mažo svorio žodžio tiesiniame kode radimo atakos eksperimentai

Pasinaudojus 8 paveikslėlyje pateikta informacija apie kriptosistemos parametrus bei tuos parametrus atitinkančiomis konkrečiomis lentelėmis (7 paveikslėlis kaip pavyzdys) buvo įvykdytos atakos ir gauti rezultatai, kurie pateikiami 3-ioje lentelėje.

3 lentelė. Mažo svorio žodžio tiesiniame kode radimo atakos rezultatų bendroji statistika

McEliece kriptosistemos parametrai	$n = 32$, $m = 5$, $t = 4$, $k = 12$	$n = 64$, $m = 6$, $t = 5$, $k = 34$	$n = 128$, $m = 7$, $t = 8$, $k = 92$	$n = 256$, $m = 8$, $t = 13$, $k = 152$	$n = 512$, $m = 9$, $t = 12$, $k = 404$
Atakos parametrai	$p = 1$, $\sigma = 2$	$p = 1$, $\sigma = 4$	$p = 1$, $\sigma = 6$	$p = 1$, $\sigma = 7$	$p = 2$, $\sigma = 16$
Atakos vykdymo skaičius	1000	1000	1000	1000	1
Vidutinis atakos iteracijų skaičius	8,99	24,77	286,57	16515,43	50946
Vidutinis atakos laikas milisekundėmis	0,94	12,08	281,58	53977,35	53165226
Pradinės Gauso eliminacijos laikas milisekundėmis	0,01	2,1	19,98	205,65	4415
Galutinės Gauso eliminacijos laikas milisekundėmis	0,01	1,07	11,7	127,57	2131

Remiantis trečiąja lentele galima pastebėti, jog vidutinė atakos trukmė, kai $\mathrm{m} = 5$, yra vos 1 milisekundė, o m reikšmę padidinus vos trimis, ataka vidutiniškai užtrunka net apie 57,5 tūkstančio kartų ilgiau. Taip pat įdomu pastebėti, jog taikant Gauso eliminaciją tik tiesiškai nepriklausomiems stulpeliams (galutinė Gauso eliminacija atakoje), jos vykdymo trukmė

sumažėja per pusę (lyginant su Gauso eliminacija, vykdoma su atsitiktiniais ir nebūtinai tiesiškai nepriklausomais stulpeliais). Nors vertinti vienintelį atakos bandymą su paskutiniojo trečiosios lentelės stulpelio McEliece kriptosistemos ir mažo svorio žodžio tiesiniame kode radimo atakos parametrais yra sudėtinga, tačiau galima daryti prielaidą, jog panaudojus saugiausius kriptosistemos parametrus, kai $\mathrm{m} = 9$, bendras atakos laikas teoriškai turėtų būtų ilgesnis nei 15 valandų.

4.3. Mažo svorio žodžio tiesiniame kode radimo atakos duomenų ekstrapoliacija

Pasinaudojus trečiosios lentelės duomenimis galima bandyti ekstrapoliuoti gautus mažo svorio žodžio tiesiniame kode radimo atakos rezultatus siekiant numatyti, kiek užtruktų ši ataka, jei ji būtų vykdoma su saugiausiais McEliece kriptosistemos ir efektyviausiais atakos parametrais, kai $\mathrm{m} = 9$ ir daugiau. Ekstrapoliacija buvo atliekama tokiu būdu – kadangi vidutiniški atakos laikai su skirtingais kriptosistemos ir atakos parametrais didėja per daug sparčiai, jiems buvo pritaikyta logaritminė funkcija pagrindu 2. Toks sprendimas buvo priimtas dėl to, jog jei patys atakų laikai atitinka y ašį, o x – kriptosistemos parametras m, pats m yra reiškinio $n = 2^m$ rodiklis. Tuo remiantis buvo sudaryta ketvirtoji lentelė.

4 lentelė. Preliminarios funkcijos, nusakančios atakos laikų priklausomybę nuo kriptosistemos parametro m, žinomi taškai

x (McEliece kriptosistemos parametro m reikšmė)	**f(x)** ($log_2 a$, kur a yra vidutinė atakos trukmė milisekundėmis)
5	-0,0862
6	3,5949
7	8,1374
8	15,72

Pasinaudojus ketvirtąja lentele bei internetiniu įrankiu (https://www.dcode.fr/function-equation-finder), sudarančiu pasirinktos funkcijos tipo formulę, buvo gautos dvi formulės. Pasirinkti tipai buvo parabolė ir eksponentė, nes abi funkcijos pasižymi labai sparčiai ir progresyviai didėjančiomis y reikšmėmis (dėl to netinka ir tiesinė lygtis). Gautos formulės:

1. $f(x) = 0.975375x^2 - 7.48377x + 13.0572$ (parabolė).
2. $f(x) = 0.965237e^{0.394078x} - 6.90978$ (eksponentė).

Kadangi yra pakankamai sudėtinga nuspręsti, kurią formulę naudoti, buvo padaryta tokia prielaida: atakos laikas yra tiesiogiai susijęs su atakai reikalingų atlikti dvejetainių operacijų skaičiaus formule, kurios dalis reikšmių yra apskaičiuotos 8-ame paveikslėlyje pateiktoje lentelėje, o kita dalis – šaltinyje [KI03] (kai $m = 10,11$). Dėl šios priežasties galima sukurti naują x ir y priklausomybę, kur $x = m$, o $f(x) = W_{p,\sigma}$. Remiantis tuo buvo gauta 5-oji lentelė.

5 lentelė. Preliminarios funkcijos, nusakančios atakai atlikti reikalingų dvejetainių operacijų skaičiaus priklausomybę nuo kriptosistemos parametro m, žinomi taškai

x (McEliece kriptosistemos parametro m reikšmė)	**f(x)** ($log_2 b$, kur b yra atakai reikalingų dvejetainių operacijų skaičius)
5	9,3686
6	12,1615
7	16,7631
8	24,5859

Pasinaudojus penktąja lentele bei internetiniu įrankiu (https://www.dcode.fr/function-equation-finder), sudarančiu pasirinktos funkcijos tipo formulę, buvo gautos tokios formulės:

1. $f(x) = 1.25747x^2 - 11,3218x + 34.6115$ (parabolė);
2. $f(x) = 0.30121e^{0.519699x} + 5,32971$ (eksponentė).

Remiantis [KI03], kai $m = 10$, reikalingas dvejetainių operacijų skaičius atakai yra apie 2^{62}, kai $m = 11$ – apie 2^{106}, o šių skaičių atitikmenys formulėse atsispindi štai taip:

1. $f(10) = 1.25747 \times 10^2 - 11,3218 \times 10 + 34.6115 \approx 47,14$, $f(11) \approx 62,23$.
2. $f(10) = 0.30121e^{0.519699 \times 10} + 5,32971 \approx 59,77$, $f(11) \approx 96.87$.

Remiantis šiais skaičiavimais galima daryti prielaidą, jog eksponentės funkcija pateikia tikslesnius dvejetainių operacijų skaičiaus rezultatus nei parabolės. Dėl šios priežasties yra geriau naudoti eksponentės funkciją ekstrapoliacijai nei parabolės. Šeštoje lentelėje pateikiami ekstrapoliuoti atakos laiko duomenys pasinaudojus $f(x) = 0.965237e^{0.394078x} - 6.90978$.

6 lentelė. Ekstrapoliacijos būdu gauti preliminarūs atakos laikai didesniems m parametrams

McEliece kriptosistemos parametras m	Preliminarus atakos laikas pritaikius ekstrapoliaciją (milisekundėmis)
9	$2^{26,58}$

McEliece kriptosistemos parametras m	Preliminarus atakos laikas pritaikius ekstrapoliaciją (milisekundėmis)
10	$2^{42,76}$
11	$2^{66,75}$
12	$2^{102,33}$

Remiantis šeštąja lentele bei pasirinkus $m = 9$, galima apskaičiuoti, jog bandymams naudotas kompiuteris atakai įvykdyti vidutiniškai užtruktų apie 27,89 valandas, o kai $m = 11$ – net apie 3,935 milijardo metų. Siekdami palyginti šeštos lentelės duomenis su šiuo metu galingiausiu pasaulyje superkompiuteriu, visų pirma apsibrėžkime jo technines charateristikas.

Oficialiame kompanijos „Fujitsu“ puslapyje (https://www.fujitsu.com/global/about/innovation/fugaku/specifications/) nurodoma, jog šios įmonės „Fugaku“ superkompiuteris susideda iš 158976 mazgų *A64FX*, iš kurių kiekvienas pasižymi 52 branduolių 64 bitų *ARM* architektūros procesoriumi. Kadangi bandymams naudotas kompiuteris naudoja *Intel* procesorių, kurio architektūra yra *x86* (64 bitų), atakų laikų palyginimas vyks tokiu būdu:

1. Apskaičiuosime, kiek slankaus kablelio operacijų per sekundę (*FLOPS*) gali atlikti bandymams naudotas kompiuteris bei visi superkompiuterio mazgai kartu.
2. Pagal bandymams naudoto kompiuterio procesoriaus apkrovos kiekį procentais atakos vykdymo metu bei pasiekiamą maksimalų procesoriaus *FLOPS* skaičių apsibrėšime, koks slankaus kablelio operacijų per sekundę greitis yra panaudojamas vienam atakos bandymui.
3. Atakos trukmę, kai $m = 11$, padalinsime tiek kartų, kiek kartų yra didesnis superkompiuterio *FLOPS* skaičius lyginant su bandymams naudotu kompiuteriu.

Fugaku superkompiuterio *FLOPS* greitis (operacijoms su 32 bitų slankaus kablelio kintamuoju) yra $1{,}03 \times 10^{18}$ (nurodyta oficialiame *Fugaku* superkompiuterio techninių specifikacijų puslapyje), o bandymams naudoto kompiuterio - $403{,}2 \times 10^{9}$ (https://askgeek.io/en/cpus/Intel/Core-i5-7200U). Vidutiniškai vykdyti atakai paleista programa sunaudoja apie 28 procentus visų procesoriaus resursų. Programa nenaudoja paralelizmo, tačiau vis tiek naudoja 8 gijas, kurioms pagal prioritetą bei skirtingų procesoriaus apkrovų lygius yra išskiriami skaičiavimo resursai nepriklausomai nuo branduolio. Remiantis panaudota procentine procesoriaus sąnaudų išraiška galime apskaičiuoti, jog vienam atakos bandymui reikia maždaug $(403{,}2 \times 10^{9}) \times 0{,}28 = 112{,}896 \times 10^{9}$ *FLOPS* greičio. Tuomet

superkompiuteris šią ataką galėtų atlikti $1{,}03 \times 10^{18} \div 112{,}896 \times 10^{9} \approx 9034863{,}946$ kartų greičiau nei bandymams naudotas kompiuteris idealiu atveju išnaudojus visus įmanomus superkompiuterio resursus ir darant prielaidą, jog nepaisant procesorių architektūrų operacijų skirtumų programa būtų vykdoma identišku efektyvumu. Po atliktų skaičiavimų atakos vykdymo laikas naudojantis superkompiuteriu sumažėtų iki $2^{66{,}75} \div 9034863{,}946 \approx 1{,}374 \times 10^{13}\ ms$, kas yra maždaug 435,54 metų. Svarbu paminėti, kad tai yra trumpiausias įmanomas atakos laikas, nes, kaip pavyzdys, tuo pačiu metu paleistos dvi programos, vykdančios ataką su identiškais kriptosistemos ir atakos parametrais, dėl atsitiktinumo faktoriaus tik teoriškai gali sutrumpinti atakos laiką per pusę (vienu metu yra atliekama du kartus daugiau spėjimų). Tad tai nebūtinai reiškia, jog atakos laikas dėl dviejų paleistų programų būtinai sutrumpės per pusę. Įdomumo dėlei paskaičiavus numatomą trumpiausią laiką superkompiuteriui įvykdyti ataką, kai $m = 12$, yra gaunamas toks rezultatas: $2^{102{,}33} \div 9034863{,}946 \approx 7{,}055 \times 10^{23}\ ms$, kas yra maždaug apie $2{,}237 \times 10^{13}$ metų, o kai $m = 10$: $2^{42{,}76} \div 9034863{,}946 \approx 824367{,}758\ ms$, kas yra vos 13,7 minučių.

REZULTATAI IR IŠVADOS

Pasiekti rezultatai:

1. Susipažinta su McEliece kriptosistema.
2. Susipažinta su susijusio pranešimo, pranešimo persiuntimo bei mažo svorio žodžio tiesiniame kode radimo atakomis.
3. Realizuotos 2 punkte paminėtos atakos.
4. Surinkta bei išanalizuota statistika vykdant 2 punkte minėtas atakas.
5. Padarytos išvados.

McEliece kriptosistemos saugumą baigiamojo darbo kontekste galima vertinti tik vykdytų atakų, minėtų 2 skyriuje, atžvilgiu.

Išvados, gautos įvykdžius kritines atakas (susijusio pranešimo bei pranešimo persiuntimo):

1. Remiantis atlikto tyrimo su susijusio pranešimo bei pranešimo persiuntimo atakų rezultatais yra patartina vengti siųsti pasikartojantį pranešimą nesaugiu kanalu.
2. Daugiausiai laiko užimanti dalis susijusio pranešimo bei pranešimo persiuntimo atakų algoritme yra Gauso eliminacija. Optimizavus šio algoritmo realizaciją, atakų laikus būtų galima sumažinti.
3. L_0 aibės galių vidurkio ir vidutinio spėjimų skaičiaus, reikalingo pasirinkti klaidų vektoriaus neiškraipytus stulpelius, neatitikimas pagal pirmąją formulę tiesiogiai susijęs su L_0 aibės galios skirstiniu.
4. Didėjant teoriniam vidutiniam spėjimų skaičiui, rekalingam pasirinkti klaidų vektoriaus neiškraipytus stulpelius, proporcingai reikia didinti ir atakų bandymų skaičių, jog praktiniai skaičiavimai būtų kuo artimesni teoriniams.

Nors susijusio pranešimo bei pranešimo persiuntimo atakos ir reikalauja papildomos informacijos, ją išgauti nėra sudėtinga, kaip, pavyzdžiui, aptikti pranešimo persiuntimo sąlygą (bandyti sudėti 2 nesaugiame kanale perimtus vektorius ir stebėti, ar jų sumos vektoriaus svoris nėra artimas 2 klaidų vektorių svorių sumai, atsižvelgiant į galimus kriptosistemos parametrus). Šios atakos yra išskirtinai nukreiptos prieš McEliece kriptografinę sistemą, nes naudojasi būtent šios kriptografinės sistemos užšifravimo specifika – atstumas tarp tų pačių užšifruotų atvirųjų pranešimų vektorių niekada neviršys dviejų klaidų vektorių Hemingo svorių sumos. Šį kriptosistemos elgesį galima traktuoti kaip silpnąją jos vietą. Atakos yra orientuotos į tai, kaip kriptografinė sistema bus panaudojama, t.y. kaip tas pats ar susiję atvirieji pranešimai yra užšifruojami. Dėl šios priežasties atsižvelgiant į atakų naudotus principus galima pradėti apibrėžti

McEliece kriptografinės sistemos naudojimosi protokolą, į kurį būtų galima įtraukti šią rekomendaciją – naudoti atsitiktinio simbolio (ar simbolių) pridėjimą prie atvirojo pranešimo. Tokiu atveju užšifruojant tą patį atvirąjį pranešimą visuomet bus gaunami užkoduoti ir iškraipyti vektoriai, tarp kurių Hemingo atstumas viršys dviejų klaidų vektorių Hemingo atstumą. Kalbant apie atakų algoritmo efektyvumą, atakų vykdymo laiką galima sutrumpinti pritaikant tokį nepriklausomų viešojo rakto matricos stulpelių paieškos būdą, kaip ir mažo svorio žodžio tiesiniame kode radimo atakoje (vykdant Gauso eliminaciją ir suradus tiesiškai priklausomą viešojo rakto matricos stulpelį pasirinkti kitą stulpelį, nei vykdyti Gauso eliminaciją iš naujo). Taip pat remiantis atakų algoritmais bei suradus tiesiškai nepriklausomų stulpelių indeksų aibę įvykdžius ataką vieną kartą, ši aibė gali būti panaudojama ir kitiems atakų bandymams, jei visi tos aibės nariai įeis į kitų atakų bandymų metu gautas L_0 aibes. Viską apibendrinus svarbu paminėti, jog nors šios atakos esant išskirtinėms sąlygoms leidžia įsilaužėliui sužinoti, kokie pranešimai buvo siunčiami nesaugiu kanalu, tačiau bendru požiūriu jos kriptosistemos neįveikia, t.y. nėra sužinomas privatus raktas.

Išvados, gautos įvykdžius nekritinę ataką (mažo svorio žodžio tiesiniame kode radimas):

1. Remiantis ekstrapoliacijos ir papildomų skaičiavimų rezultatais bei mažo svorio žodžio tiesiniame kode radimo ataka, pasirinkus pakankamai didelius McEliece kriptosistemos parametrus (pavyzdžiui, kai $m = 11$ arba $m = 12$), kriptosistema nėra įveikiama šios atakos požiūriu.
2. Remiantis ekstrapoliacijos ir papildomų skaičiavimų rezultatais, mažo svorio žodžio tiesiniame kode radimo ataka ir McEliece kriptosistemos parametrais, kai $m = 10$ (tai buvo rekomenduojamas parametras, kai buvo pristatyta pati kriptosistema 1978 metais [McE78]), kriptosistema yra iš dalies įveikiama.
3. Remiantis mažo svorio žodžio tiesiniame kode radimo atakos bei McEliece kriptosistemos užšfiravimo algoritmais galima teigti, jog nepriklausomai nuo to, kiek kartų iš eilės bus įvykdytos atakos, jose bei pačioje kriptosistemoje naudojamas atsitiktinumo faktorius apriboja galimybes atakų vykdymo laikams palaipsniui mažėti, t.y. neįmanoma pasinaudoti prieš tai buvusios atakos duomenimis.

Nors mažo svorio žodžio tiesiniame kode radimo atakai nėra reikalinga jokia papildoma informacija, sėkmingai išsiaiškinti nesaugiu kanalu siųstą bei perimtą pranešimą, kai yra naudojami saugūs kriptosistemos parametrai, pavyzdžiui, kai $m = 11$, praktiškai nėra įmanoma net ir naudojant galingiausią superkompiuterį. Ši ataka taip pat nekelia pavojaus kriptografinės sistemos įveikimui, nes sėkmingu atakos atveju yra gaunamas tik atvirasis pranešimas, o ne

privatus kriptosistemos raktas. Svarbu atkreipti dėmesį į tai, jog ši ataka taip pat yra nukreipta prieš atsitiktinio klaidų vektoriaus panaudojimą.

Bendrosios išvados, atliktos remiantis vykdytomis atakomis, jų algoritmais bei gautais vykdytų eksperimentų rezultatais:

1. Remiantis vykdytų atakų algoritmais bei McEliece kriptosistemos užšfiravimo algoritmu galima teigti, jog lygiagrečiai vykdomos atakos dėl atsitiktinumo faktoriaus gali būti įvykdomos greičiau, nei jas vykdant viena po kitos.
2. Remiantis gautais rezultatais, tyrimais, pasinaudojus atliktomis išvadomis, rekomendacijomis bei darant prielaidą, jog atliktos atakos kelia didžiausią pavojų McEliece kriptosistemos saugumui, galima daryti išvadą, jog McEliece kriptografinė sistema su pasirinktais saugiais parametrais (pavyzdžiui, kai $m = 11$ arba $m = 12$) yra galima alternatyva šiuo metu naudojamoms kriptosistemoms spartaus kvantinių kompiuterių vystymosi kontekste iš saugumo perspektyvos.

Siekiant naudoti McEliece kriptografinę sistemą kaip alternatyvą esamoms, tokioms kaip RSA, yra labai svarbu parinkti ne tik saugiausius parametrus, tačiau ir apsibrėžti naudojimosi kriptosistema protokolą. Remiantis visų įvykdytų atakų rezultatais galima daryti išvadą, jog siekiant naudotis McEliece kriptosistema ne tik dabar, tačiau ir netolimoje ateityje, yra patartina naudoti $m >= 11$. McEliece kriptosistemos atsparumas P. Shor algoritmui, skirtam kvantiniams kompiuteriams, bei šio saugumo tyrimo gauti rezultatai (kriptosistemos atsparumas mažo svorio žodžio tiesiniame kode radimo atakai) taip pat leidžia daryti išvadas, jog ši kriptografinė sistema turi šansų būti naudojama laikais, kai pasaulis praktiškai pritaikys didelio kubitų skaičiaus kvantinius kompiuterius.

ŠALTINIAI

[Ber97] Thomas A. Berson. *Failure of the McEliece Public-Key Cryptosystem under Message-Resend and Related-Message Attack*. Burton S. Kaliski Jr. (Ed.), *Advances in Cryptology - CRYPTO '97, Proceedings of 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August* 1997, *Lecture Notes in Computer Science* 1294, p. 213-220, 1997.

[CS98] Anne Canteaut, Nicolas Sendrier. *Cryptanalysis of the Original McEliece, Advances in Cryptology – ASIACRYPT'98,* p. 187-199, 1998.

[Kes20] Gary C. Kessler. *An Overview of Cryptography*. Prieiga per internetą: https://www.garykessler.net/library/crypto.html. Žiūrėta 2021-05-26.

[KI03] Kazukuni Kobara, Hidemi Imai. *On the One-Wayness Against Chosen-Plaintext Attacks of Loidreau's Modified McEliece PKC, IEEE Transactions on Information Theory 49(12),* p. 3160-3168, 2003.

[McE78] Robert J. McEliece. *A Public-Key Cryptosystem Based On Algebraic Coding Theory, DSN Progress Report, Jet Propulsion Laboratory, Pasadena, CA*, p. 114–116, 1978.

[Neh20] Neha T. *Public Key Cryptography*. Prieiga per internetą: https://binaryterms.com/public-key-cryptography.html. Žiūrėta 2021-05-26.

[NMB+12] Robert Niebuhr, Mohamed Meziani, Stanislav Bulygin, Johannes A. Buchmann. *Selecting Parameters for Secure McEliece-based Cryptosystems, International Journal of Information Security*, p. 137-147, 2012.

[OR20] Lukasz Olejnik, Robert Riemann. *TechDispatch #2/2020: Quantum Computing and Cryptography*. Prieiga per internetą: https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-22020-quantum-computing-and_en. Žiūrėta 2021-05-26.

[Sot19] Ivan Alejandro Moreno Soto. *McEliece*. Prieiga per internetą: https://github.com/rexemin/McEliece. Žiūrėta 2021-05-26.

SĄVOKŲ APIBRĖŽIMAI

Algoritmas – tai baigtinė aiškiai apibrėžtų, kompiuteriu įgyvendinamų instrukcijų seka.

Baitas – tai skaitmeninės informacijos vienetas, kurį sudaro 8 bitai.

Bitas – *binary digit* (dvejetainis skaitmuo) – tai pagrindinis kompiuterio skaičiavimo vienetas.

***Bool* –** tai kintamasis, saugantis vieną iš dviejų reikšmių – teisinga ar neteisinga.

***Cache* atmintis –** procesoriaus naudojama didelio greičio atmintis, naudojama siekiant sumažinti laiką ir energiją norint pasiekti duomenis iš pagrindinės atminties.

Gauso eliminacija – eilučių redukcija – tai yra tiesinės algebros algoritmas, skirtas spręsti tiesinių lygčių sistemą.

Gija – tai mažiausia užprogramuotų instrukcijų seka, kurią tvarkyklė gali valdyti savarankiškai.

Kriptografija - tai mokslas, naudojantis matematiką duomenims užšifruoti ir dešifruoti.

Kubitas - tai pagrindinis kvantinės informacijos vienetas - klasikinio dvejetainio bito kvantinė versija, fiziškai realizuota dviejų būsenų įtaisu.

***Liskov Substitution* principas –** jei S yra T potipis, tuomet T tipo objektus galima pakeisti S tipo objektais (t.y. T tipo objektas gali būti pakeistas bet kuriuo S potipio S objektu), nekeičiant nei vienos iš pageidaujamų programos savybių (korektiškumas, atlikta užduotis ir kt.).

***Open-closed* principas –** programinės įrangos esybės (klasės, moduliai, funkcijos ir kt.) turėtų būti atviri išplėtimui, tačiau uždaryti modifikavimui.

***Singleton* –** tai programinės įrangos projektavimo šablonas, ribojantis klasės kūrimo objekto skaičių iki vieno.

String - tai kintamasis, saugantis simbolių seką.

Superpozicija – kvantinė mechanikos ypatybė, leidžianti tam tikroms dalelėms turėti kelias reikšmes vienu metu.

Taktinis dažnis – tai skaičius, nusakantis, kiek taktų per sekundę yra atliekama procesoriuje.

Viešojo rakto kriptografinė sistema - tai kriptografinė sistema, naudojanti raktų poras: viešuosius raktus, kurie gali būti platinami plačiai, ir privačius raktus, kuriuos žino tik savininkas.

SANTRUMPOS

C# - programavimo kalba, sukurta bei vystoma *Microsoft* kompanijos.

HTTPS - yra hiperteksto perdavimo protokolo (HTTP) plėtinys, naudojamas saugiam ryšiui kompiuterių tinkle.

JSON – *JavaScript Object Notation* – duomenų keitimosi formatas.

LPDDR3 – tai trečios kartos dvigubo duomenų perdavimo greičio sinchroninė dinaminė operatyvioji atmintis, kuri sunaudoja mažiau energijos ir yra skirta mobiliesiems kompiuteriams.

RSA - viešojo rakto kriptosistema, kurios algoritmą 1977 metais sukūrė Ronald Rivest, Adi Shamir ir Leonard Adleman.

SOLID – tai 5 projektavimo šablonų akronimas, skirtas programinės įrangos projektavimą padaryti suprantamesnį, lankstesnį bei prižiūrimesnį.

SSD - tai kietojo kūno saugojimo įrenginys, skirtas saugoti duomenis kompiuteryje.

TLS - kriptografinis protokolas, skirtas užtikrinti ryšių saugumą kompiuterių tinkle.