Kauno technologijos universitetas

Informatikos fakultetas

Steganografijos metodas RTP protokole

Baigiamasis magistro projektas

Karolis Zdanevičius

Projekto autorius

Prof. dr. Algimantas Venčkauskas

Vadovas

Kaunas, 2021

Kauno technologijos universitetas

Informatikos fakultetas

Steganografijos metodas RTP protokole

Baigiamasis magistro projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Karolis Zdanevičius

Projekto autorius

Prof. dr. Algimantas Venčkauskas

Vadovas

Prof. dr. Jevgenijus Toldinas

Recenzentas

Kaunas, 2021

Kauno technologijos universitetas

Informatikos fakultetas

Karolis Zdanevičius

Steganografijos metodas RTP protokole

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Karolis Zdanevičius

Patvirtinta elektroniniu būdu

Zdanevičius, Karolis. Steganografijos metodas RTP protokole. Magistro baigiamasis projektas / vadovas prof. dr. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatikos katedra fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): informatikos inžinerijos kryptis, technologijos mokslų studijų sritis.

Reikšminiai žodžiai: steganografija, RTP, protokolas, slaptas, paskutinis bitas, Reed, Solomon.

Kaunas, 2021. 49 p.

Santrauka

Steganografija tinkle leidžia siųsti slaptą informaciją naudojantis įvairiais, protokolų nenaudojamais, antraštės laukais. Toks steganografijos metodas jau yra naudojamas seniai. Pagrindiniai tokio steganografijos tipo trūkumai yra siunčiamos slaptos informacijos integralumas bei jos nedidelis kiekis. Šiame darbe buvo analizuojami įvairūs TCP/IP grupės protokolų steganografijos metodai. Išsikeltas pagrindinis tikslas – sukurti naują arba patobulinti esamą steganografijos metodą RTP protokole. Tikslas buvo pasiektas patobulinus paskutiniojo balso duomenų bito keitimo metodą. Buvo panaudotas *Reed-Solomon* klaidų aptikimo ir taisymo kodas, kuris įvykus paketų praradimui, gali atstatyti pradinę žinutę. Taip pat darbe buvo tiriama ar galima siunčiamos žinutės bitus slėpti ne tik paskutiniame balso duomenų bite, o bet kuriame iš paskutinių keturių, balso duomenų, bitų arba naudojant du paskutinius balso duomenų bitus.

Zdanevičius, Karolis. Steganography method in RTP protocol. Master's Final Degree Project / supervisor prof. dr. Algimantas Venčkauskas; The Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): field of informatics engineering, technological sciences area.

Keywords: steganography, RTP, protocol, secret, last bit, Reed, Solomon.

Kaunas, 2012. 49 p.

Summary

Steganography in networks allows sensitive information to be sent using a variety of header fields that are not used by protocols. Such steganographic methods have been used for a long time now. The main disadvantages of this type of steganography are the integrity of the transmitted confidential information and its small amount. In this work, various steganography methods of TCP/IP stack protocols were analyzed. The main goal is to develop a new or improve the existing steganography method in the RTP protocol. The goal was achieved by improving the method of changing the last voice data bit. A *Reed-Solomon* error detection and correction code was used. This method can recover the original message in the event of packet loss. It was also investigated whether the bits of the sent message could be hidden not only in the last bit of voice data, but in any of the last four bits of voice data or even using the last two bits of voice data.

.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų sąrašas

Santrumpos:

ARP – angl. Address Resolution Protocol - yra ryšio protokolas, naudojamas norint rasti kanalo lygio (OSI modelis) adresą.

BCH – angl. - Bose–Chaudhuri–Hocquenghem codes - ciklinių klaidų taisymo kodų klasė, kurioje yra naudojami polinomai ir baigtiniai laukai.

DNS – Domain Name System - yra hierarchinė ir decentralizuota kompiuterių, paslaugų ar kitų informacinių išteklių, prijungtų prie interneto ar privataus tinklo, pavadinimų sistema.

FTP – angl. File Transfer Protocol – elektroninių bylų siuntimo protokolas.

HICCUPS – angl. Hidden Communication System for Corrupted Networks – steganografijos metodas, kai yra siunčiami kanalo lygio paketai su tyčia klaidingomis kontrolinėmis sumomis.

ICMP – angl. Internet Control Message Protocol – protokolas naudojama tinklo įrenginiuose, įskaitant maršrutizatorius, siųsti klaidų pranešimus, rodančią sėkmingą ar nesėkmingą komunikaciją su kitu IP adresu.

IGMP – angl. Internet Group Management Protocol - yra ryšio protokolas, kurį naudojamas IP tinkluose, kad nustatytų grupinio transliavimo adreso narystę.

IMAP – angl. Internet Message Access Protocol – protokolas skirtas atsisiųsti elektroninius laiškus iš serverio.

IP – angl. Internet Protocol – komunikacijos protokolas skirtas persiųsti duomenų paketus.

IPsec – angl. Internet Protocol Security - yra saugaus tinklo protokolų rinkinys, kuris autentifikuoja ir užšifruoja duomenų paketus, kad būtų užtikrintas saugus interneto ryšys.

OFDM – angl. Orthogonal Frequency-Division Multiplexing - yra skaitmeninio perdavimo tipas ir skaitmeninių duomenų kodavimo metodas kai yra naudojami keli dažniai.

PPP – angl. Point-to-Point Protocol - ryšio protokolas tarp dviejų maršrutizatorių tiesiogiai be jokio pagrindinio kompiuterio ar kito tinklo įrenginio.

RS-232 – angl. Recommended Standard 232 – protokolas skirtas nuosekliam duomenų perdavimui.

RSTEG – angl. Retransmission Steganography - technika, kai yra „išprovokuojamas“ paketo persiuntimas, siekiant siųsti slaptą informaciją.

RTP – angl. Real Time Transfer Protocol – protokolas skirtas siųsti garso ir vaizdo duomenis.

SMTP – angl. Simple Mail Transfer Protocol – standartinis elektroninių laiškų siuntimo protokolas.

SSH – angl. Secure Shell Protocol - yra kriptografinis tinklo protokolas, skirtas saugiai siųsti informaciją neapsaugotame tinkle.

STP – angl. Spanning Tree Protocol - yra tinklo protokolas, kuris sukuria be kilpę laidinio tinklų loginę topologiją.

TCP – angl. Transmission Control Protocol - suteikia patikimą, surikiuotą ir klaidas aptinkantį informacijos duomenų srautą iš oktetų (baitų), veikiantį informacinėse sistemose, kurios bendrauja per IP tinklą.

VoIP – angl. Voice over IP - yra balso ryšio ir kt. daugialypės terpės sesijų siuntimo būdas naudojantis interneto protokolo (IP) tinklus.

Wi-Fi – yra belaidžio tinklo protokolų šeima, paremta IEEE 802.11 standartų grupe.

ĮVADAS

Steganografijos metodas RTP protokole – informacijos ir informacinių technologijų saugos programos magistro baigiamasis darbas.

Sprendžiama problema

Steganografijos metodų taikymas tinkle paskutiniais metais yra gana aktyviai tyrinėjama sritis. Šios srities ekspertai yra plačiai išnagrinėję pagrindinius TCP/IP protokolus ir juose taikomus steganografijos metodus. Tačiau atsiranda naujos versijos gerai žinomų protokolų (pvz. HTTP/2), protokolų standartai yra keičiami ar atnaujinami.

Didžiausia problema steganografijai tinkle yra persiunčiamų slaptų duomenų kiekis bei jų integralumas. Steganografija RTP protokole gali persiųsti labai didelius kiekius informacijos, tačiau naudojant UDP protokolą nėra garantuotas visų RTP paketų persiuntimas galutiniam vartotoju. Arba paketai gali būti persiųsti ne pagal eilę. Šių problemų sprendimui galima būtų panaudoti klaidų aptikimo ir taisymo algoritmus.

Darbo tikslas ir uždaviniai

Tikslas – nustatyti ir išanalizuoti metodus naudojamus slaptiems duomenų perdavimams RTP protokolu. Sukurti ir išbandyti patobulintą steganografijos RTP protokole metodą, kuris naudotų klaidų aptikimo ir taisymo algoritmą.

Uždaviniai:

- susipažinti su steganografijos metodais TCP/IP protokolų steke;
- atlikti žinomų steganografijos metodų analizę;
- išanalizuoti steganografijos metodus RTP protokole;
- sumodeliuoti naują steganografijos metodą RTP protokole arba patobulinti esamą;
- išanalizuoti gautus rezultatus.

Darbas susideda iš keturių dalių. Pirmoje dalyje yra apžvelgiami ir analizuojami žinomi TCP/IP šeimos protokolų steganografiniai metodai, jų galimybės. Antroje dalyje yra analizuojami steganografijos metodai RTP protokole, aprašoma, kaip būtų galima patobulinti steganografijos RTP protokole metodą, kai slaptai informacijai persiųsti yra naudojamas paskutinysis balso duomenų bitas. Trečiojoje dalyje yra aprašomas prototipas, jo veikimo galimybės ir apribojimai, bei atliktas tyrimas. Paskutinė darbo dalis yra išvados.

1. STEGANOGRAFIJOS TCP/IP ANALIZĖ

Analizės tikslas:

- išanalizuoti, kas yra steganografija, panaudojimo galimybės ir steganografijos metodus;
- išanalizuoti steganografijos galimybes tinkle;
- išanalizuoti ir palyginti steganografijos TCP/IP protokole sprendimus.
- išanalizuoti steganografijos RTP protokole būdus.

1.1. STEGANOGRAFIJA

Žodis steganografija išvertus iš graikų kalbos reiškia paslėpti rašmenys. Steganografijos tikslas yra paslėpti informaciją, t. y., jog apie apsikeičiamą informaciją žinotų tiktai siuntėjas ir gavėjas. Šiais skaitmenizacijos laikais galima sakyti, jog tai yra informacijos slėpimas informacijoje. Steganografija dažnai yra dar vadinama slaptu kanalu. Svarbu steganografijos nepainioti su kriptografija, kas yra rašmenų užšifravimas, bet ne slėpimas. Kad būtų įvykdyta steganografija reikalingi du dalykai – žinutė (informacija) ir nešėjas (angl. – *carrier*). Žinutė tai yra slapta informacija, kurią norime perduoti, o nešėjas - tai į ką bus įkomponuota žinutė [1].

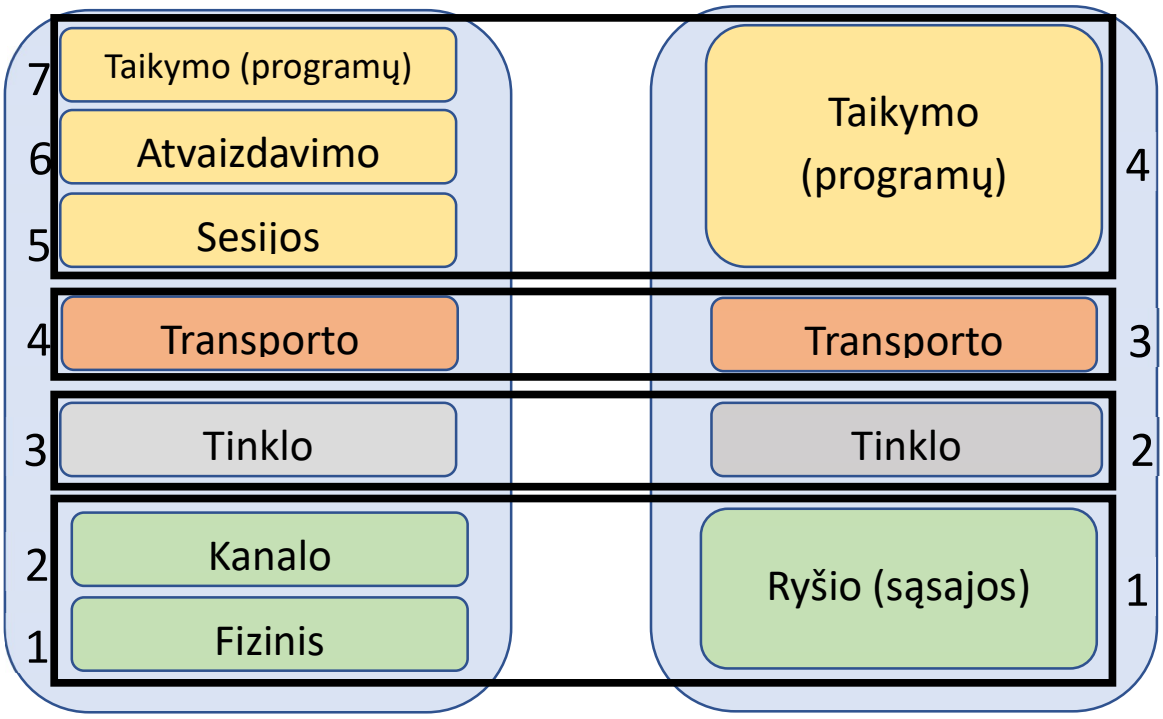
Šaltiniuose dažniausiai yra minimi keturi pagrindiniai steganografijos būdai, kurie suskirstyti pagal tai, kokioje terpėje (t. y. koks yra nešėjas) yra paslėpta informacija:

- Steganografija tekste - galime teigti jog tai seniausias steganografijos būdas – egiptietiški hieroglifai, Cezario šifras ir t.t. Steganografija tekste yra metodas, kai yra naudojamas tekstas norint paslėpti žinutę. Šiais laikais steganografija naudojant tekstą galima trimis būdais [2]: formatavimo metodas – tai kai informacija slepiama keičiant teksto formatą; atsitiktinis ir statistinis metodas – naudojamas algoritmas, kuris remiantis pateikta slapta informacija sugeneruoja naują tekstą [3]; kalbotyros metodas – tai sintaksės ir semantikos derinys.
- Steganografija vaizde - vaizdai yra populiariausi objektai ir seniausiai šiuolaikiniame pasaulyje naudojami objektai steganografijai. Skaitmeninių vaizdų srityje egzistuoja daugybė skirtingų vaizdo failų formatų, dauguma jų skirti konkrečioms programoms. Šiems skirtingiems vaizdo failų formatams egzistuoja skirtingi steganografiniai algoritmai. Vaizdų steganografijos metodus galima suskirstyti į dvi grupes: vaizdo srities ir transformavimo srities [4]. Vaizdo srities metode, žinutės įterpiamos tiesiogiai tarp pikselių, o transformavimo - vaizdai pirmiausia transformuojami, o tada pranešimas įdedamas į atvaizdą.
- Steganografija garse - kai slapta informacija yra įterpiama į skaitmeninius garso įrašus, vadiname steganografija garse.
- Steganografija tinkle - steganografijos metodas, kai slapta informacija yra įterpiama į įvairius OSI modelio tinklo protokolus. Yra keletas taikymo būdų, kai naudojamos neišnaudotos paketų antraščių vietos arba naudojami metodai, modifikuojantys paketų srautus, arba hibridinis metodas, kai modifikuoja abu – paketų turinį ir jų transliavimo laiką bei tvarką [5].

1.2. STEGANOGRAFIJA TINKLE

Kiekvieną sekundę tinklais yra siunčiamas milžiniškas kiekis informacijos. Kad būtų užtikrintas sklandus informacijos dalijimasis ir informacija nebūtų iškraipoma, yra naudojama begalė skirtingų protokolų. Protokolas yra standartinis taisyklių rinkinys, leidžiantis elektroniniams prietaisams komunikuoti tarpusavyje. Šios taisyklės apima tai, kokio tipo duomenys gali būti perduodami, kokios komandos gali būti naudojamos duomenims siųsti ir gauti, kaip patvirtinamas duomenų perdavimas ir pan. Protokolų taisyklės yra aprašytos RFC (angl. *Request for Comments* trumpinys). Tai tarptautiniai dokumentai, kuriuos gali rengti įvairios institucijos - Interneto inžinerijos darbo grupė (angl. *The Internet Engineering Task Force*), Interneto tyrimų darbo grupė (angl. *Internet Research Task Force*) ir kitos.

Kai skirtingi įrangos gamintojai pradėjo plėtoti skirtingų technologijų tinklinę įrangą bei skirtingus protokolus, Tarptautinė Standartizacijos Organizacija (ISO) nusprendė sukurti bendrą tinklų standartų ir metodų dokumentą. Taip atsirado atvirų sistemų sujungimo bazinis modelis (angl. ***The Basic Reference Model for Open Systems Interconnection***) arba trumpiau – **OSI modelis**. Modelis yra grįstas 7 lygiais. Jame duomenims perduoti laikomasi principo „iš apačios į viršų“. Taip pat dažnai galime sutikti ir TCP/IP modelį. Jis panašus į OSI modelį, tačiau yra suskirstytas į 4 lygius ir neturi aiškių atskyrimo taškų tarp paslaugų, sąsajų ir protokolų, bei jame laikomasi horizontaliojo požiūrio. Abiejų modelių palyginimas pavaizduotas 1 paveiksle.



1 pav. OSI ir TCP/IP modelių palyginimas

Kiekviename lygyje yra skirtingi protokolai, naudojami informacijos apdorojimui, siuntimui ir gavimui (žr. 1.1 lentelę).

1.1 lentelė. OSI ir TCP/IP modelių protokolai pagal lygius

OSI modelis		TCP/IP modelis		Protokolai
7 lygis	Taikymo	4 lygis	Taikymo	HTTP, FTP, Telnet, SMTP, DNS
6 lygis	Atvaizdavimo			IMAP, SSH
5 lygis	Sesijos			Winsock, aplikacijų programavimo sąsajos (API)
4 lygis	Transporto	3 lygis	Transporto	TCP, UDP
3 lygis	Tinklo	2 lygis	Tinklo	IP, ICMP, IPSec, IGMP
2 lygis	Kanalo	1 lygis	Ryšio	PPP, Wi-Fi, ARP, STP
1 lygis	Fizinis			RS 232, Wi-Fi

TCP/IP modelio lygiai:

Ryšio (sąsajos) lygis – išskirtinė TCP/IP steko savybė – tinklo sąsajų lygio (žemiausio lygio funkcijų) interpretavimas. Šio lygio protokolai turi garantuoti integraciją į kitų tinklų sudėtinį tinklą. TCP/IP tinklas turi turėti bet kurio kito tinklo integravimo į save priemones, nesvarbu, kokia duomenų perdavimo technologija tame tinkle būtų taikoma.

Tinklo lygis – būtent šis lygis užtikrina paketų perdavimo tinklu galimybę, naudodamas racionaliausią maršrutą. Steke pagrindiniu tinklo lygio protokolu (pagal OSI modelį) yra IP protokolas (*Internet Protocol*). Jis buvo kurtas paketinių duomenų perdavimui sudėtiniais tinklais, kuriuos sudaro daug vietinių tinklų, susijusių tiek lokaliais, tiek globaliais ryšiais. Dėl šios priežasties IP protokolas gerai veikia sudėtingos topologijos tinkluose. Kadangi IP protokolas paketus perduoda be sujungimo patvirtinimo, jis negarantuoja paketų pristatymo į paskirties vietą. Tinklo lygiui priskiriami ir visi protokolai, susiję su maršrutizacijos lentelių sudarymu ir modifikavimu, tokie kaip maršrutinės informacijos surinkimo protokolas RIP (*Routing Internet Protocol*) ir OSPF (*Open Shortest Path First*), taip pat tarptinklinių valdančiųjų pranešimų ICMP (*Internet Control Message Protocol*). Paskutinysis protokolas skirtas informacijos apie klaidas apsikeitimui tarp tinklo maršrutizatorių ir paketą siunčiančio mazgo. Specialių paketų pagalba ICMP praneša, jei paketo neįmanoma pristatyti, jei paketo gyvavimo ar surinkimo iš fragmentų trukmė būna per ilga, arba jei parametrų reikšmės tampa anomaliomis, pasikeičia persiuntimo maršrutas ir aptarnavimo tipas ir pan.

Transporto lygis – kadangi tinklo lygyje nėra sujungimo patvirtinimo, nesuteikiama jokių garantijų, kad visi paketai pasieks paskirties tašką. Patikimo duomenų perdavimo problemą sprendžia pagrindinis TCP/IP steko lygis, vadinamas transporto. Šiame lygyje veikia protokolas TCP (*Transmission Control Protocol*) ir UDP (*User Datagram Protocol*), pagal kurį sujungimo patvirtinimas nėra naudojamas. TCP protokolas garantuoja patikimą duomenų perdavimą, sudarant loginius sujungimus. Jis palaiko dvipusį duomenų perdavimo režimą bei leidžia be klaidų perduoti bitų srautą iš vieno kompiuterio į kitą. TCP srautą padalina į dalis – segmentus bei perduoda juos žemesniajam tarptinklinės sąveikos lygiui. Kai segmentai nusiunčiami į kitą kompiuterį, protokolas juos vėl paverčia nenutrūkstamu bitų srautu. Protokolas UDP garantuoja paketų perdavimą be sujungimo patvirtinimo, kuris, kaip ir protokolas IP, veikia tik kaip jungiančioji grandis tarp tinklo protokolų ir daugelio pridėtinio lygio tarnybų.

Taikymo lygis – taikomasis lygis apjungia visas tarnybas. Per daugelį naudojimo metų įvairiose šalyse bei organizacijose TCP/IP stekas sukaupė daugybę pridėtinio lygio protokolų ir tarnybų. Taikomasis lygis realizuojamas programinėmis sistemomis. Skirtingai nuo kitų trijų lygių protokolų, taikomojo lygio protokolai nesirūpina duomenų perdavimu per tinklą. Šis lygis nuolat plečiasi, kadangi prie gana seniai sukurtų tarnybų, tokių, kaip *Telnet, FTP, TFTP, DNS, SNMP* prisideda naujesnės tarnybos, pavyzdžiui *SSH*.

Józef'as Lubacz'as ir kt. steganografiją tinkle siūlo skirstyti keliais metodais[6]:

1. gali būti pagrįstas protokolo funkcijomis, susijusiomis su OSI modelio sluoksniais (žr. 1.2 lentelę);

2. gali būti pagrįstas protokolų (angl. *Protocol Data Unit*, PDU) modifikavimo tipu.

1.2 lentelė. Steganografijos tinkle metodai grįsti OSI modelio sluoksnių funkcijomis [6]

OSI modelio lygiai	Pavyzdžiai
Taikymo	HTTP antraštės manipuliacijos
Atvaizdavimo	Mažiausios reikšmės bito modifikacijos VoIP
Sesijos	SIP antraštės manipuliacijos
Transporto	Tyčinis TCP segmentų persiuntimas
Tinklo	Paketų rūšiavimas ir IP antraštės manipuliacijos
Kanalo	Tyčia sugadinti kadrai
Fizinis	Nepagrįstas OFDM simbolių didinimas bevieliame tinkle

Pagal antrą metodą dažniausiai yra išskiriami tris modifikacijų tipai:

- paketo modifikacija – nenaudojamų laukų modifikavimas:
  - duomenų dalies modifikacija;
  - protokolų laukų modifikacija;
  - maišyta modifikacija;
- paketo srauto modifikavimas:
  - pakeistas eiliškumas;
  - tyčinis paketų praradimus;
  - vėlinimų generavimas tarp paketų;
- hibridinis – paketų laukų ir siuntimo modifikavimas naudojamas kartu:
  - RSTEG metodas;

- HICCUPS metodas;

Steffen‘as Wendzel‘is ir kt. savo straipsnyje [7] siūlo slaptus kanalus klasifikuoti pagal naudojamus steganografijos būdų šablonus ir išskiria 11 kategorijų:

1. dydžio moduliacijos šablonas – slaptas kanalas naudoja PDU antraštės dydį, kad užkoduotų paslėptą pranešimą;

2. sekos šablonas – slaptas kanalas keičia PDU elementų seką, kad užkoduotų paslėptą informaciją;

   a. padėties schema – slaptas kanalas keičia nurodyto PDU elemento padėtį, kad būtų užkoduota paslėpta informacija;

   b. elementų skaičiaus šablonas – slaptas kanalas užkoduoja paslėptą informaciją pagal persiųstų PDU skaičių;

3. papildomo dubliavimo šablonas – slaptas kanalas sukuria naują erdvę duotame PDU, kur slepiami duomenys;

4. PDU iškraipymo / praradimo šablonas – slaptas kanalas generuoja sugadintus PDU, kuriuose yra paslėptas duomenų srautas, arba, generuoja paketų praradimą, kas ir yra paslėpta informacija;

5. atsitiktinės reikšmės šablonas – slaptas kanalas įterpia paslėptus duomenis į PDU antraštės elementą, kuriame yra „atsitiktinė“ reikšmė;

6. reikšmės moduliavimo šablonas – slaptas kanalas pasirenka vieną iš n reikšmių, kurios PDU antraštėje gali būti ir taip užkoduojamas paslėptas pranešimas;

   a. reikšmės šablonas – slaptas kanalas naudoja raidžių reikšmių modifikaciją PDU antraštėje koduoti paslėptus duomenis;

   b. mažiausios reikšmės bitų (ang. *Least Significant Byte*, LSB) šablonas – slaptas kanalas naudoja mažiausiai reikšmingą PDU antraštės bitą (-us), kad koduotų paslėptus duomenis;

7. rezervuoto / nenaudojamo lauko šablonas – slaptas kanalas užkoduoja paslėptus duomenis į rezervuotą ar nenaudotą PDU lauką;

8. PDU gavimo laiko schema – slaptas kanalas keičia laiko intervalus tarp tinklo PDU (tarp atvykimo laikų), kad būtų užkoduoti paslėpti duomenys;

9. srauto spartos šablonas – slapto kanalo siuntėjas keičia srauto spartą slapto kanalo gavėjui;

10. PDU eilės šablonas – slaptas kanalas koduoja duomenis naudodamas specialų PDU eiliškumą tarp slapto siuntėjo ir gavėjo;

11. pakartotinio siuntimo šablonas – slaptas kanalas pakartotinai išsiunčia anksčiau išsiųstus ar gautus PDU.

Matome, jog visos 11 kategorijų vis tiek galima suskirstyti į tris kategorijas, kaip pasiūlė J. Lubacz‘as ir kt. Kiekvienas steganografijos metodas turi savų pliusų ir savų minusų [6]. Modifikuojant PDU antraštes, persiunčiamos slaptos informacijos kiekis yra didesnis nei modifikuojant vartotojo duomenų sritį, toks modifikavimas yra nesudėtingas. Minusas yra tas, jog atsiranda galimybė prarasti PDU funkcionalumą ir yra lengvai aptinkama. Modifikuojant PDU duomenų sritį, persiunčiamų slaptų duomenų kiekis bus mažesnis nei modifikuojant PDU antraštes, yra sunkiau įgyvendinama ir aptinkama, gali suprastėti vartotojo duomenų kokybė. Naudojant PDU srauto modifikavimo metodą, slaptų duomenų persiuntimo kiekis bus mažas, reikalinga siuntėjo ir gavėjo sinchronizacija, gali atsirasti vėlinimai tačiau steganografija yra lengvai įgyvendinama ir sunkiai aptinkama. Geriausi metodai yra maišyti. Jie yra sunkiai aptinkami, nereikalauja siuntėjo-gavėjo sinchronizacijos, galima persiųsti sąlyginai didelį slaptos informacijos kiekį ir yra lengvai įgyvendinami. Vieninteliu minusu Józef‘as Lubacz‘as ir kt. įvardija tai, jog atsiranda galimybė, kad suprastės vartotojo duomenų kokybė.

Kadangi steganografijos metodai yra skirti slaptam duomenų siuntimui, tai yra rimta kibernetinio saugumo grėsmė. Todėl yra sukurta nemažai technologijų atlikti steganalizę. Steganalizę galime padalinti į dvi kategorijas – pasyvi steganalizė ir aktyvi steganalizė [8]. Pasyvi steganalizė tai metodas, kai bandoma nustatyti ar pasirinktame objekte yra paslėpta informacija. Jei yra, tuomet naudojant aktyvią steganalizę yra bandoma „ištraukti“ slaptą žinutę naudojant algoritmų atakas.

Tačiau Wojciech‘as Frączek‘as ir kt. siūlo penkias gilaus slėpimo technikas (angl. Deep Hiding Techniques, DHT) padidinančias steganografijos neaptikimo galimybes [9]:

- steganogramų išsklaidymas (angl. *Steganogram Scattering*, SGS) – kai naudojami skirtingi metodai, apimantys paskirstytos steganogramos siuntimą;
- steganogramų šokinėjimas (angl. *Steganogram Hopping*, SGH) – periodiškas steganografinio metodo keitimas vieno paslėpto ryšio metu, taip darant įtaką slapto kanalo lokalizacijai;
- nešėjo modifikacijų kamufliažas (angl. *Carrier Modifications Camouflage*, CMC), kurio tikslas - maskuoti steganogramos įterpimą į paslėptą duomenų laikmeną;
- protokolų sąveikos steganografija (angl. *Inter-Protocol Steganography*, IPS) – technika, kuri naudoja du ar daugiau skirtingus tinklo protokolus, kad įgalintų slaptą ryšį ir apsunkintų jo aptikimą;
- daugiapakopė steganografija (angl. *Multi-Level Steganography*, MLS) – metodas, leidžiantis panaudoti esamo steganografinio metodo veikimą (viršutinio lygio metodas), sukurti dar vieną (žemesnio lygio metodą). Žemutinio lygio metodas visiškai remiasi aukštesniojo lygio metodu.

1.3. STEGANOGRAFIJOS VERTINIMAS

Siekiant įvertinti steganografijos metodo veiksmingumą vienas iš būdų gali būti lyginti persiųstų „slaptų“ bitų skaičių. Tokiu atveju visus slaptus kanalus galima analizuoti pagal bendrą per sekundę perduotų steganogramos bitų skaičių (neapdorotų bitų skaičius, angl. *Raw Bit Rate* – RBR) arba pagal bendrą perduotų steganogramos bitų skaičių vienu paketu (paketo neapdorotų bitų skaičius, angl. *Packet Raw Bit Rate* – PRBR) [10]. Idealiu atveju, jei neprarandami jokie paketai, kai kurių slaptų kanalų talpa gali būti išreikšta $K = PRBR \times N$, kur N yra paketų, išsiųstų per vieną

sekundę, skaičius. Jei lyginami specifiniai slapti kanalai, pavyzdžiui naudojami VoIP, galima naudoti ir specifinius matavimus, kaip persiųstos slaptos informacijos kiekis bitais per vieną padarytą skambutį. A. Mileva ir B. Panajotov‘as sudarė palyginamąsias lenteles [11] kuriose lyginami steganografijos metodai įvairiais lygiais (žr. 1.3 lentelę):

1.3 lentelė. Steganografijos metodų palyginimas

Įrankis / Autorius	Metai	Metodas	Tipas	PRBR (bitais)
NUSHU	2004	Pradinis TCP eilės numeris	Paketo modifikavimas	32
TCP-Script	2008	TCP pliūpsniai	Paketo srauto modifikavimas	N/A
CLACK	2009	Acknowledge Sequence Number storage	Paketo modifikavimas	32
RSTEG	2010	Persiuntimai	Paketo modifikavimas	32
Autorius J. S. Thyer‘is	2008	UDP dydis ir kontrolinė suma	Paketo modifikavimas	iki 6 baitų
Eßer‘is et al	2005	HTTP atsakymo uždelsimas	Paketo srauto modifikavimas	1
HTTunnel	2005	Tarpo simbolių skaičius	Paketo modifikavimas	N/A
Iodline	2010	DNS	Paketo modifikavimas	iki 255 baitų
LACK	2008	Specialiai uždelsiamas paketų siuntimas	Paketo srauto modifikavimas	N/A
Mazurczyk‘as et al	2008	RTP laiko žymos naudojimas	Paketo modifikavimas	32

1.4. STEGANOGRAFIJA TCP/IP

Toliau pateikiant po vieną ar daugiau pavyzdžių apžvelgsime atskirai kiekvieno TCP/IP lygio steganografijos metodus.

1.4.1. Steganografija ryšio (sąsajos) lygiu

Žemiausias TCP/IP lygis yra ryšio (sąsajos) lygis. Šio lygio nėra pasiūlyta daug steganografijos metodų, tačiau vieną apžvelgsime. Yra galimybė slėpti duomenis fiziniame (PHY) 802.15.4 protokolo sluoksnyje. Šis protokolas apibūdina žemo dažnio belaidžio asmeninio tinklo veikimą (LR-WPAN). 802.15.4 protokolo MAC sluoksnio kadrai yra skirtingi ir priklauso nuo siunčiamo paketo rūšies. MAC (angl. *Media Access Contol*) sluoksnis naudoja 4-ių skirtingų tipų kadrus [12]:

1. duomenų kadras;

2. švyturio kadras;

3. patvirtinimo kadras;

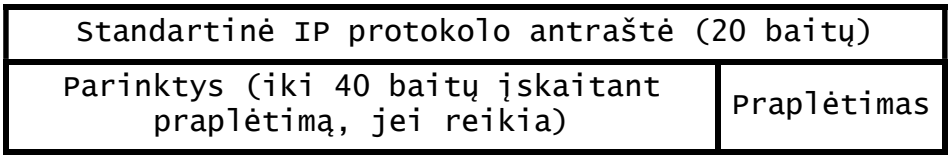
4. MAC komandų kadras.

David‘as Martins‘as ir kt. 2010 metais konferencijoje, Nicoje, pasiūlė steganografijos būdus visiems keturiems kadrų tipams. Visuose tipuose jis siūlo informaciją slėpti kadro kontroliniame, duomenų sekos numerio ir adreso informacijos laukuose. Tačiau 2015 metais IEEE patvirtino atnaujintą 802.15.4 standartą, kuriame kadrų struktūra yra pateikta kiek kitokia nei pateikė D. Martins‘as – pateikta daugiau laukų, kai kurių laukų dydžiai skiriasi, todėl negalima tvirtinti, jog šiuo metu pasiūlyti metodai vis dar veiktų.

1.4.2. Steganografija tinklo lygiu

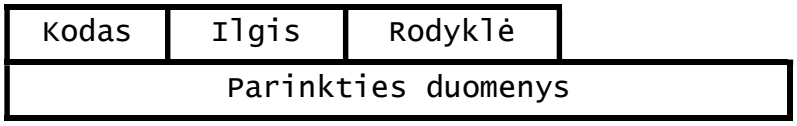
A. Mileva ir kt. straipsnyje [11] pateikia IP protokolo 4 versijos (IPv4) steganografijos metodų palyginimą, taikant paketų modifikaciją. Lyginant gerokai išsiskyrė metodas pavadintas “Paketo kelio įrašas” (angl. *Record route option*). Pagal palyginimą šis steganografijos metodas vienu PDU gali persiųsti iki 320 slaptų bitų. Metodas pasiūlytas 2007 metais [13].

4-ių bitų *HLEN* laukas nurodo IP protokolo antraštės ilgis. IP protokolo antraštės be parinkčių ilgis yra 20 baitų, tačiau su parinktimis gali būti iki 60 baitų ilgio (žr. 2 pav.).



2 pav. Standartinė IP protokolo antraštė

Svarbu paminėti, jog dauguma IP paketų internete nenaudoja parinkčių lauko. Todėl daugeliu atvejų slaptas kanalas gali naudoti nepanaudotus 40 baitų, susijusių su IP protokolo antraštės lauku „parinktis“ įterpiant paslėptą informaciją į IP paketus. IP protokolo antraštės lauko „IP parinktis“ nėra naudojamas kiekviename IP pakete. Pirmiausia įtraukiamos parinktys, skirtos tinklo testavimui ar derinimui. Parinkčių apdorojimas yra standartinė ir neatsiejama IP protokolo dalis. 3 paveiksle parodyta IP antraštės parinkties struktūra.



3 pav. IP protokolo antraštės parinkties struktūra

Lauke „Kodas“ nurodomas parinkties tipas IP protokolo antraštėje. Lauko ilgis nurodo lauko parinkties dydį. Rodyklės laukelyje atliekama tam tikra funkcija, atsižvelgiant į parinkties tipą. IP protokolo pakete yra aštuoni galimi variantų tipai. Šis steganografijos metodas naudoja „maršruto įrašo“ parinktį, kuri yra naudojama maršrutui sekti.

Šis procesas vyksta taip – parinktyje „Įrašyti maršrutą“ yra atskiras 8 bitų „Rodyklės“ laukas, kuris dedamas lauko „Duomenys“ duomenų pradžioje. Rodyklė nurodo baito vietą, kurioje turėtų būti įrašytas dabartinio maršrutizatoriaus IP adresas. Jei rodyklė yra didesnė už parinkties ilgį, daugiau vietos nėra. Jei yra pakankamai vietos, maršrutizatorius parašys savo keturių baitų IP adresą rodyklės

nurodytoje vietoje, o po to padidins rodyklę taip, kad ji parodytų į kitą poslinkį lauke „Parinkties duomenys“. Įdomu tai, kad RFC 791 teigia, jog jeigu yra šiek tiek vietos, bet neužtenka vietos visam adresui įterpti, laikoma, kad originalus paketas yra klaidingas ir atmetamas. Procesas tęsiasi tol, kol nebeliks daugiau vietos, arba kol paketas bus pristatytas į galutinę gavėjui.

Zouheir‘is Trabelsi‘is ir kt. [13] bandymo metu į ICMP paketo parinktį „Maršruto įrašas“ įterpė sakinį „*This is a covert channel*“ paversdamas jį IP adresais ir persiuntė. Kadangi žinutė užėmė tik 24 baitus, o rodyklė buvo nustatyta ties 28 baitais, bandymo metu prie slaptos žinutės maršrutizatoriai pridėjo savo adresus, kol „Maršruto įrašo“ parinkties laukas buvo visiškai užpildytas.

Didžiausias šio metodo pliusas yra didelis persiunčiamos slaptos informacijos kiekis bei slaptumas, nes žinutė yra užšifruojama IP adresais. Tačiau žinutė irgi negali būti bet kokia, kad nebūtų gautas blogas IP adresas, pavyzdžiui – *112.3.0.0* arba *0.111.123.0*.

1.4.3. Steganografija transporto lygiu

Transporto lygio pagrindiniai naudojami protokolai yra TCP („Transmission Control Protocol“) ir UDP („User Datagram Protocol). Z. Liu ir kt. [14] pasiūlė steganografijos metodą pagrįstą TCP protokolo kontroline suma. TCP protokolo laukai parodyti 4 paveiksle.

<table>
<tr><td colspan="3">Šaltinio prievadas</td><td>Paskirties prievadas</td></tr>
<tr><td colspan="4">Eilės numeris</td></tr>
<tr><td colspan="4">Patvirtinimo numeris</td></tr>
<tr><td>TCP antraštės ilgis</td><td>Išlaikymas</td><td>Žyma</td><td>Lango dydis</td></tr>
<tr><td colspan="3">Kontrolinė suma</td><td>Skubos rodyklė</td></tr>
<tr><td colspan="4">Duomenys (pasirinktinas laukas)</td></tr>
</table>

4 pav. TCP protokolo antraštės laukai

TCP kontrolinė suma yra skaičiuojama naudojant TCP antraštes, duomenis ir pseudo-antraštes. Pseudo-antraštės sudarytos iš IP protokolo antraščių: šaltinio adresas, paskirties adresas, protokolo antraštė (pvz. ar tai yra TCP ar UDP ar kitas protokolas), ilgis (TCP ar UDP ar kitas naudotas protokolas) ir rezervuoti 8 bitai. Gautos kontrolinės sumos ir antraščių suma turi būti lygi vienam. Pagrindinė pasiūlyto metodo idėja yra ta, kad kai siuntėjas siunčia duomenis gavėjui, informacija šifruota RC4 šifru atsitiktinai įterpiama į TCP kontrolės sumos antraštes lauką. Straipsnio autoriai tik 8 bitus kontrolinės sumos keičia užkoduota informacija. Tada siuntėjas naudoja pakeistą kontrolinę sumą tam, kad modifikuotų TCP duomenų lauką, kad užtikrintų kontrolinės sumos teisingumą. Šis metodas yra galimas ir UDP protokole. Šio metodo pliusais, straipsnio autoriai įvardija didesnį slaptumą, didesnį informacijos saugumą, nes informacija yra šifruojama ir didesnį saugumą nuo statistinės steganalizės.

SCTP (angl. *Stream Control Transmission Protocol*) buvo aprašytas IETF Transporto darbo grupės (SIGTRAN) 2000 m. Protokolas buvo sukurtas dėl vienos konkrečios priežasties – telefonijos signalų perdavimas per IP tinklus. Tačiau dėl savo savybių protokolas gali veikti ir kaip paprastas transporto sluoksnio protokolas. SCTP, kaip ir TCP, teikia patikimą srautą užtikrindamas sekos

eiliškumą bei „spūsčių“. SCTP taip pat leidžia nustatyti siunčiamų duomenų gavimo seką, o tai reiškia, kad duomenys yra pristatomi į viršutinį sluoksnį, kai tik jie yra gaunami.

W. Frączek‘as ir kt. dar 2011 metai aprašė [15] steganografijos metodus SCTP. Buvo aprašyti devyniolika metodų ir visiems pasiūlytos kontrpriemonės. Buvo modifikuojami pačio SCTP paketo duomenys ir bandoma modifikuoti paketų srautus. Straipsnyje rašoma: “ Visi šie metodai gali sukelti konfidencialios informacijos nutekėjimą ir turėtų būti traktuojami, kaip grėsmė tinklo saugumui. Daugelio jų galima būtų išvengti keičiant SCTP standartą – kur įmanoma, buvo pasiūlyti tam tikri patobulinimai“.

Oficialiame IETF puslapyje [16] Transporto sluoksnio darbo grupė (*TSVWG*) nurodo, jog SCTP protokolo standartas yra palaikomas ir nuolat atnaujinamas. Todėl yra galimybė, jog W. Frączek‘as. ir kt. pasiūlyti metodai šiandien nebeveikia arba gali būti atsiradę galimybių naujiems steganografijos metodams įgyvendinti.

1.4.4. Steganografija taikymo lygiu

HTTP (angl. *Hypertext Transfer Protocol*) yra turbūt populiariausias taikomojo lygio protokolas naudojamas naršant po internetinius puslapius. Tai yra dvejetainis protokolas. HTTP veikia kliento / serverio modelyje ir veikia kaip užklausos ir atsakymo protokolas. Kai klientas prisijungia prie serverio, jo naršyklė siunčia HTTP užklausą, norint pamatyti ar gauti pasirinktus duomenis. Serveris atsako HTTP atsakymu.

Dyatlov‘as ir kt. [17] siūlo slaptus kanalus, naudojant HTTP užklausos / atsakymo antraštę ir (arba) pačią HTTP užklausą / atsakymą. Pačiame protokole nėra jokio HTTP antraštės dydžio apribojimų. Bet visų HTTP antraščių dydis priklauso nuo platformos – „Apache“ serveriai priima antraštes, kurių dydis iki 8 KB, IIS iki 8 KB ar 16 KB, atsižvelgiant į versijas.

2015 m. gegužės mėn. pasirodė nauja HTTP protokolo versija HTTP/2 (IETF RFC 7540). HTTP/2 yra dvejetainis protokolas ir, palyginti su jo pirmtaku, turi daug patobulinimų ir pranašumų, tokių kaip:

- antraštės glaudinimas: HTTP antraštės dydis yra žymiai sumažintas naudojant specialius kadrus ir glaudinimą;
- tankinimas (angl. *multiplexing*): keletas HTTP užklausų gali būti siunčiamos tuo pačiu TCP ryšiu kaip atskiri srautai, o jų atsakymai gali būti gaunami iš eilės tuose pačiuose srautuose. Ši savybė pašalina kelių TCP ryšių poreikį;
- srauto priklausomybės ir prioritetai: klientas gali nurodyti serveriui, kuris iš srautų yra prioritetinis ir turi būti pirmiausiai pateiktas;
- serverio siunčiami ištekliai (angl. *server push*): jei serveris „žino“, kad tam tikri ištekliai bus reikalingi tam tikrai svetainei ir bus jų paprašyta vėliau, serveris gali siųsti šiuos išteklius be užklausos, o klientas talpins išteklius vėliau.

B. Dimitrova ir kt. [18] siūlo devynis steganografijos metodus naujojoje HTTP/2 protokolo versijoje:

- slaptas kanalas, papildant paketų antraštes. Kanalas būtų dvikryptis ir galėtų persiųsti tik 1 bitą slaptos informacijos vienu paketu;

- slaptas kanalas, naudojantis srauto identifikatorių. Kanalas galėtų persiųsti iš dalies didelį kiekį slaptos informacijos;

- slaptas kanalas, naudojantis *PING* paketą. Galima būtų siųsti iki 64 bitų slaptos informacijos vienu paketu;

- slaptas kanalas, naudojantis srauto prioritetus ir priklausomybes. Galima siųsti iki 18 bitų slaptos informacijos vienu paketu;

- slaptas kanalas, naudojantis skirtingą konkrečios rūšies kadrų skaičių. Tokiu būdu persiunčiamų slaptų duomenų kiekis priklauso nuo tam tikrų paketų skaičiaus.

- slaptas kanalas, naudojantis slapukų antraštės lauką. Kanalas gali būti tik vienpusis – į serverį, slaptos persiunčiamos informacijos dydis labai mažas – 1 bitas vienam srautui;

- slaptas kanalas, naudojantis *SETTINGS* paketus. Kanalas yra vienpusis ir vienu paketu galima persiųsti iki 5 bitų slaptos informacijos;

- slaptas kanalas, naudojantis srauto valdymą. Kanalas gali būti tik vienpusis, o slaptos informacijos kiekis priklausytų nuo srautų skaičiaus – 1 bitas vienam srautui;

- slaptas kanalas, naudojantis *HPACK* (antraštės suspaudimas). Slaptos informacijos kiekis priklauso nuo antraštės laukų – 1 bitas vienam laukui.

1.5. STEGANOGRAFIJA RTP PAKETE

Balso duomenų srauto paskutiniojo bito pakeitimas. Naudojant daugybę garso kodavimo formatų, paskutinysis bitas iš kiekvieno garso fragmento gali būti naudojamas kaip pertekliniai laikmenos elementai pranešimo duomenims įterpti. Norint iliustruoti, tarkime, kad garso failas, užkoduotas 8 bitų kodavimu, turi šiuos 8 baitus duomenų, kurie bus naudojami kaip slaptasis kanalas *0xB2 0xF5 0x8C 0xAC 0xD1 0x94 0x13 0xA8* (užkoduotas garso fragmentas), bei slapta žinutė „*A*“ lygi vienam baitui 0x41. Paverskime viską į dvejetainę sistemą:

garso fragmentas: *10110010 11110101 10001100 10101100 11010001 10010100 00010011 10101000*, o žinutė: *1000001*. Pakeičiame garso fragmento baitų paskutiniuosius bitus mūsų slaptos žinutės bitais – fragmento baitus imame nuo dešiniausio ir keliaujame į kairę. Žinutės bitus taip pat imame nuo paskutiniojo (angl. *Least Significant Byte*, LSB) paeiliui iki pirmojo. Tuomet gauname modifikuotą fragmentą: *10110011 11110100 10001100 10101100 11010000 10010100 00010010 10101001*. Šioje situacijoje matome, jog iš aštuonių baitų, reikšmes pakeitė penki, tačiau tai visuomet bus skirtinga.

W. Mazurczyk‘as ir K. Szczypiorski‘is mini [10], jog norint praktiškai įvertinti slaptų kanalų veiksmingumą, pirmiausia reikia apsibrėžti tris galimus būdus, į kuriuos reikia atsižvelgti analizuojant srautų slaptus kanalus:

- pralaidumas, kurį galima apibūdinti RBR (bendras neapdorotų bitų skaičius, angl. *Raw Bit Rate*), apibūdinantis, kiek bitų gali būti išsiųsta per vieną laiko vienetą (pvz. per 1-ą sekundę),

arba PRBR (neapdorotų bitų skaičius pakete, angl. *Packet Raw Bit Rate*), kuris nurodo, kiek informacijos bitų gali būti slapta išsiųsta viename pakete (bitai/pakete);

- bendras per pokalbį perduotų slaptų duomenų (bitų) kiekis, kuris gali būti siunčiamas viena kryptimi, naudojant taikomą metodą tipiniam VoIP skambučiui. Tai reiškia, kad, reikia sužinoti, kiek slaptos informacijos galima išsiųsti įprasto VoIP skambučio metu;

- slaptas duomenų srauto pasiskirstymas pokalbio metu - kiek duomenų buvo perduota tam tikru skambučio momentu.

Taikyti steganografiją naudojant RTP paketus gali būti labai naudinga dėl slaptos informacijos perdavimo kiekio. Tačiau kadangi RTP paketai siunčiami naudojant UDP paketus, nėra užtikrinamas slaptos informacijos vientisumas. Iš ankščiau pateiktos analizės matome, jog naudojant RTP paketo antraštės laukus galima persiųsti tikrai didelį kiekį slaptos informacijos. Tačiau tai yra atvirta informacija, lengvai aptinkama, ir perimama (nebent būtų naudojamas šifravimo mechanizmas). Taip pat norint naudoti kai kuriuos laukus reikia, kad būtų tenkinamos įvairios techninės sąlygos, kurios nedarytų įtakos atskiriems RTP paketo antraštės laukams.

ANALIZĖS IŠVADOS

Apžvelgus ir išanalizavus steganografijos būdus tinkle, o tiksliau TCP/IP šeimos protokoluose galima daryti išvadą, jog:

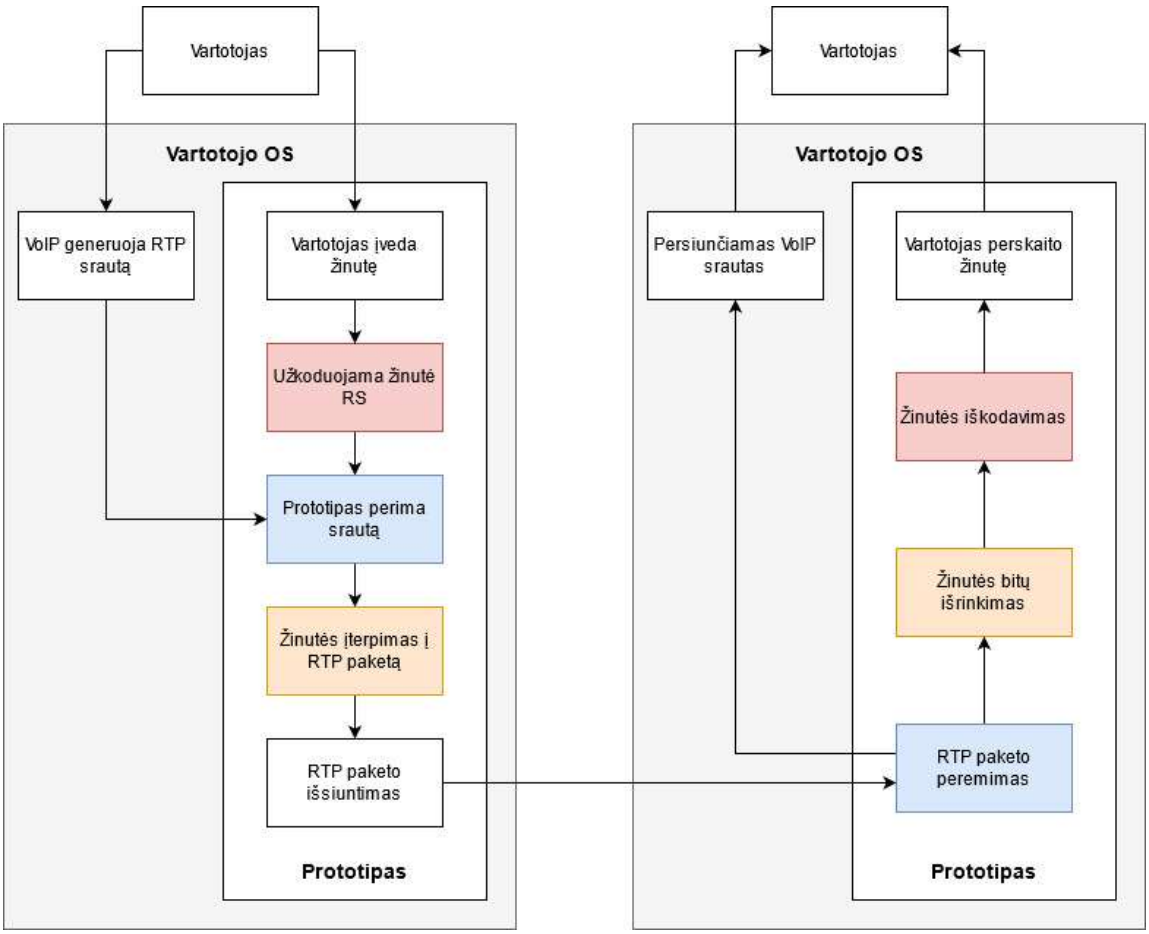
1. Steganografiją galima pritaikyti visiems TCP/IP modelio lygiams.
2. Steganografija tinklo sluoksnyje yra sunkiausiai atliekama.
3. Taikant steganografiją tinkle dažniausia problema yra persiunčiamos informacijos dydis, todėl taupant vietą duomenys yra nešifruojami.
4. Dauguma steganografijos metodų taikomų TCP/IP yra sukurti seniai, todėl:
    a. lengviau aptinkami steganalizės būdais;
    b. daugelyje protokolų išnaudotos galimybės sukurti naujiems steganografiniams metodams;
    c. pačių protokolų standartai būna atnaujinami ir pasiūlyti metodai, gali neveikti.
5. Taikant steganografiją RTP protokole, galima persiųsti didžiausią kiekį slaptos informacijos, tačiau didžiausia problema yra duomenų praradimas.

Remiantis atlikta analize ir pateiktomis išvadomis, keliamas darbo tikslas – išsiaiškinti veikiančius RTP steganografijos metodus, bei pasiūlyti patobulintą steganografijos metodą užtikrinantį persiunčiamų slaptų duomenų vientisumą, atlikti eksperimentą ir palyginti gautus rezultatus.

2. STEGANOGRAFIJOS RTP PROTOKOLE PROJEKTAS

RTP protokolas yra patogus tuo, jog galima persiųsti didelį kiekį slaptų duomenų – per vieną pokalbio VoIP programine įranga minutę vidutiniškai galima persiųsti apie 2800 paketų su paslėpta informacija. Tačiau net vienas prarastas paketas gali iškraipyti siunčiamą žinutę, ypač jei ji yra ilgesnės apimties. Patobulinto steganografijos RTP protokole metodo prototipas realizuojamas trimis pagrindinėmis dalimis (žr. 5 pav.):

1. žinutės kodavimas „Reed-Solomon“ (toliau tekste RS) kodu (raudona spalva);
2. RTP srauto perėmimu (mėlyna spalva);
3. žinutės įterpimas/išrinkimas į/iš RTP balso duomenų srautą/o (ruda spalva);



5 pav. Prototipo veikimo algoritmas

Vartotojas prototipe įves norimą nusiųsti žinutę, kuri bus koduojama RS klaidų aptikimo ir taisymo kodu - prie norimos slėpti žinutės bus pridėti papildomai baitai, gauti žinutės polinomą dalinant iš neskaidomojo polinomo. Toliau, prototipas branduolio lygmenyje perims VoIP programos generuojamą RTP srautą, kur bus galima prieš išsiunčiant paketą, jį modifikuoti. Gavėjo pusėje prototipas tiesiog skenuos visą srautą ir išrinks informaciją iš specialiai pažymėtų paketų bei iškoduos ją, kad vartotojas galėtų perskaityti.

W. Mazurczyk‘as ir kt. mini [10, 19] mini keletą steganografijos metodų naudojant RTP protokolą:

- paketo antraštės laukų keitimas;
- balso duomenų srauto modifikavimas.

Siekiant aiškiau išdėstyti paminėtus metodus, žemiau pateiktame paveiksliuke (žr. 6 pav.) yra pateikta RTP antraštės struktūra ir laukų pavadinimai.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16-31
V		**P**	**X**	**CC**				**M**	**PT**							Eilės numeris (**SN**)
Laiko žyma (**TS**)																
Sinchronizacijos šaltinio (**SSRC**) identifikatorius																
Papildomų šaltinių (**CSRC**) identifikatoriai																
Antraštės plėtinio identifikatorius																Antraštės plėtinys

6 pav. RTP antraštės struktūra

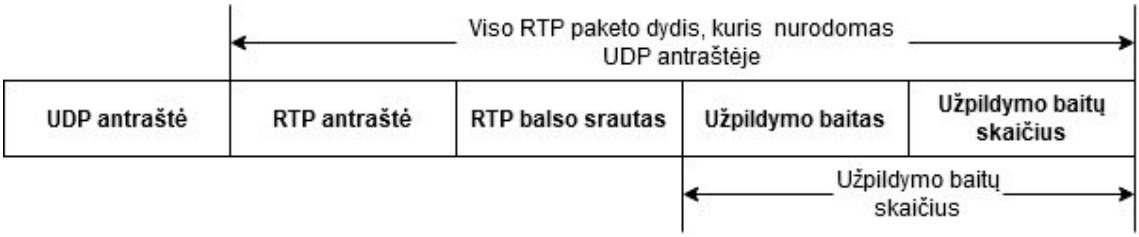
Pirmieji dvylika baitų yra kiekviename RTP pakete, tuo tarpu CSRC identifikatorių sąrašas pateikiamas tik tada, kai juos įterpia maišytuvas. Laukai turi tokias reikšmes:

- **V** (*version*) – versija (2 bitai) – šis laukas nurodo RTP versiją. Yra dvi. (reikšmė „1“ naudojama su pirmąją RTP juodraštine versija, o reikšmė „0“ naudojama protokole iš pradžių įdiegta į „vat“ garso įrankį.). Naujausia dabar naudojama versija yra 2 (antra);
- **P** (*padding*) – papildomi baitai antraštėje (1 bitas) – jei šis bitas nustatytas tuomet antraštėje sukuriami vienas ar daugiau papildomų baitų. Dažniausiai naudojama su šifravimo protokolais;
- **X** (*extension*)– plėtinys (1 bitas) – jei yra nustatytas išplėtimo bitas, prie fiksuotos antraštės PRIVALO eiti tiksliai vienas antraštės plėtinys, kurio formatas apibrėžtas RFC 3550;
- **CC** (*CSRC count*) – CSRC skaičius (4 bitai) – CSRC skaičiuje yra CSRC (papildomas šaltinis, angl. contributing source) identifikatorių, einančių po fiksuota antrašte, skaičius;
- **M** (*marker*) – žymeklis (1 bitas) – žymeklio aiškinimą nusako profilis. Skirtas tam, kad pažymėtų specifinius įvykius, pavyzdžiui srauto ribas;
- **PT** (*payload type*) – paketo nešamos garso informacijos tipas (7 bitai) – šis laukas nurodo RTP nešamos informacijos formatą ir nustato jo aiškinimą naudojant programą;
- **SN** (*sequence number*) – sekos numeris (16 bitų) – kiekvienos išsiųstos RTP duomenų paketo eilės numeris padidinamas po vieną, ir gavėjas gali jį naudoti paketo praradimo aptikimui ir paketų sekos atkūrimui. Pradinė eilės numerio reikšmė TURI būti atsitiktinė (nenuspėjama), kad apsunkintų žinomo paprasto teksto šifravimo išpuolius, net jei pats šaltinis nešifruoja;
- **TS** (*time stamp*) – laiko žyma (32 bitai) – laiko žyma pirmame srauto pakete privalo būti atsitiktinis laikas (negali būti lygus nuliui), sekančiuose paketuose jis yra didinamas paketizacijos intervalais priklausomai nuo naudojamo garso kodavimo;

- **SSRC** (*synchronization source*) – sinchronizacijos šaltinis (32 bitai) – SSRC lauke nurodomas sinchronizacijos šaltinis. Šis identifikatorius turėtų būti pasirinktas atsitiktinai, turint mintyje, kad nė vienas sinchronizacijos šaltinis toje pačioje RTP sesijoje neturės to paties SSRC identifikatoriaus;
- **CSRC** (*contributing source*) – CSRC sąrašas: nuo 0 iki 15 elementų (32 bitai) - CSRC sąraše nurodomi šiame pakete esančių garso ir/ar vaizdo srauto šaltiniai. Identifikatorių skaičius nurodomas CC laukelyje, galima nustatyti tik 15 šaltinių. Laukas yra naudojamas, kai yra vykdomas konferencinis pokalbis.

2.1. RTP PAKETO ANTRAŠTĖS LAUKŲ KEITIMAS

Paketo antraštės užpildymas. RTP paketo antraštė yra užpildoma, kai yra nustatomas antraštės užpildymo bitas (P bitas). Kai kuriems šifravimo algoritmams gali reikėti antraštės užpildymo lauko (angl. padding). Jei užpildymo nustatymo bitas yra nustatytas, paketo antraštės pabaigoje atsiranda vienas ar keli papildomi užpildymo baitai, kurie nėra balso duomenų srauto dalis (7 pav.). Duomenų, kuriuos galima pridėti po antrašte, skaičius yra apibrėžtas paskutiniame užpildymo baite, nes jame yra skaičius, kiek baitų turėtų būti nepaisoma, įskaitant ir patį paskutinįjį. Tokiu būdu, galima slėpti informaciją papildomuose baituose, tačiau reikia įsitikinti, jog papildomi baitai nebus naudojami VoIP įrangos.



7 pav. Užpildymo baitų naudojimas

Antraštės plėtinio naudojimas. Antraštės plėtinys yra naudojamas, kai yra nustatomas *X* bitas. Pagal interneto inžinerijos darbo grupės (angl. *The Internet Engineering Task Force*, IETF) RFC (angl. *Request for Comments*) 8285 standartą antraštės plėtinio metodas leidžia ne daugiau kaip vieną plėtinį vienam RTP paketui, kur yra nurodytas 16 bitų plėtinio identifikatorius ir naudojama 16 bitų nurodyti plėtinio laukų kiekį (plėtinio laukai yra 32 bitų ilgio žodžiai). Plėtinys eina iškarto po RTP antraštės. Tokie plėtiniai yra naudojami pakankamai retai, todėl galėtų būti naudojami slaptai informacijai pernešti. Tačiau, kaip ir su antraštės užpildymu, pirmiausiai reikia įsitikinti ar naudojama VoIP programa nenaudos plėtinio kokiems nors specifiniams poreikiams.

Pradinės eilės numerio ir laiko žymos laukų vertės. Šios vertės yra generuojamos atsitiktinai kiekvieno naujo VoIP ryšio metu. Eilės numeris yra 16 bitų ilgio skaičius, taigi skaičius gali būti tarp 0 ir 65535. Kiekvieno naujo VoIP ryšio metu yra sugeneruojamas atsitiktinis eilės numeris, kuris su kiekvienu nauju išsiunčiamu paketu yra padidinamas vienetu. Šioje vietoje daug duomenų nepavyks paslėpti – vieno ryšio metu tik 16 bitų.

Laiko žyma yra taip pat generuojama atsitiktinai, kiekvieno naujo VoIP ryšio metu ir didinama atitinkamai pagal pasirinktą garso kodavimo dažnį. RTP paketo laiko žyma yra 32 bitų ilgio skaitinė išraiška, todėl vieno ryšio metu galima persiųsti tik 32 bitus slaptos informacijos.

Sinchronizacijos šaltinio identifikatoriaus naudojimas. Sinchronizavimo šaltinio (SSRC) identifikatorius yra 32 bitų skaičius, unikaliai identifikuojantis vieną RTP srauto šaltinį. Konkrečioje daugialypės terpės konferencijoje kiekvienam siuntėjui yra parenkamas atsitiktinis SSRC ir tikimasi, kad taip bus išspręsti konfliktai, kai du šaltiniai pasirenka tą pačią vertę. Kadangi šis kodas nesikeičia viso užmegzto ryšio metu, todėl galima persiųsti tik 32 bitus slaptos informacijos.

Papildomų šaltinių (CSRC) identifikatoriai. CSRC sąraše nurodomi pakete esantys balso duomenų srauto šaltiniai. Identifikatorių skaičių nurodo RTP paketo antraštės CC laukas. Kadangi CC laukas yra 4 bitų, galima nustatyti tik 15 šaltinių. CSRC identifikatoriai įterpiami VoIP maišikliais, naudojant SSRC identifikatorius. CSRS identifikatorius yra 32 bitų ilgio. Taigi jei nėra naudojamas konferencinis pokalbis vienu RTP paketu galima persiųsti maksimaliai 480 bitų informacijos. Svarbu šioje vietoje, kad visas paketas neviršytų nustatyto tinklo maksimalaus perdavimo vieneto (angl. M*aximum Transmission Unit*, (MTU)).

Iki šiol, įvairiuose šaltiniuose dažniausiai aprašomas steganografijos metodas VoIP sistemose yra paskutiniojo bito keitimo metodas [10]. Naudojant šį metodą per 1 minutę (priklausomai nuo tinklo galimybių) galima persiųsti vidutiniškai 350 ženklų žinutę. Tačiau dėl interneto ryšio problemų ar kitų trikdžių paketai gali būti prarandami. Praradus bent vieną paketą, žinutės nebūtų įmanoma perskaityti. Dėl šios priežasties, vienas iš siūlomų metodo patobulinimų yra naudoti klaidų aptikimo ir taisymo kodą. Tačiau, vienas iš patobulinimo minusų yra pernešamos slaptos informacijos kiekio sumažėjimas.

2.2. RTP PAKETO BALSO DUOMENŲ SRAUTO MODIFIKAVIMAS

Balso duomenis persiunčiant internetu, jie dažniausiai yra koduojami G.711 kodeku, naudojant *μ-law* (buvo kuriamas JAV ir Japonijos rinkoms) arba *A-law* (buvo kuriamas likusiam pasauliui) formatus. Tai yra ITU (Tarptautinės telekomunikacijų sąjungos) standarte apibrėžtas kodekas. Vienu RTP paketu persiunčiamas balso duomenų kiekis, naudojant populiariausią G.711 kodeką, svyruoja tarp 40 ir 320 baitų (populiariausias ir dažniausiai naudojamas dydis yra 160 baitų). W. Mazurczyk‘as ir K. Szczypiorski‘is siūlo modifikuoti tik paskutinįjį balso duomenų bitą, kaip „nereikalingą“ [10].

m-law ir A-law							
s	0	0	0	a	b	c	d
s	0	0	1	a	b	c	d
s	0	1	0	a	b	c	d
s	0	1	1	a	b	c	d
s	1	0	0	a	b	c	d
s	1	0	1	a	b	c	d
s	1	1	0	a	b	c	d
s	1	1	1	a	b	c	d

8 pav. Užkoduoto balso bitai

Tačiau pagal ITU-T „*Neprarandamo G.711 impulso kodo suspaudimo moduliacijos rekomendacijas*“, tik pirmas bitas yra informacinis, likę yra duomenų. Todėl kitas steganografinio metodo patobulinimas būtų tas, jog bus galima naudoti ne tik paskutinįjį balso duomenų bitą, bet pasirinkti vieną iš paskutinių keturių, siunčiant slaptą informaciją.

Pagal N. Aoki [20] G.711 užkoduoti balso bitai yra pateikti aukščiau (žr. 8 pav.). Kairiausias bitas (8-tas skaičiuojant nuo dešinės pusės) nurodo ženklą – kuriame diapazone (teigiamame ar neigiamame) yra garsas, 7-5 bitai nurodo žingsnį X ašyje, 4-1 bitai nurodo gamos akordą (poslinkį Y ašyje). Taigi iš esmės keičiant paskutinįjį bitą, yra kažkiek koreguojamas balso kodavimas ir teoriškai nepadarant didesnės žalos turėtų būti galimybė keisti bet kurio balso duomenų srauto okteto vieną iš paskutinių keturių bitų.

Kadangi paketo antraštės laukų modifikavimas labai priklauso nuo galimai naudojamos VoIP įrangos, prototipas bus realizuojamas modifikuojant balso duomenų srauto vieną iš paskutinių 4-ių bitų. Tačiau, siekiant išsiaiškinti poveikį balso kokybei, bus atliekami bandymai panaudojant ir kitus okteto bitus.

2.3.KLAIDU APTIKIMO IR TAISYMO KODAI

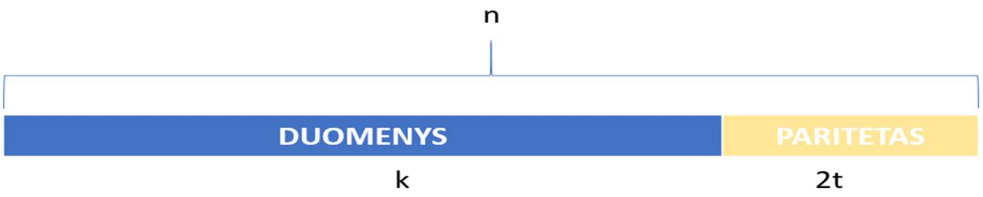
Kadangi RTP paketa siunčiami tinkle dažniausiai naudojant UDP srautą (yra galimybė naudoti TCP srautą), dėl įvairaus „triukšmo“ tinkle arba dėl blogo ryšio kanalo, galimi paketų iškraipymai – paketo bitas ar net keli bitai yra „pakeičiami“ ir gaunami klaidingi paketai arba galimi paketų praradimai – kai vienas ar daugiau paketo bitų tiesiog nėra persiųsti. Dėl šios priežasties yra sukurta nemažai klaidų aptikimo ir taisymo kodų. Populiariausi ir, matyt, geriausiai žinomi klaidų aptikimo kodai yra pariteto bito (arba bitų lyginumo) naudojimas, „pakartojimo kodas“, kontrolinė sumos naudojimas, maišos funkcija. Kontrolinė suma yra naudojama ir mums labai gerai pažįstamuose TCP bei UDP protokoluose. Geriausiai žinomi klaidų taisymo kodai yra „Hammingo“ kodas, cikliniai BCH kodai, tiesiniai blokiniai kodai, FEC kodas (angl. *Forward error correction*), bei „Reed-Solomon“ kodas.

Klaidų aptikimo ir taisymo kodų veikimas yra pagrįstas papildomos informacijos pridėjimu prie siunčiamos informacijos. Pavyzdžiui naudojant lyginumo kodą (siekiant aptikti vieno bito klaidą) prie siunčiamos informacijos yra pridedamas vienas papildomas bitas, kuris nurodo kiek siunčiamoje informacijoje yra bitų su reikšme „1”. Paimkime raidę „A“, kurios dvejetainė išraiška yra *01000001*. Kadangi gautoje išraiškoje turime du bitus, kurie lygūs „1”, naudojant lyginumo bito kodą prie turimos reikšmės pridedame papildomai „0“. Jei „1“ skaičius būtų nelyginis, informacijos pabaigoje turėtume pridėti „1”, kad vienetų skaičius būtų lyginis. Tačiau naudojant šį kodą mes galime tik aptikti, jog yra klaida, tačiau negalime nustatyti, kur klaida yra bei jos ištaisyti.

Siekiant ištaisyti klaidas galime naudoti „Hamming“ kodą. Tai yra tiesinis kodas, kuris gali aptikti iki dviejų bitų klaidas ir ištaisyti vieno bito klaidas. Tam naudoja bitų lyginumą. Pavyzdžiui norint persiųsti raidę „C“, kurios dvejetainė išraiška yra *01000011*. Naudojant „Hamming“ kodo algoritmą gauname dvylikos bitų žodį – prie turimų aštuonių bitų informacijos turime pridėti keturis lyginumo bitus. Kiekvieno lyginumo bito eilės numeris parodo, kiek gautos informacijos eilutės bitų, kurių reikšmė „1“ yra lyginama ir kiek praleidžiama. Lyginumo bitų eilės numeriai visada yra 2n , kur n yra sveikasis natūralusis skaičius.

Tačiau, jei reikia ištaisyti didesnį kiekį klaidų, reikėtų naudoti „Reed-Solomon“ (toliau RS) klaidų taisymo kodą. „Reed-Solomon“ algoritmas yra plačiai naudojamas telekomunikacijose ir duomenų laikmenose. Tai yra visų CD ir DVD skaitytuvų, WiMAX ar DSL technologijų ir net daugelio brūkšninių kodų dalis. RS yra blokinis algoritmas, nes apdoroja pranešimo duomenis kaip atskirus blokus. Tai pat yra ir polinominis algoritmas, nes jis naudoja modulinius polinomus kodavimo ir dekodavimo procesuose. RS ima skaitmeninių duomenų bloką ir prideda papildomų „atsarginių“ bitų. RS dekoderis apdoroja kiekvieną bloką ir jei yra randamos klaidos bando jas ištaisyti. Kokį kiekį ir kokio tipo klaidų kodas galės ištaisyti priklauso nuo RS kodo savybių. Jei, sakykime, norime išsiųsti *k* bitų teksto pranešimą, RS atsiųs $n = k + 2s$ ilgio pranešimą ir garantuos, kad teisingą pranešimą galima atkurti kitame gale, jei yra mažiau sugadintų bitų nei s. Dažniausiai naudojamų parametrų pavyzdys: $k = 223$, $s = 16$, $n = k + 2s = 255$, suteikiant galimybę ištaisyti 16 sugadintų skaitmenų iš kiekvieno 255 ilgio bloko.

Tai reiškia, kad šifravimo įrenginys paima k bitų duomenų simbolius ir prideda pariteto simbolius, kad sudarytų n simbolio kodinį žodį. Yra n-k lyginumo simboliai iš kiekvieno s bito. RS dekoderis gali ištaisyti iki t simbolių, kuriuose yra kodinio žodžio klaidų, kai $2t = n\text{-}k$ (žr. 9 pav.). Šioje diagramoje parodytas tipiškas „Reed-Solomon“ kodo žodis (tai vadinama sisteminiu kodu, nes duomenys paliekami nepakeisti, o pariteto simboliai pridedami):

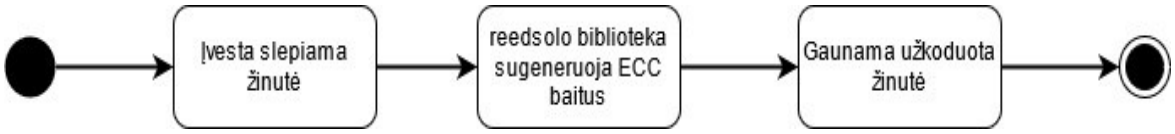


9 pav. RS kodinio žodžio struktūra

Polinomų naudojimas yra pagrindinė klaidų taisymo kodų, tokių kaip „Reed-Solomon“, priežastis: užuot tik turėdami pranešimą kaip (ASCII) skaičių seriją, mes jį matome kaip polinomą, kuris yra apibrėžtas baigtinio lauko aritmetikos taisyklėmis. Kitaip tariant, vaizduodami duomenis naudojant polinomus ir baigtinių laukų aritmetiką, prie duomenų pridėjome struktūrą (klaidų aptikimo ir taisymo baitus). Pranešimo vertės liko tos pačios, tačiau ši nauja struktūra dabar leidžia mums valdyti pranešimą, jo simbolių reikšmes, naudojant gerai apibrėžtas matematines taisykles. Ši struktūra yra visada išorėje ir nepriklausoma nuo duomenų, ir leidžia mums aptikti klaidas ir pataisyti sugadintą pranešimą.

Pagal standartinį „Reed-Solomon“ veikimą, matome, jog atsižvelgiant į tai, kiek maksimaliai klaidų norime ištaisyti, priklausys nuo klaidų aptikimo ir taisymo baitų (angl. *Error Count and Correction*, ECC) skaičius. Tarkime, jog slepiamos žinutės ilgis yra 10 baitų ir norime, kad būtų galimybė ištaisyti 5 klaidas. Tokiu atveju, prie slepiamos žinutės reikia pridėti papildomus 10 pariteto baitų, dėl ko gauname iš viso 20 baitų dydžio slepiamą žinutę. Tačiau, jei yra galimybė nurodyti klaidų pozicijas žinutėje, RS kodas tuomet gali ištaisyti tiek klaidų, kiek yra nustatytą klaidų aptikimo ir taisymo baitų. Tam tikslui, gali pasitarnauti RTP paketo antraštės eilės numerio laukas. Kadangi viso ryšio metu, visi RTP paketai yra numeruojami, yra galimybė sekti prarastus paketus, kartu ir nustatyti prarastų bitų vietas žinutėje. Bet nebus galimybės nurodyti klaidų pozicijų žinutėje, jei jos

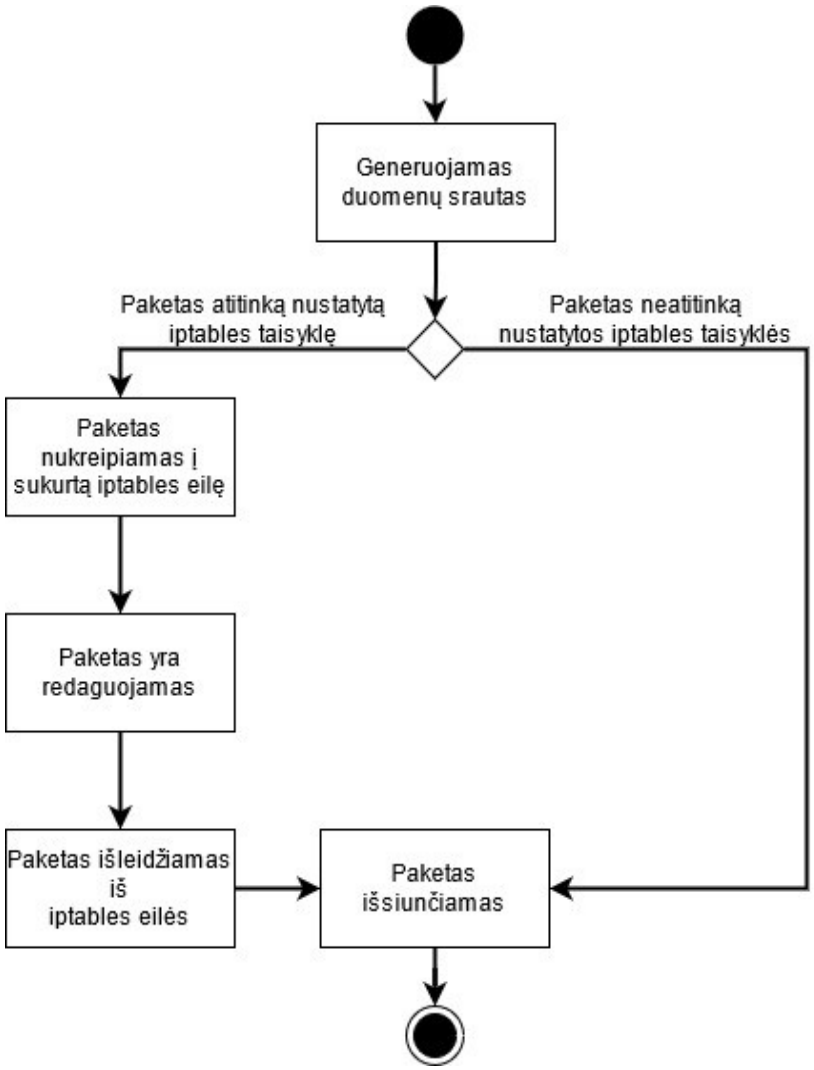
atsiras dėl balso duomenų pakitimų atsiradusių blogoje ryšio terpėje. Supaprastintas bibliotekos veikimo algoritmas pavaizduotas 10 paveiksliuke.



10 pav. Žinutės kodavimas RS kodu

2.4. DUOMENŲ SRAUTO PERĖMIMAS

Siekiant modifikuoti RTP paketus reikia taip pat kontroliuoti ir VoIP programinės įrangos generuojamą srautą. Tai patogiausia atlikti ‚Linux“ operacinėje sistemoje pasinaudojant „iptables“ ugniasiene.



11 pav. RTP srauto perėmimas

Prototipas galės manipuliuoti „iptables“ ugniasiene, taip kontroliuodamas RTP tiek išsiunčiamą, tiek gaunamą srautus. Siekiant perimti programų generuojamą srautą tame pačiame kompiuteryje neišeina paprastai sukurti jungties (angl. *socket*). Dėl šios priežasties kuriamame prototipe bus naudojama „Python“ biblioteka „Netfilterqueue“, kuri yra paremta „libnetfilter_queue“

(„netfilter“ projekto dalis). Būtent ši biblioteka yra skirta naudoti „Linux“ sistemose. Ji gali valdyti kompiuteryje generuojamus arba gaunamus tinklo srautus bei manipuliuoti jais pasitelkiant „iptables“. „iptables“ yra vartotojo erdvės pagalbinė programa, leidžianti sistemos administratoriui sukonfigūruoti „Linux“ branduolio ugniasienės IP paketų filtravimo taisykles.

Prototipe bus nurodytos sąlygos, kurios leis perimti tame pačiame kompiuteryje generuojamą RTP srautą. Perimtas paketas bus modifikuojamas ir perduodamas atgal „iptables“, kad būtų išsiųstas toliau.

2.5. SLAPTŲ DUOMENŲ ĮTERPIMAS

Trečia prototipo sudedamoji dalis yra pačios žinutės įterpimas į RTP srautą. Kadangi tai yra patobulintas LSB metodas, kai siunčiama informaciją yra įterpiama į RTP paketo balso srauto duomenų lauką. Kiekvienas žinutės bitas yra įterpiamas vietoj vieno RTP paketo balso duomenų lauko bito. Pirmiausiai prisiminkime RTP paketo antraštės struktūrą (žr. 12 pav.).

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td colspan="16">16-31</td></tr>
<tr><td colspan="2">V</td><td>P</td><td>X</td><td colspan="4">CC</td><td>M</td><td colspan="7">PT</td><td colspan="16">Eilės numeris (SN)</td></tr>
<tr><td colspan="32">„Laiko žyma (TS)</td></tr>
<tr><td colspan="32">Sinchronizacijos šaltinio (SSRC) identifikatorius</td></tr>
<tr><td colspan="32">Papildomų šaltinių (CSRC) identifikatoriai</td></tr>
<tr><td colspan="16">Antraštės plėtinio identifikatorius</td><td colspan="16">Antraštės plėtinys</td></tr>
</table>

12 pav. RTP paketo antraštė

Prototipas naudos „žymeklio“ bitą (1 bitas) siekiant pažymėti paketus, kuriuose bus slepiama žinutė (žr. 12 pav.). Kadangi kuriamas prototipas manipuliuoja RTP srautu tarp VoIP aplikacijų, todėl galime modifikuoti antraštės bitus, pagal poreikį. Siekiant nustatyti žinutės pabaigą bus modifikuojamas dar vienas antraštės laukas – „plėtinio“ („P“ bitas, žr. 12 pav.).

Dar viena galima prototipo modifikaciją galėtų būti paketo antraštės CC lauko (susijusių šaltinių skaičiaus) keitimas. Laukas yra keturių bitų ilgio. Pavyzdžiui, naudojant CC lauką, prototipo vartotojas galėtų nurodyti keičiamo balso duomenų bito numerį. Sakykime, jei yra norima keisti patį paskutinį bitą, CC laukas galėtų turėti reikšmę *0000*, jei antrą nuo galo bitą, tuomet reikšmė galėtų būti *0010*. Bet reikėtų atsargiau naudoti šį lauką, nes konferenciniame pokalbyje, jis galėtų būti naudojamas.

Steganografijos tinkle veiksmingumą yra patogiausia vertinti PRBR (slaptos informacijos kiekis viename duomenų pakete) arba RBR (slaptos informacijos kiekis persiunčiamas per vieną sekundę). Siūlomo metodo PRBR yra 1 bitas, kas yra mažiausia galima reikšmė steganografijoje tinkle, tačiau RBR yra vidutiniškai 46 bitai/s (priklauso nuo interneto greičio). A. Mileva ir B. Potavo‘as palygino steganografijos našumą PRBR kiekiu įvairiuose protokoluose [11]. Lentelėje (žr. **NR.** lentelę) matome, jog PRBR įvairiuose protokoluose yra labai skirtingas. Tačiau prie kiekvieno iš metodų šalia privalumo, turi išskirtą ir minusą. Planuojamas metodas, neturi nei vieno lentelėje paminėto minuso. Stengiantis padidinti PRBR kiekį, nuspręsta pabandyti pritaikyti ir kitą metodo patobulinimą – naudoti du paskutinius garso duomenų okteto bitus slaptai informacijai siųsti. VoIP

generuoja milžiniškus RTP srautus, todėl padidinus PRBR tik vienu bitu, bus jaučiamas žymus persiunčiamos slaptos informacijos padidėjimas per sekundę.

2.1 pav. Steganografijos našumo palyginimas

Metodas	Protokolas	PRBR (Bitais)	Privalumai	Trūkumai
Antraštės kontrolinė suma	IP	16	Galima pritaikyti IP, TCP, UDP	Lengva eliminuoti
TTL naudojimas	IP	1	Sunkus aptikimas	Nenormalus TTL
ICMP atsakymai	ICMP	24-56	IP tunelis per ICMP	Lengva eliminuoti
TCP laiko žymos	TCP	1	Nereikia sinchronizacijos	Kai kurios žymos gali būti praleistos
Antraštės kontrolinė suma	TCP	16	Galima pritaikyti IP, TCP, UDP	Reikia žinoti originalų TTL
Kontrolinės sumos naudojimas	UDP	Iki 6	Lengva pritaikyti	Lengva eliminuoti
„GET“ užklausos naudojimas	HTTP	Kintantis	Gali „apeiti“ ugniasienes	Lengva aptikti, jei informacija slepiama „GET“ karkase
„TXT“ įrašų naudojimas	DNS	Iki 255	IPv4 per DNS	Dalina IP paketus
Neigiamas podėlis	DNS	1	Nereikia turėti DNS serverio	Reikia pasirinkti gerus žemesnio lygio domenus
Plėtinio lauko naudojimas	RTP	8	-	Lengva eliminuoti
Laiko žymos lauko naudojimas	RTP	32	Sunkus aptikimas	Veikia tik su vieninteliu (pirmu) pokalbio paketu

Šios prototipo dalies sėkmingas įgyvendinimas priklauso nuo VoIP paslaugos tiekėjo techninės infrastruktūros. Buvo tikrinami keli VoIP paslaugų tiekėjai – „jitsi“ (adr. https://meet.jit.si/), „Google Meet“ (adr. https://meet.google.com/) ir „Linphone“. Vienintelis VoIP paslaugos teikėjas pilnai atitinkantis keliamus reikalavimus prototipui yra „Linphone“. Nes, panašu, jog likę tiekėjai, naudoja kažkokias saugumo priemones – nėra leidžiama pakeisti RTP paketo

versijos. Jei yra modifikuojamas papildomų šaltinių kiekio laukas (antraštės CC laukas), paketas „Wireshark“ programoje yra pažymimas, kaip sugadintas. Sekantis pastebėtas trūkumas naudojantis „Google Meets“ yra dažnas „žymeklio“ lauko naudojimas. Dėl to būtų nuskaitomi ne tik slepiamos žinutės bitai, bet ir su žinute nesusiję bitai.

2.6. ŽINUTĖS IŠTRAUKIMAS IŠ RTP SRAUTO

Prototipas gautų RTP paketų, kurių žymeklio bitai bus nustatyti, nuskaitys naudingosios įkrovos pasirinktus bitus ir taip atliks žinutės rekonstrukciją. Siunčiamos žinutės pabaiga bus pažymėta nustatytu plėtinio bitu. Sakykime, jog žinutės ilgis yra iki 256 ženklų įskaitant ECC baitus. Vienam ženklui sudaryti yra naudojami 8 bitai, taigi gauname, kad siunčiant pilną žinutę reikės 2048 paketų. Iš visos pilnos žinutes, priklausomai, nuo to kur yra praradimai, prototipas „gali prarasti“ (jei ECC skaičius yra 10) nuo dešimties iki aštuoniasdešimties paketų, kas yra 0.5% – 3.9% visos žinutės paketų skaičiaus.

Prototipas nuskaitys ir į dėklą kaups žinutės paketų eilės numerius. Eilės numeriai bus skirstomi į grupes po aštuonis – kiekvienam žinutės ženklui. Tokiu būdu bus galima sužinoti, kuriuose ženkluose ir po kiek buvo prarasta paketų.

13 pav. Žinutės išgavimo iš RTP procesas

2.7. VOIP SISTEMA

VoIP sistema bus realizuojama naudojantis „Linphone“ VoIP programine įranga. Naudojantis „Linphone“ yra galimybė naudoti tiek realų įmonės siūlomą tarpinį SIP serverį, tiek galimas tiesioginis ryšys tarp įrenginių. Sistemoje bus naudojamas *A-law* kodekas, kurio skaitmeninis dažnis yra 8000 Hz. Naudojant tiesioginį ryšį bus paprasčiau imituoti ir valdyti paketų praradimų kiekius. Srauto paketų praradimai bus imituojami naudojant „Linux“ tinklo emuliatorių „netem“.

PROJEKTO IŠVADOS

Šiame skyriuje buvo apžvelgta kuriamo prototipo projektas. Tai yra patobulintas RTP paketo balso duomenų srauto pasirinkto bito keitimo metodas. Šis metodas naudoja žinutės kodavimą RS, kad įvykus paketų praradimui būtų galima atstatyti siunčiamą žinutę. Naudojant šį metodą, yra keletas apribojimų:

- ištaisomų klaidų kiekis – standartiškai yra puse ECC baitų skaičiaus. Jei yra žinomos klaidų pozicijos galima ištaisyti tiek klaidų, kiek yra ECC baitų. Siekiant padidinti ištaisomų klaidų kiekį bent vienu, prie žinutės automatiškai yra pridedami du ECC baitai;
- naudojamų ženklų apribojimas – galima naudoti tik UTF-8 koduotės simbolius, nes su kitomis koduotėmis neveikia „reedsolo“ biblioteka;
- VoIP paslaugos tiekėjas neturėtų naudoti žymeklio bito, bei turėtų ignoruoti kitus paketų antraštės modifikavimus. Naudojant šį metodą bus modifikuojamas ne tik RTP paketo

naudingosios įkrovos paskutinysis bitas, bet ir pačio paketo antraštės žymeklio, bei plėtinio laukai. Šis modifikavimas yra būtinas siekiant nustatyti paketus kuriuose yra siunčiama žinutė bei siunčiamos žinutės pabaigą.

Optimalus žinutės ilgis ir ECC baitų kiekis paaiškės atlikus prototipo bandymus.

3. STEGANOGRAFIJOS RTP PROTOKOLE PROTOTIPO REALIZACIJA

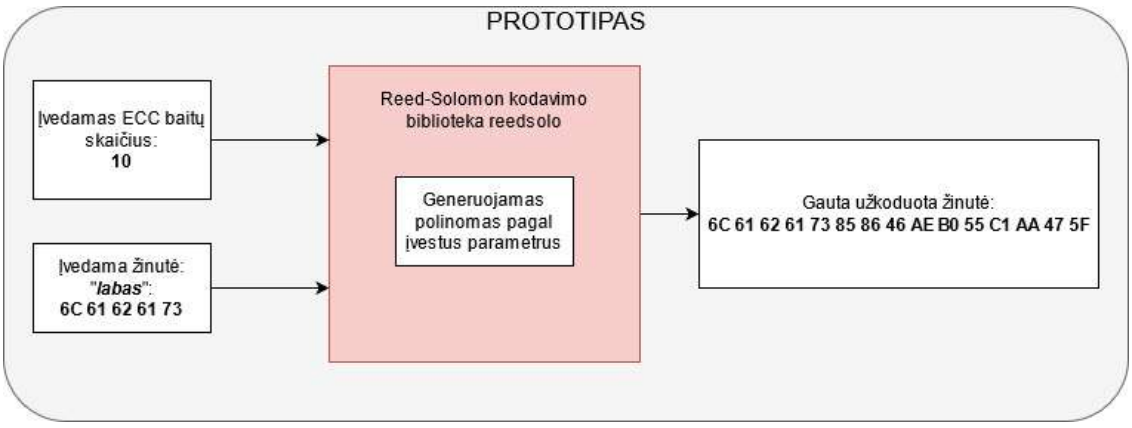
Visas prototipas yra realizuotas virtualioje mašinoje naudojant „VirtualBox“ programą (6.1 versija) ir „Linux Ubuntu“ (20.04 versija) operacinę sistemą. Šis pasirinkimas buvo pagrinde pasirinktas dėl naudojamos NFQ bibliotekos, kuri sukurta valdyti „iptables“ („Linux“ sistemos ugniasienės) srautus. Pats prototipas yra programuojamas „Python“ programavimo kalba (3.8 versija). Prototipui bandyti yra naudojama „Linphone“ VoIP programa (3.12.0 versija). „Linphone“ pasirinkta todėl, kad siūlo nemokamą savo SIP tarpinį serverį, kur prototipą galima išbandyti realiomis sąlygomis. Taip pat siekiant imituoti tam tikrą atsitiktinių paketų praradimą bus naudojamas „Linux“ įrankis „netem“, kuriuo galima imituoti norimą kiekį duomenų paketų praradimų.

3.1. ŽINUTĖS KODAVIMAS RS BIBLIOTEKA

Šioje dalyje prototipas užkoduoja vartotojo įvestą žinutę RS kodu. Kaip ir kiti BCH kodai, „Reed-Solomon“ kodai yra koduojami padalijant pranešimą reprezentuojantį polinomą iš neskaidomo polinomo. Taip yra sugeneruojami ir prie žinutės pridedami papildomi, klaidų aptikimo ir taisymo (toliau ECC).

Šiai daliai įgyvendinti prototipe yra naudojama „Python“ biblioteka „reedsolo“ (1.5.4 versija).Vartotojo įvesta žinutė yra koduojama UTF-8 koduote ir paverčiama baitais. Tuomet ji yra koduojama „reedsolo“ biblioteka. Pirmi du veiksmai atliekami privalomai, kad „reedsolo“ biblioteka galėtų užkoduoti žinutę. Galiausiai, gautas rezultatas yra verčiamas į dvejetainę sistemą, kur vienas ženklas yra atvaizduojamas 8-iais bitais ir viskas yra sukeliama į dėklą.

Naudojant pavyzdinius nustatymus, kai ECC yra 10 baitų biblioteka gali aptikti ir ištaisyti iki 5 klaidų. Aptinkamų klaidų skaičių galima didinti keičiant bibliotekos ECC baitų skaičių. Norint aptikti papildomai vieną klaidą žinutėje, biblioteka prideda papildomus du ECC baitus. Jei bibliotekai yra pateikiamos klaidų vietos žinutėje – pateikiamos pozicijos, kur ženklas buvo pakeistas ar prarastas, tuomet ji gali ištaisyti tokį klaidų kiekį, kiek ECC baitų buvo pridėta.



14 pav. Žinutės kodavimas RS kodu

Aukščiau pateiktame paveiksliuke (14 pav.) pademonstruota, kaip biblioteka veikia. Yra nustatomas ECC baitų skaičius, šiuo atveju – 10. Įvedama žinutė „labas“ (kuri šešioliktainiu pavidalu atrodo taip – *6C 61 62 61 73*). Po užkodavimo gauname tokią žinutę – *6C 61 62 61 73 85 86 46 AE B0 55 C1 AA 47 5F*. Tai prie žodžio „labas“ pridėti 10 ECC baitų.

Naudojama biblioteka pagal nutylėjimą koduoja žinutes, kurių ilgis yra 256 ženklų įskaitant ECC baitus. Esant reikalui koduoti ilgesnes žinutes, biblioteka pati suskaido žinutę į reikiamo ilgio dalis ir taip prideda reikiamą kiekį ECC baitų.

Biblioteka turi vieną apribojimą – koduoja tik UTF-8 koduotės ženklus. Bandant naudoti kitos koduotės ženklus, neišeis perskaityti žinutės. Todėl atliekant eksperimentus nebus galima naudoti specialių ženklų ar lietuviškų rašmenų. Žemiau, 15 paveiksliuke yra pateikiama, kaip atrodo persiųsta „*ąčęėįšųūž*“ žinutė.

```
Po persiuntimo prototipo rastu klaidu pozicijos:  []
Gauta duomenu paketu: 264

reedsolo rastu klaidu pozicijos:  []
Gauta zinute:  \xff\xfe\x05\x01\r\x01\x19\x01\x17\x01/\x01a\x01s\x01k\x01~\x01
```

15 pav. Lietuviškų rašmenų atvaizdavimas

```
bytearray(b'magistro baigiamasis darbas<\x01\xb7c\xa3\xd0\xf1\x0f\xb7\xdb\xea4\xa9')
```

16 pav. Žinutė užkoduota su „reedsolo“ biblioteka

```
Po persiuntimo prototipo rastu klaidu pozicijos:  [3, 12, 15, 17, 30, 31, 37]
Gauta duomenu paketu: 313

reedsolo rastu klaidu pozicijos:  [3, 12, 15, 17, 30, 31, 37, 29, 16, 2]
Gauta zinute:  magistro baigiamasis darbas
```

17 pav. „reedsolo“ bibliotekos veikimas prototipe

16 paveiksle matome, kaip atrodo su RS užkoduota žinutė. Tai yra sakinys „magistro baigiamasis darbas“ ir pridėti 13 ECC baitų. 17 paveiksle pateiktas pritaikytas prototipui „reedsolo“ veikimas. Pirmoje eilutėje matomas prototipo aptiktų klaidų pozicijas visoje žinutėje, t. y. kur buvo prarasti paketai. Antroje eilutėje yra pateikiamas duomenų paketų skaičius, kuris buvo gautas siunčiant žinutę. Šis skaičius yra gaunamas sekant gautus paketus su nustatytu žymeklio bitu ir tikrinant sekos eiliškumą. RTP paketai gali būti persiunčiami, kad būtų kuo mažesnis balso duomenų praradimas, tačiau prototipas tokį (persiųstą paketą) laikys klaidingu, nes jis neatitiks sekos eiliškumo. Toliau yra pateikiamos visos žinutės (įskaitant ir ECC baitus) klaidų skaičius, kurį aptiko „reedsolo“ biblioteka (dažnai ši informacija ir prototipo skaičiuojamos klaidos nesutampa). Paskutinėje eilutėje yra rodoma tik ištaisyta prototipo apdorota žinutė. 18 paveiksle pavaizduota, kai prototipo aptiktos klaidos neatitinka „reedsolo“ aptiktų klaidų.

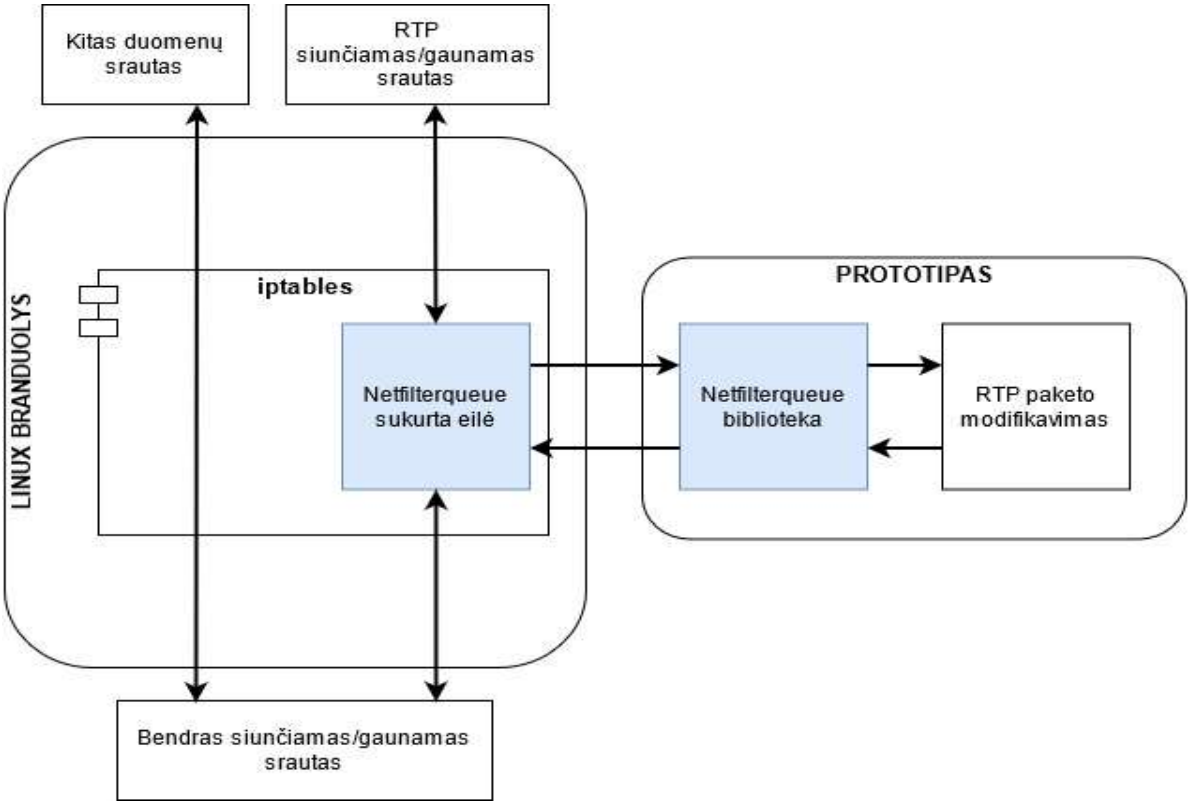
```
Po persiuntimo prototipo rastu klaidu pozicijos:  [3, 8, 14, 28]
Gauta duomenu paketu: 260

reedsolo rastu klaidu pozicijos:  [3, 8, 14, 28, 27, 13, 7]
Gauta zinute:  prototipo tyrimo pvz
```

18 pav. Klaidų pozicijų neatitikimas

3.2. RTP SRAUTO PERĖMIMAS

Kita prototipo sudedamoji dalis yra RTP srauto perėmimas. Siekiant perimti programų generuojamą srautą tame pačiame kompiuteryje neišeina paprastai sukurti jungties (angl. socket). Dėl šios priežasties kuriamame prototipe bus naudojama „Python“ biblioteka NFQ. 19 paveiksliuke yra pavaizduotas srauto perėmimo veikimas.

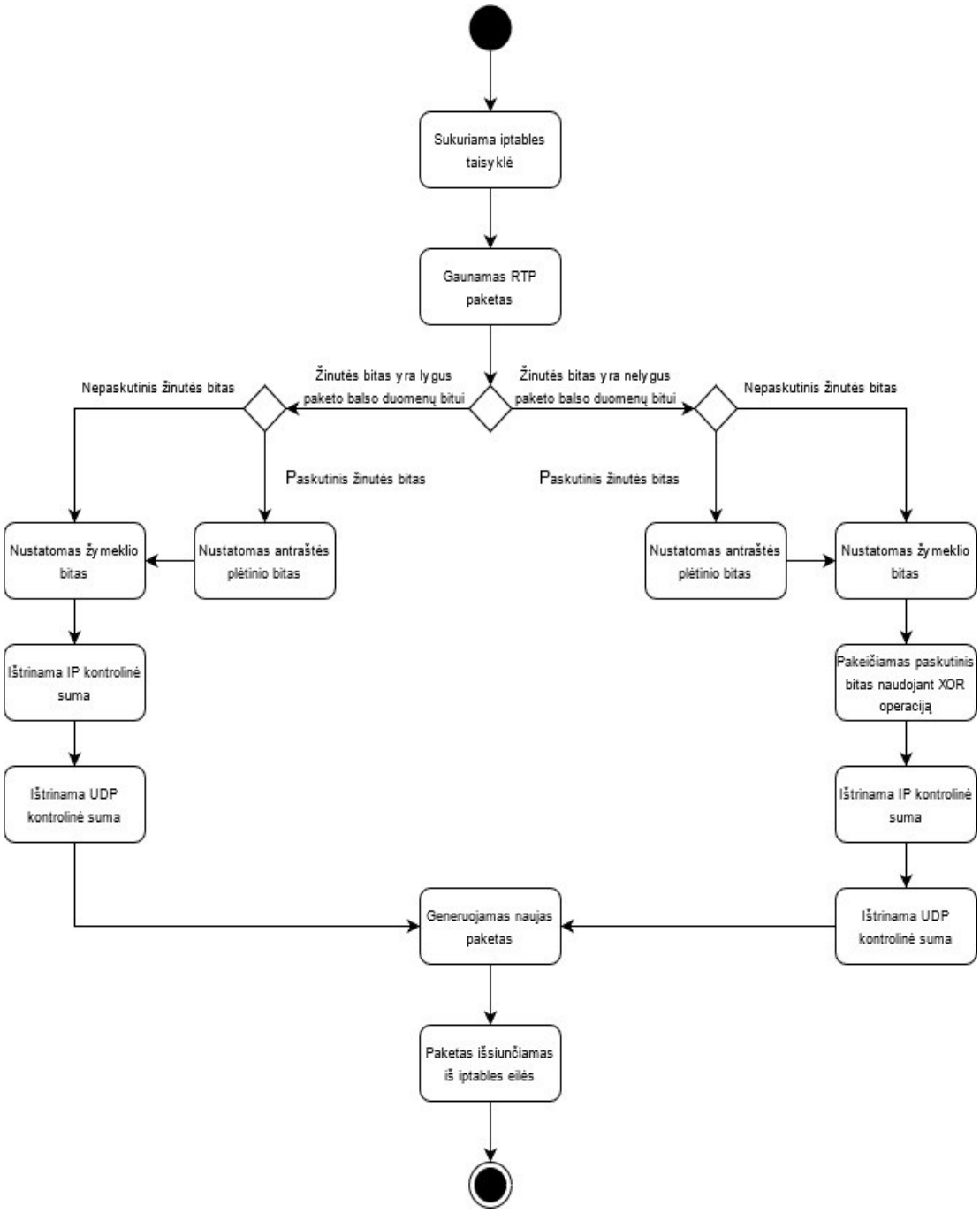
Prototipe yra įvykdoma komanda, kuri sukuria „iptables“ taisyklę: „iptables -I OUTPUT -s 10.0.1.0/24 -p udp –sport 7078 -j NFQUEUE –queue-num 1“. Komandoje yra nurodyta, kad taisyklė yra taikoma siunčiamam srautui (žodis „OUTPUT“), toliau yra nurodomas siunčiantysis IP adresas arba adresų grupė, bei prievado tipas ir numeris. Prototipe nurodytas prievado numeris 7078 yra naudojamas „Linphone“ standartiškai, bet esant poreikiui, galima koreguoti. Toliau pasirinkimas „-j NFQUEUE“ nurodo paskirties eilę, kurią naudoja „Netfilterqueue“ biblioteka. Į šią eilę bus nukreipti visi taisyklę atitinkantys paketai. Bei galiausiai yra nurodomas paskirties eilės numeris (tokių eilių gali būti ir daugiau). Gaunamas srautas taip pat yra nukreipiamas į „iptables“, todėl esant poreikiui, galima būtų atstatyti modifikuotų paketų antraštes į pradines reikšmes. Toliau jau prototipe yra padaromas sujungimas su „iptables“ nustatyta eile ir galima pradėti manipuliuoti duomenų srautu. Realizuojant prototipą buvo pastebėta, jog „Netfilterqueue“ biblioteka yra arba silpnai arba visai nebepalaikoma – ji neveikia su naujausia „Python“ versija (3.9), su „Python“ 3.8.5 versija tam tikros komandos neveikia (pvz. „set_payload“), reikia diegti senesnę „Netfilterqueue“ versiją, o informacijos apie galimas problemas ir jų sprendimus yra arba labai mažai arba visai nepateikta.



19 pav. RTP srauto perėmimas

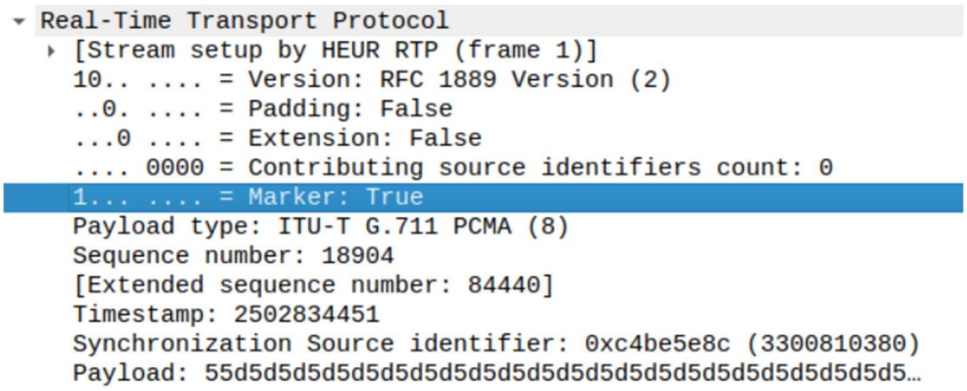
3.3. ŽINUTĖS ĮTERPIMAS Į RTP SRAUTĄ

Trečia prototipo sudedamoji dalis yra pačios žinutės įterpimas į RTP srautą. Kadangi tai yra patobulintas LSB metodas, kai siunčiama informaciją yra įterpiama į RTP paketo balso srauto duomenų lauką.



20 pav. Žinutės įterpimas į RTP paketą

Tam, kad prototipas galėtų nustatyti žinutės pabaigą yra naudojamas antraštės plėtinio bitas (angl. *padding*) – jis yra nustatomas. Šis bitas yra nustatomas pakete, kuriame yra slepiamos žinutės paskutinysis bitas.



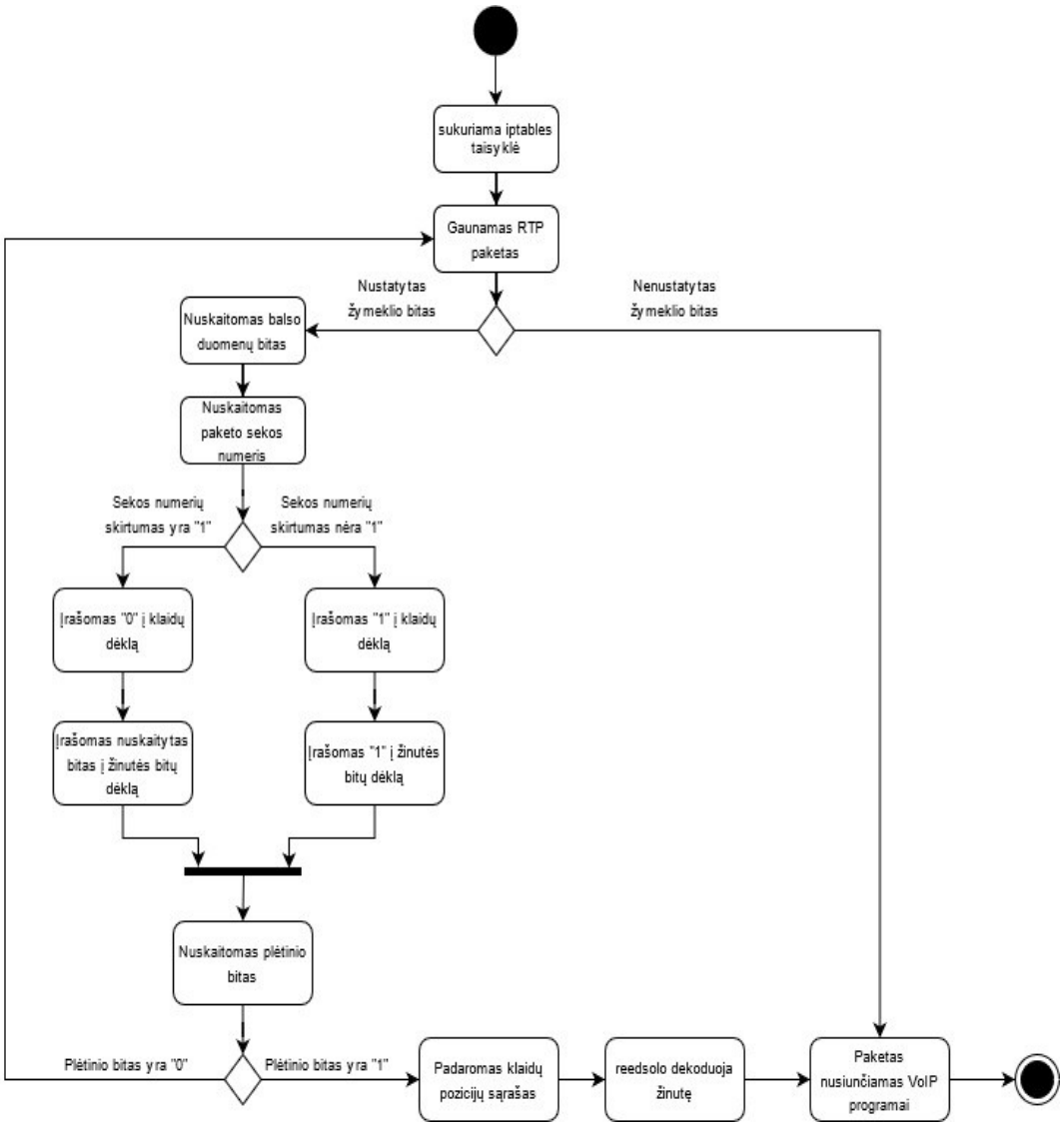
21 pav. RTP antraštės duomenys

Prototipe sukurtas *netfilterqueue* metodas gauna informaciją iš „iptables“ sukurtos eilės, kurioje pagal nustatytą taisyklę yra kaupiami paketai. Toliau yra aktyvuojama prototipo funkcija „siusti“, kuri ir atlieka reikiamas paketo modifikacijas. Tam yra naudojama „Python“ biblioteka „scapy“. „scapy“ biblioteka yra gana plačiai naudojamą ir turi neblogą aprašymą. Todėl pradėti naudotis ja yra lengva. Sunku buvo rasti specifinės informacijos apie RTP paketų modifikavimą, nes tai nėra populiarus protokolas plačioje visuomenėje, o aprašymai susiję su RTP protokolu menki. Biblioteka naudoja savus duomenų tipus, pavyzdžiui „*scapy.layers.l2.Ether*“, todėl norint atlikti paketų modifikacijas tenka manipuliuoti duomenų kodavimais.

Prototipas lygina pasirinktą RTP paketo balso duomenų bitą su įvestos žinutės pirmuoju bitu. Jei bitai yra lygūs, prototipas tik pakeičia reikiamus paketo antraštės laukus. Jei žinutės bitas nėra lygus RTP balso duomenų bitui, tuomet reikia iš naujo sukurti visą RTP paketo balso duomenų lauką su pakeistu bitu. Taip daroma todėl, jog NFQ ir „scapy“ naudoja savo sukurtus duomenų tipus ir paprasčiausiai pakeisti bito, naudojantis bitų loginėmis operacijomis, nėra galimybės. Nuskaitytų paketo duomenų tipai yra keičiami, tuomet yra atliekama loginė *ekskliūzyvaus arba* (angl. *XOR*) operacija su reikšme „1“. Tokiu būdu yra pakeičiama bito reikšmė. Tada žinutės bitų dėklas yra sumažinamas vienu bitu (nuo dėklo pradžios), kuris yra įkeltas į RTP paketo balso duomenų lauką. Galiausiai prototipas ištrina paketo tiek IP, tiek UDP kontrolines sumas ir yra sugeneruojamas naujas paketas, kuris galiausiai išsiunčiamas iš iptables eilės. Žinutės įterpimo algoritmas yra pavaizduotas 21 paveiksle.

3.4.ŽINUTĖS BITŲ IŠRINKIMAS IŠ RTP SRAUTO

Prototipas sukurs „iptables“ taisyklę, kad gaunamus RTP paketus siųstu į NFQ eilę. Tuomet bus tikrinamas ar paketas eilėje turi nustatytą žymeklio bitą bei ar nėra nustatytas plėtinio bitas. Jei pakete yra nustatytas žymeklio bitas, prototipas nuskaito nustatytą duomenų srauto bitą ir kelia jį į dėklą. Kai gaunamas paketas su plėtinio bitu, prototipas nuskaito paskutinį slepiamos žinutės bitą ir atlieka rekonstrukciją – iš dėklo imama po 8 bitus ir yra konstruojami baitai. Gauti baitai, tuomet, yra iškoduojami „reedsolo“ biblioteka.



22 pav. Žinutės išgavimo iš RTP algoritmas

Tam, kad būtų galima ištaisyti daugiau klaidų (tokį kiekį, koks yra nustatytas ECC baitų kiekis), „reedsolo“ bibliotekai galima pateikti klaidų žinutėje (įskaitant ir ECC baitus) pozicijas. Todėl prototipas tikrina kiekvieno gauto paketo su žymeklio bitu sekos numerius. Tikrinimas prasideda nuo gauto antro paketo. Tuomet iš naujai gauto paketo eilės numerio atimamas prieš tai gauto paketo eilės numeris. Jei atsakymas lygus vienetui, tada į klaidų dėklą yra įrašomas „0“. Priešingu atveju – įrašomas „1“ ir papildomai į žinutės bitų dėklą pridedamas „1“. Paskutinis veiksmas reikalingas tam, jog žinutės bitų skaičius atitiktų pradinį bitų skaičių. Tuomet klaidų dėkle imamos visos esančios pozicijos ir skaidomos po 8 bitus. Jei viename 8 bitų bloke yra bent vienas „1“, tuomet tai yra laikoma, kaip klaidingas baitas ir yra nurodoma jo pozicija dėkle. Kai yra vykdomas žinutės iškodavimas, klaidingų pozicijų sąrašas yra pateikiamas „reedsolo“ bibliotekai. Žinutės išgavimo iš RTP paketo algoritmas pavaizduotas 22 paveiksliuke.

3.5. PROTOTIPO TYRIMAS

Remiantis analizės išvadomis buvo nuspręsta patobulinti steganografijos metodą RTP pakete siekiant padidinti siunčiamų slaptų duomenų integralumą bei įvertinti galimybę naudoti ne tik paskutinįjį balso duomenų bitą.

Atliekant tyrimą patobulintas metodas buvo bandomas tiek realiomis sąlygomis, tiek kontroliuojamoje aplinkoje. Abiem atvejais buvo naudojamos dvi virtualios mašinos su „Linux Ubuntu“ 20.04 versija, sukurtomis naudojantis „VirtualBox“, bei „Linphone“ programinė įranga. Tiriant prototipą realiomis sąlygomis, buvo naudojamas Linphone nemokamas tarpinis SIP serveris. Kontroliuojamoje aplinkoje „Linphone“ vykdė VoIP skambutį tiesiogiai tarp virtualiose mašinose esančių programinės įrangos klientų, tačiau Linux esančio tinklo emuliatoriaus „netem“ pagalba buvo imituojami paketų praradimai. Balso skambučiai buvo analizuojami naudojantis „Wireshark“ ir „Onmipeek“ tinklo srauto paketų analizavimo įrankiais.

Tyrimo metu buvo siekiama išsiaiškinti:

1. Koks tinklo srauto paketų praradimas yra toleruojamas taikant „Reed-Solomon“ klaidų aptikimo ir taisymo metodą;
2. Kokią įtaką balso duomenims daro skirtingų balso duomenų bitų naudojimas;
3. Kokią įtaką balso duomenims daro dviejų balso duomenų bitų naudojimas;

„Reed-Solomon“ kodavimo efektyvumo tyrimas

Pradžioje buvo atliktas bandymas reguliuojamoje aplinkoje – kai nėra jokių trikdžių ir paketų praradimų. Tai buvo atskaitos taškas. Pirmiausiai buvo nustatomi ECC baitai – 10, o tyrimo metu siunčiama žinutė – „*1234567890*“ (10 baitų). Siunčiama iš virtualios mašinos A (IP adresas 10.0.1.41) į virtualią mašiną B (IP adresas 10.0.1.38).

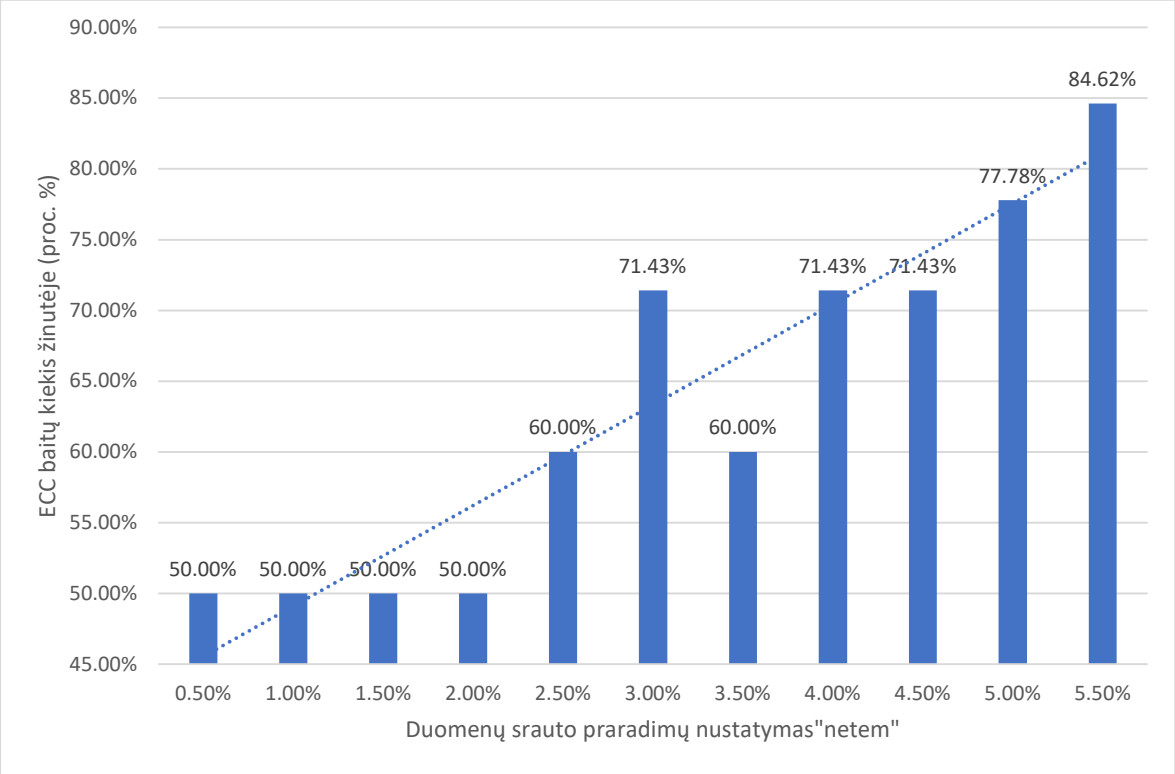
Toliau naudojantis „netem“ įrankiu buvo palaipsniui didinamas prarandamų duomenų srauto paketų kiekis. Didinimo žingsnis yra 0,5%. Žemiau, 3.1 lentelėje yra atlikti „netem“ bandymai su Linux įrankiu „iperf“, kuris yra skirtas tinklo testavimui. Buvo matuojamas tinklo našumas su įvairiais „netem“ nustatymais. Reikia pabrėžti, kad „netem“ paketus „numeta“ visiškai atsitiktine tvarka. Iš 3.1 lentelės matome, jog „netem“ nustatytas paketų praradimas atitinka 60% nuo realios situacijos.

3.1 lentelė. Tinklo emuliatoriaus „netem“ veikimas

„netem“ nustatymas (%)	Išsiųsta paketų (vnt.)	Nenusiųsti paketai (vnt.)	Rezultatas (%)
0,5	2675	11	0,4
1	2676	24	0,9
1,5	2676	29	1,1
2	2675	63	2,4
2,5	2675	67	2,5
3	2676	88	3,3

3,5	2675	92	3,4
4	2676	106	4
4,5	2675	128	4,8
5	2676	152	5,7

Toliau, buvo tiriama kiek ECC baitų reikia, siekiant iškoduoti gautą žinutę penkis kartus iš eilės su tam tikru „netem“ nustatymu. Nepavykus to padaryti, buvo didinamas ECC baitų skaičius. Didinimo žingsnis – penki ECC baitai.

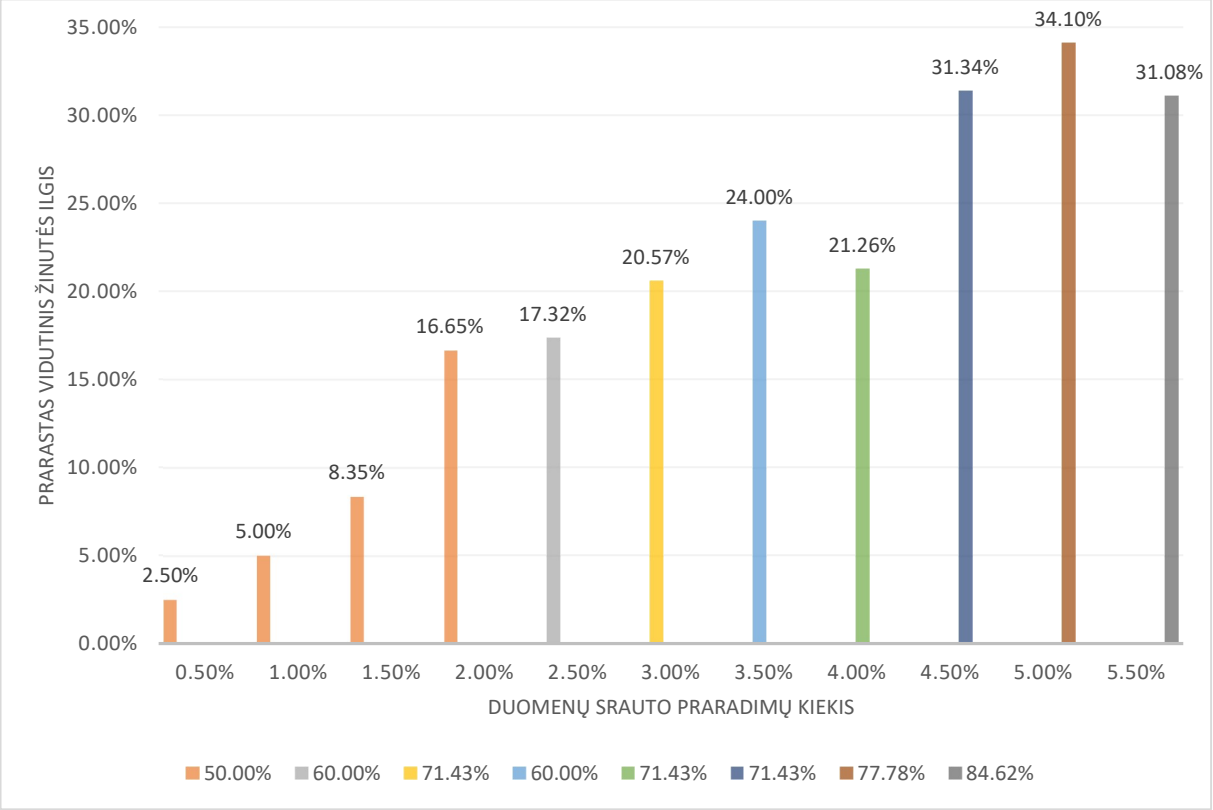


23 pav. ECC baitų kiekis žinutėje

24 paveiksliuke pateikta, kiek procentų ECC baitai turėjo sudaryti visos žinutės, kad esant tam tikriems tinklo srauto praradimams, gavėjas galėtų gauti ir iškoduoti žinutę penkis kartus iš eilės.

Žemiau, esančiame paveiksliuke (žr. 25 pav.) yra pateikta, koks visos žinutės (įskaitant ir ECC baitus) ilgis vidutiniškai buvo prarastas, kai gavėjas žinutę gavo ir galėjo dekoduoti. Svarbu pabrėžti, kad atliekant tyrimą, buvo susidurta su anomalija, kai „netem“ buvo uždėtas nustatymas – 5%. Nesvarbu, koks kiekis ECC baitų buvo naudojamas, tačiau žinutės pristatyti gavėjui penkis kartus iš eilės niekaip nepavyko. 23 ir 24 paveiksluose yra pateikiamas gautas geriausias rezultatas – 33,33% pristatytų žinučių.

Tiriant prototipą realiomis sąlygomis – naudojant „Linphone“ tarpinį SIP serverį (IP adresas *54.37.202.229*), deja, nei karto nebuvo prarastas nei vienas paketas, todėl „Reed-Solomon“ kodo nei karto neprireikė.



24 pav. Vidutiniai žinutės praradimai

Iš atliktos tyrimo dalies, galima teigti, jog metodo patobulinimas naudojant „Reed-Solomon“ klaidų aptikimo ir taisymo kodą siekiant persiųsti slaptas žinutes veikia. Net naudojant mobiliuosius duomenis, šiais laikais interneto duomenų srauto praradimai yra vis mažesni. Tačiau esant didesniems duomenų paketų praradimams, galima siųsti tik trumpas žinutes. Kitu atveju, net ir labai didelis ECC baitų kiekis nebepadeda atstatyti sugadintos žinutės.

Skirtingų balso duomenų bitų naudojimas žinutei slėpti

Kitas tyrimo etapas buvo skirtas išsiaiškinti kokią įtaką pokalbiui daro RTP balso duomenų skirtingų bitų modifikavimas. Kai yra naudojami G.711 balso kodeko *μ-law* arba *A-law* formatai, kiekvieno okteto paskutiniai keturi bitai yra skirti užkoduoti gamos akordus.

Tyrimo metu, atliekant VoIP skambutį, buvo leidžiamas tas pats muzikinis kūrinys ir persiunčiama ta pati žinutė – „*magistrinio baigiamojo darbo tyrimas*“. Žinutė kiekvienu atveju buvo slepiama naudojant kitus balso duomenų bitus. Pirmu bandymu buvo slepiama paskutiniame balso duomenų bite, antru bandymu – antrame nuo galo, trečiu bandymu – trečiame nuo galo, paskutiniu bandymu – ketvirtame nuo galo. Visais bandymais gavėjo virtualioje mašinoje buvo naudojamas „Wireshark“ tinklo srauto paketų analizavimo įrankis, siekiant įrašyti srautus ir vėliau analizuoti. Su „Wireshark“ buvo analizuojamos fliuktuacijos (angl. jitter) – gautų paketų vėlavimo variacija. Vėliau, naudojant „Omnipeek“ tinklo srauto paketų analizavimo įrankį buvo analizuojami balso duomenų kokybiniai parametrai:

- Pokalbio kokybinis vertinimas – MOS-CQ;
- Teorinis aukščiausias MOS įvertis naudojamam kodekui – MOS-Nom;

- R faktorius pagal ITU-T G.107 standartą;

Visus parametrus „Omnipeek“ skaičiuoja automatiškai. Tarptautinės telekomunikacijų sąjungos rekomendacijose ITU-T G.107 yra pateiktas vartotojų pasitenkinimo ir kokybinių MOS ir R faktoriaus reikšmių santykis (3.2 lentelė).

3.2 lentelė. ITU-T G.107 rekomenduojama kokybės pasitenkinimo skalė

MOS reikšmė (žemutinė riba)	R reikšmė (žemutinė riba)	Vartotojo pasitenkinimas kokybe
4.34	90	Labai patenkintas
4.03	80	Patenkintas
3.60	70	Kai kurie vartotojai nepatenkinti
3.10	60	Dauguma vartotojų nepatenkinti
2.58	50	Beveik visi vartotojai nepatenkinti

Tyrimo metu VoIP skambutis buvo vykdomas ir slapta žinutė buvo siunčiama iš virtualios mašinos A (IP adresas *10.0.1.41*), kurioje buvo įjungta garsi įvestis (garso išvestis buvo išjungta) į virtualią mašiną B (IP adresas *10.0.1.38*), kurioje buvo įjungta garso išvestis (garso įvestis buvo išjungta). Skambutis buvo vykdomas tiesiogiai tarp „Linphone“ aplikacijų nenaudojant jokio tarpinio SIP serverio. Virtualioje mašinoje B, siekiant įrašyti ir išsaugoti duomenų srautus, buvo naudojama „Wireshark“ programa.

Pirmu bandymu atliekamas skambutis, nesiunčiant jokios slaptos žinutės, siekiant turėti atskaitos tašką.

3.3 lentelė. Bazinio skambučio kokybiniai rodikliai

R faktorius	MOS-CQ	Fliuktuacijos (ms)	MOS-Nom
100	4,27	12,164	4,19

Matome, jog R faktorius yra aukščiausios reikšmės, o MOS-CQ viršija „Omnipeek“ kodekui numatytą nominalią vertę.

Antruoju bandymu buvo siunčiama slapta žinutė, kuri buvo slepiama balso duomenų paskutinių oktetų, paskutiniuose bituose.

3.4 lentelė. Kokybiniai rodikliai, kai žinutė slepiama paskutiniame bite

R faktorius	MOS-CQ	Fliuktuacijos (ms)	MOS-Nom
84	3,39	12,808	4,19

Pagal rodiklius matosi, jog balso kokybė yra suprastėjusi. Ir iš tiesų garsas siunčiant žinutę trūkinėjo. Trečiuoju bandymu buvo keičiamas balso duomenų okteto antras nuo galo bitas.

3.5 lentelė. Kokybiniai rodikliai, kai žinutė slepiama antrame nuo galo bite

R faktorius	MOS-CQ	Fliuktuacijos (ms)	MOS-Nom
30	2,02	13,521	4,19

Nors kokybiniai rodikliai yra ženkliai žemesni, tačiau klausantis įrašo, didesnio pablogėjimo lyginant su antruoju bandymu nebuvo pastebėta. Ketvirtuoju bandymu buvo keičiamas balso duomenų okteto trečias nuo galo bitas.

3.6 lentelė. Kokybiniai rodikliai, kai žinutė slepiama trečiame nuo galo bite

R faktorius	MOS-CQ	Fliuktuacijos (ms)	MOS-Nom
23	3,01	16,620	4,19

Kokybiniai parametrai yra panašiai blogi, kaip ir atlikus trečiąjį bandymą. Klausantis įrašo, skirtumų tarp antro, trečio ir ketvirto bandymų nesigirdi. Paskutiniuoju bandymu buvo keičiamas balso duomenų okteto ketvirtas nuo galo bitas.

3.7 lentelė. Kokybiniai rodikliai, kai žinutė slepiama ketvirtame nuo galo bite

R faktorius	MOS-CQ	Fliuktuacijos (ms)	MOS-Nom
16	3,06	12,161	4,19

Paskutiniojo bandymo kokybiniai rodikliai yra tokie pat blogi, kaip ir kitų bandymų su siunčiama žinute metu. Klausantis įrašo, didesnių skirtumų nuo buvusių bandymų nesigirdėjo.

Iš atlikto tyrimo galima pamatyti, jog keičiant paskutinįjį bitą, kokybiniai įrašo rodikliai buvo geresni nei keičiant kitus bitus. Tačiau klausantis įrašų, skirtumų nesigirdėjo. Todėl galima daryti išvadą, jog slaptą informaciją galime slėpti ir kituose balso duomenų bituose.

Dviejų balso duomenų bitų naudojimas žinutei slėpti

Paskutinis tyrimo etapas buvo atliekamas siekiant nustatyti, kokią įtaką balso kokybei darytų balso duomenų paskutinio okteto paskutinių dviejų bitų keitimas. Šiam tyrimui buvo šiek tiek patobulintas prototipas. Šiam tyrimui buvo tikrinami tie patys kokybiniai rodikliai, kaip ir praeitame tyrime.

3.8 lentelė. Kokybiniai rodikliai, kai žinutė slepiama dviejuose bituose

R faktorius	MOS-CQ	Fliuktuacijos (ms)	MOS-Nom
36	3,30	13,801	4,19

Iš gautų kokybinių rodiklių galima pastebėti, jog teoriškai kokybė suprastėja stipriai, bet ne blogiau nei keičiant tik vieną balso duomenų bitą. Tačiau praklausius įrašą padarytą su „Wireshark“ programa, galima drąsiai teigti, jog balso kokybė buvo ženkliai geresnė, lyginant su įrašais darytais, keičiant tik viena bitą. Buvo gerokai mažiau garso trūkinėjimų. Taip pat šiuo būdu buvo dvigubai padidintas PRBR ir RBR.

PROTOTIPO REALIZACIJOS IR TYRIMO IŠVADOS

Šiame skyriuje buvo apžvelgta kuriamo prototipo realizacija. Tai yra patobulintas RTP paketo balso duomenų srauto pasirinkto bito keitimo metodas. Šis metodas naudoja žinutės kodavimą RS, kad įvykus paketų praradimams, būtų galima atstatyti siunčiamą žinutę. Naudojant šį metodą, yra keletas apribojimų:

- ištaisomų klaidų kiekis – standartiškai yra iki pusės nustatytų ECC baitų skaičiaus. Jei yra žinomos klaidų pozicijos, tuomet teoriškai galima ištaisyti tiek klaidų, koks yra nustatytas ECC skaičius. Siekiant padidinti ištaisomų klaidų kiekį bent vienu, prie žinutės automatiškai yra pridedami du ECC baitai;
- naudojamų ženklų apribojimas – galima naudoti tik UTF-8 koduotės simbolius, nes su ilgesniais neveikia „reedsolo“ biblioteka;
- naudojant prototipą realiomis sąlygomis, gali nebūti galimybės modifikuoti pasirinktų antraštės laukų.

Naudojant šį metodą buvo modifikuojamas ne tik RTP paketo naudingosios įkrovos paskutinysis bitas, bet ir pačio paketo antraštės „žymeklio“ ir „plėtinio“ laukai. Šis modifikavimas yra būtinas siekiant nustatyti paketus kuriuose yra siunčiama žinutė.

Atlikus tyrimus galima daryti išvada, jog pavyko sukurti patobulintą steganografijos RTP protokole metodą. „Reed-Solomon“ klaidų aptikimo ir taisymo kodas, padeda atstatyti žinutę net esant ženkliems tinklo paketų praradimams – 5,5% prarastų paketų. Verta paminėti, kad esant tokiems dideliems paketų praradimams, tenka naudoti dideli ECC baitų skaičių (ECC baitų kiekis turėtų sudaryti apie 83% visos siunčiamos žinutės ilgio), bei trumpas žinutes. Tačiau generuojamo RTP srauto pilnai užtenka viskam persiųsti.

Atlikus tyrimą, taip pat, buvo išsiaiškinta, jog galima naudoti bet kurį (iš paskutinių keturių) balso duomenų, paskutinio okteto, bitą. Kokybiniai parametrai buvo geriausi, kai buvo naudojamas paskutinysis bitas, tačiau klausantis padarytų įrašų, apčiuopiamų skirtumų nesigirdėjo – visi garso įrašai siunčiant žinutę trūkinėjo, bet esmę suprasti buvo galima.

Atlikus bandymus, paaiškėjo, jog prototipą galima patobulinti naudojant du paskutinius balso duomenų bitus. Tyrimas parodė, jog naudojant šį metodą, balso kokybiniai rodikliai nesiskyrė nuo tų, kai buvo naudojamas tik vienas bitas informacijai slėpti. O klausantis įrašo, galima teigti, kad garso reali kokybė buvo net geresnė.

IŠVADOS

Atlikus analizę, galima teigti, jog steganografijos metodą galima taikyti visuose TCP/IP modelio lygiuose. Dauguma analizuotų metodų buvo sukurti seniai, tačiau naudojami protokolai yra nuolatos atnaujinami ar tobulinami, todėl kai kurie metodai šiandiena gali ir nebeveikti. Didžiausia problema taikant steganografiją tinkle yra persiunčiamos slaptos informacijos kiekis – dažniausiai jis yra labai mažas. Kita problema yra persiunčiamos slaptos informacijos integralumas. Jei transporto lygmenyje yra naudojamas TCP protokolas, integralumo klausimas yra išspręstas, tačiau jei yra naudojamas UDP protokolas, tuomet reikia metodo, galinčio užtikrinti duomenų integralumą.

Dėl šios priežasties buvo nuspręsta sukurti patobulintą steganografijos RTP protokole metodą. Pagrindinis patobulinimas yra „Reed-Solomon“ klaidų aptikimo ir taisymo kodo naudojimas. Speciali biblioteka pagal naudotojo įvestą žinutę sugeneruoja papildomus klaidų aptikimo ir taisymo baitus. Tokiu būdu praradus dalį paketų arba gavus juos ne iš eilės atsiranda gavėjas turi galimybė atstatyti pradinę žinutę.

Patobulintu variantu taip pat buvo siekiama išsiaiškinti ar galima modifikuoti ne tik paskutinįjį balso duomenų bitą. Pagal atliktą analizę, išsiaiškinta, jog balso duomenų okteto (naudojant G.711 kodeką) paskutiniai keturi bitai yra lygiaverčiai pagal juose esančią informaciją. Todėl buvo nuspręsta ištirti VoIP garso kokybinius parametrus, naudojant skirtingus RTP paketo balso duomenų bitus.

Šis metodas vienu paketu, gali persiųsti tik 1-ą bitą slaptos informacijos. Tai yra mažiausia galima reikšmė, lyginant su kitais steganografijos metodais RTP protokole. Bet lyginant kiek slaptos informacijos galima persiųsti vieno skambučio metu šis metodas lenkia kitus. Kadangi VoIP skambučiai generuoja milžiniškus kiekius duomenų paketų, šis trūkumas laikytinas nežymiu.

Atlikus prototipo tyrimą, buvo nustatyta, jog „Reed-Solomon“ klaidų aptikimo ir taisymo kodo naudojimas pasiteisino. Žinutę buvo galima atstatyti ir perskaityti esant net 5,5% paketų praradimams. Išanalizavus kokybinius duomenis, kai buvo keičiami skirtingi balso duomenų bitai, nustatyta, kad teoriškai, naudojant ne paskutinįjį balso duomenų bitą, yra daroma įtaką garso įrašo kokybei. Tačiau klausantis padarytų įrašų, skirtumų pastebėta nebuvo. Paskutinis tyrimas buvo atliktas, siekiant išsiaiškinti, kokią įtaką garso duomenų kokybei darytų, jei žinutė būtų slepiama dvejuose paskutiniuose garso duomenų bituose. Išanalizavus duomenis, paaiškėjo, jog skirtumo nėra ar žinutė yra slepiama viename bite ar dvejuose. Kokybiniai rodikliai buvo labai panašūs, o reali garso kokybė buvo net geresnė, nei naudojant tik vieną balso duomenų bitą – buvo mažiau trūkinėjimų.

Įvertinus visus tyrimo rezultatus, galima teigti, jog išsikeltas tikslai – nustatyti ir išanalizuoti metodus, naudojamus slaptiems duomenų perdavimams RTP protokolus ir sukurti ir išbandyti patobulintą steganografijos RTP protokole metodą, kuris naudotų klaidų aptikimo ir taisymo algoritmą, yra pasiekti.

Literatūros sąrašas

[1] M. Nosrati, R. Karimi ir M. Hariri, „An introduction to steganography methods," *World Applied Programming,* p. 5, 2011.

[2] P. Dobriyal, J. Yadav ir J. Jain, „A Review on Text Based Steganography," *Research Journal of Science & IT Management,* p. 7, 2015.

[3] M. T. Ahvanooey, Q. Li, J. Hou, A. R. Rajput ir Y. Chen, „Modern Text Hiding, Text Steganalysis, and Applications: A Comparative Analysis," *Entropy,* p. 31, 2019.

[4] V. Nagaraj, D. V. Vijayalakshmi ir D. G. Zayaraz, „Overview of Digital Steganography Methods and Its Applications," *International Journal of Advanced Science and Technology,* p. 14, 2013.

[5] W. Mazurczyk, M. Smolarczyk ir K. Szczypiorski, „Retransmission steganography and its detection," *Soft Computing,* 2011.

[6] J. Lubacz, W. Mazurczyk ir K. Szczypiorski, „Principles and overview of network steganography," *IEEE Communications Magazine,* t. 52, nr. 5, pp. 225 – 229, 2014.

[7] S. WENDZEL, S. ZANDER, B. FECHNER ir C. HERDIN, „A Pattern-based Survey and Categorization of Network CovertChannel Techniques," *ACM Computing Surveys,* pp. 1-26, 2015.

[8] R. Chandramouli, „A Mathematical Approach to Steganalysis," *Proc. SPIE Security and Watermarking of Multimedia Contents IV,* p. 12, 2002.

[9] W. Frączek, W. Mazurczyk ir K. Szczypiorsk, „How Hidden Can Be Even More Hidden?," 2011.

[10] W. Mazurczyk ir K. Szczypiorski, „Steganography of VoIP Streams," įtraukta *OTM Confederated International Conferences „On the Move to Meaningful Internet Systems"*, 2008.

[11] A. Mileva ir B. Panajotov, „Covert Channels in TCP/IP Protocol Stack," *Central European Journal of Computer Science,* pp. 45-66, 2014.

[12] D. Martins ir . H. Guyennet , „Attacks with Steganography in PHY and MAC Layers of 802.15.4 Protocol," įtraukta *2010 Fifth International Conference on Systems and Networks Communications*, Nice, 2010.

[13] Z. Trabelsi, H. El-Sayed, L. Frikha ir T. Rabie, „A novel covert channel based on the IP header record," *Int. J. Advanced Media and Communication,* t. 1, nr. 4, 2007.

[14] Z. Liu, Y. Jiang ir P. Qian, „A Data-Hiding Method Based on TCP/IP Checksum," įtraukta *Advances in Computer Science and its Applications*, 2014.

[15] W. Frączek, W. Mazurczyk ir K. Szczypiorski, „Hiding Information in a Stream Control Transmission Protocol," *Computer Communications,* 2011.

[16] IETF, „IETF Transport Area Working Group," 2020. [Tinkle]. Available: https://datatracker.ietf.org/wg/tsvwg/about/.

[17] A. Dyatlov ir S. Castro, „Exploitation of data streams authorized by a network access control system for arbitrary data transfers : tunneling and covert channels over the HTTP protocol," Grey World, 2003.

[18] B. Dimitrova ir A. Mileva, „Steganography of Hypertext Transfer Protocol," *Journal of Computer and Communications,* t. 5, pp. 98-111, 2017.

[19] W. Mazurczyk ir J. Lubacz, „LACK – a VoIP Steganographic Method," *Telecommun Systems,* t. 45, pp.153–163, 2010, DOI. 10.1007/s11235-009-9245-y.

[20] AOKI, N. „A technique of lossless steganography for G.711 telephony speech," In Proceedings - 2008 4th *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IIH-MSP 2008, DOI 10.1109/IIH-MSP.2008.122.