KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Lukas Morkeliūnas

Steganografijos metodas, naudojant UDP protokolą

Baigiamasis magistro darbas

Vadovas

Prof. Algimantas Venčkauskas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

STEGANOGRAFIJOS METODAS, NAUDOJANT UDP PROTOKOLĄ

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas

(parašas) Prof. Algimantas Venčkauskas

(data)

Recenzentas

(parašas) Prof. dr. Rimantas Plėštys

(data)

Projektą atliko

(parašas) Lukas Morkeliūnas

(data)

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos

(Fakultetas)

Lukas Morkeliūnas

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

(Studijų programos pavadinimas, kodas)

„Steganografijos metodas, naudojant UDP protokolą“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 17 m. gegužės 22 d.

Kaunas

Patvirtinu, kad mano, **Luko Morkeliūno,** baigiamasis projektas tema „UDP Steganografinis metodas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nė viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

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

Morkeliūnas, L. „STEGANOGRAFIJOS METODAS, NAUDOJANT UDP PROTOKOLĄ“. Magistro baigiamasis projektas / vadovas prof. Algimantas Venčkauskas; Kauno technologijos universitetas, informatikos fakultetas, kompiuterių katedra.

Kaunas, 2017. 71 p.

SANTRAUKA

Komunikacijai vykdyti tinkle tarp įrenginių yra naudojami tinklo protokolai. TCP/IP yra tinklo protokolų rinkinys, naudojamas internete. Slaptas informacinis kanalas – tai bet koks komunikacinis kanalas, kuris gali būti naudojamas slaptiems duomenims perduoti. Kaip ir tradiciniuose komunikaciniuose kanaluose, slaptame informacijos kanale gali susidaryti triukšmo (angl. *noise*). UDP protokolu perduodamiems duomenims nėra užtikrinama duomenų tėkmės kontrolė, taip pat UDP duomenų paketai gavėją gali pasiekti ne ta tvarka, kuria buvo siunčiami. Slaptas informacijos kanalas, paremtas UDP protokolu, neužtikrina informacijos pristatymo bei nekontroliuoja duomenų pristatymo eiliškumo. Šioms problemoms spręsti gali būti naudojami klaidų korekcijos algoritmai, skirti prarastiems duomenims atkurti, bei papildoma duomenų eiliškumą užtikrinanti informacija.

Darbo tikslas – sukurti steganografinį metodą, užtikrinantį prarastų duomenų atkūrimą bei persiunčiamų duomenų eiliškumą naudojant UDP protokolą. Sukurtas metodas turi atitikti keliamus reikalavimus: gebėti atkurti gautų duomenų pradinį eiliškumą bei, kai įmanoma, atkurti ir prarastus duomenis.

Galutinis baigiamojo darbo rezultatas – metodo realizacija panaudojant UDP protokolą. Atlikus tyrimą yra nustatomas metodo efektyvumas, lyginant su tradiciniais UDP steganografiniais metodais.

Raktažodžiai: *steganografija, UDP, klaidų korekcijos kodai, metodai.*

Morkeliūnas Lukas. *STEGANOGRAPHIC METHOD USING UDP PROTOCOL*: *Master's* thesis / supervisor assoc. prof. Algimantas Venčkauskas. Department of Computer Science, Faculty of Informatics, Kaunas University of Technology.

Kaunas, 2017. 71 p.

SUMMARY

A network protocol defines rules and conventions for communication between network devices. TCP/IP is a family of network protocols used on the Internet. Secret Channel - is any communication channel that can be used for sensitive data transmission. As with conventional communication channels - the secret information in the channel can exposed to noise. UDP has no handshaking dialogues, and thus exposes the user's program to any unreliability of the underlying network: there is no guarantee of delivery, ordering, or duplicate protection. Secret information channel based on UDP protocol does not guarantee delivery of information, data, and control over the presentation sequence. Error correcting algorithms and additional sequencing information can be used to solve these issues.

The aim of this thesis is to create a steganographic method that would ensure the recovery of lost data and packets sequencing information retention using UDP protocol. The method must meet the following requirements: to be able to restore the original order of the data received and to restore the lost data when possible.

The result of this thesis is implementation of proposed method using UDP protocol. Experiment results was used to compare proposed method effectives against other network steganography methods.

Key words: *steganography, UDP, error correcting codes, methods*

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

PDU (angl. *protocol data unit*) – protokolo duomenų vienetas.

QoS (angl. *quality of service*) – paslaugos kokybė.

UDP (angl. *user datagram protocol*) – nepatikimas duomenų perdavimo protokolas.

TCP (angl. *transmission control protocol*) – patikimas duomenų perdavimo protokolas.

IP (interneto protokolas) – taisyklių visuma, apibrėžianti duomenų mainų būdą tarp dviejų kompiuterinių sistemų.

MAC (angl. *media access control*) – unikalus tinklo įrenginio identifikatorius.

RFC (angl. *requests for comments*) – IETF organizacijos dokumentas, kuriame pateikiamos techninės ir organizacinės pastabos apie internetą.

OSI (angl. *open systems interconnection reference model*) – abstraktus ryšio protokolų, naudojamų ryšio ir kompiuteriniuose tinkluose, aprašymas.

HTTP (angl. *hypertext transfer protocol*) – pagrindinis metodas informacijai pasauliniame tinkle pasiekti.

DNS (angl. *domain name system*) – protokolas, leidžiantis kreiptis į tinklo resursus lengviau įsimenamu simboliniu vardu.

IMAP (angl. *internet message access protocol*) – elektroninio pašto serverio protokolas.

DDOS (angl. *distributed denial-of-service*) – paskirstyta paslaugos nutraukimo ataka.

VOIP (angl. *voice over ip*) – balso ryšys, perduodamas duomenų perdavimo tinklais naudojant interneto protokolą.

P2P (angl. *peer-to-peer*) – tinklo modelis, kuriame vartotojai tiesiogiai keičiasi resursais.

SDU (angl. *service data unit*) – duomenų vienetas, perkeltas iš aukštesnio OSI lygmens į žemesnį.

MTU (angl. *maximum transmission unit*) – maksimalus protokolo duomenų vienetas, galimas perduoti per vieną tinklo transakciją.

PCI (angl. *protocol-control information*) – duomenų paketo antraštės informacija.

ASCII (angl. *american standard code for information interchange*) – simbolių kodavimo standartas.

ARP (angl. *address resolution protocol*) – protokolas, naudojamas logiškai susieti MAC adresus su IP adresais.

ICMP (angl. *internet control message protocol*) – interneto kontrolės žinučių protokolas.

IEEE (angl. *institute of electrical and electronics engineers*) – profesinė asociacija.

bPP (angl. *bits per packet*) – bitų kiekis viename duomenų pakete.

TTL (angl. *time to live*) – duomenų paketo gyvavimo trukmė.

PMTUD (angl. *path mtu discovery*) – maksimalaus protokolo duomenų vieneto nustatymo metodas, naudojantis ICMP.

PLPMTUD (angl. *packetization layer path mtu discovery*) – maksimalaus protokolo duomenų vieneto nustatymo metodas, naudojantis TCP.

ICV (angl. *integrity check value*) – vientisumo patikrinimo vertė.

IGMP (angl. *internet group management protocol*) – komunikacinis IPv4 protokolas, skirtas grupinių transliacijų sudarymui.

DHCP (angl. *dynamic host configuration protocol*) – standartizuotas protokolas, skirtas dinaminiam IP adresų skirstymui.

ACK (angl. *acknowledgement*) – naudojamas TCP protokole atsakomasis ryšys, informuojantis apie sėkmingą sesijos sudarymą, gautą duomenų paketą ir kt.

ISN (angl. *initial sequence number*) – 32 bitų ilgio unikalus identifikacinis numeris, atskiriantis skirtingas TCP sesijas.

Apache – atvirojo kodo HTTP tinklo serveris.

Microsoft IIS – HTTP tinklo serveris.

Cname (angl. *canonical name record*) – naudojamas DNS protokole įrašo tipas, skirtas nurodyti subdomeno priklausomybę kitam domenui ar subdomenui.

RAID – (angl. *redundant array of independent disks*) – perteklinis nepriklausomų diskų masyvas.

IOA – Osamah Ibrahiem Abdullaziz steganografinis metodas.

KS – Krzysztof Szczypiorski steganografinis metodas.

ĮVADAS

„Steganografijos metodas, naudojant UDP protokolą“ – informacijos ir informacinių technologijų saugos magistrantūros studijų programos baigiamasis darbas.

Komunikacijai vykdyti tinkle tarp įrenginių yra naudojami tinklo protokolai. TCP/IP yra tinklo protokolų rinkinys, naudojamas internete. Slaptas informacinis kanalas – tai bet koks komunikacinis kanalas, kuris gali būti naudojamas slaptiems duomenims perduoti. Kaip ir tradiciniuose komunikaciniuose kanaluose, slaptame informacijos kanale gali susidaryti triukšmo (angl. *noise*). UDP protokolu perduodamiems duomenims nėra užtikrinama duomenų tėkmės kontrolė, taip pat UDP duomenų paketai gavėją gali pasiekti ne ta tvarka, kuria buvo siunčiami. Slaptas informacijos kanalas, paremtas UDP protokolu, neužtikrina informacijos pristatymo bei nekontroliuoja duomenų pristatymo eiliškumo. Šioms problemoms spręsti gali būti naudojami klaidų korekcijos algoritmai, skirti prarastiems duomenims atkurti, bei papildoma duomenų eiliškumą užtikrinanti informacija.

Todėl, remiantis šiais pastebėjimais, buvo suformuluotas darbo tikslas: **sukurti steganografinį metodą, užtikrinantį prarastų duomenų atkūrimą bei siunčiamų duomenų eiliškumą UDP protokole.**

Tikslui pasiekti buvo įgyvendinti šie uždaviniai:

- atlikta literatūros analizė tiriamajai sričiai apibūdinti;
- išanalizuoti esami tinklo steganografiniai metodai ir pritaikytos jų charakteristikos metodo kūrimo procese;
- išanalizuoti esami klaidų korekcijos metodai. Pasirinkti metodai pritaikyti metodo kūrimo procese;
- metodas pritaikytas duomenų perdavimui UDP protokolu;
- atliktas sukurto sprendimo tyrimas ir rezultatų analizė.

Darbą sudaro trys dalys: 1. steganografinių metodų TCP/IP tinkluose analizė, esamų steganografinių tinklo metodų tyrimas ir analizės dalyje apibrėžti tikslai problemai spręsti. 2. Aprašomas siūlomas steganografinis metodas, naudojamas UDP protokole. Jis yra sudarytas iš dviejų dalių: metodo koncepcijos aprašymo bei jo simuliacijos virtualioje aplinkoje. 3. Atliktas metodo tyrimas, jis lyginimas su kitų autorių metodais.

1. STEGANOGRAFIJA

Analizės tikslas – surasti ir išnagrinėti pasirinktus steganografinius metodus skirtinguose TCP/IP modelio sluoksniuose įvertinant slapto informacijos kanalo patikimumą, taip pat suprasti klaidų korekcijos kodų veikimo principus bei įvertinti jų veikimo specifiką ir efektyvumą.

Analizės uždaviniai:

- apžvelgti ir išanalizuoti žinomus tinklo steganografijos metodus;
- apžvelgti ir išanalizuoti žinomus klaidų korekcijos algoritmus;
- išsiaiškinti slaptos informacijos perdavimo kanalo patikimumą naudojant skirtingus steganografinius metodus;
- apibendrinti gautus analizės rezultatus tolimesnei darbo eigai.

Šiuolaikinė steganografijos technologija remiasi faktu, kad beveik visi elektroniniai dokumentai, paveikslėliai, garso ir vaizdo įrašai turi kažkiek neišnaudotų ar nereikšmingų mažiausių informacijos vienetų — bitų. Steganografija išnaudoja šių plotų galimybes: arba tam tikra informacija įrašoma į miniatiūrines, dar neišnaudotas vietas, arba tam tikra informacija pakeičia žmogui nepastebimas ar negirdimas originalaus dokumento informacijos daleles. Tada pakeistas dokumentas gali būti siunčiamas kaip niekuo neišsiskiriantis paveikslėlis ar garso įrašas.

Dažniausiai steganografija apima informacijos slėpimą skaitmeniniuose paveikslėliuose ir garso rinkmenose. Pranešimui vaizde paslėpti reikia dviejų rinkmenų – slepiančiosios vaizdo rinkmenos ir slepiamo pranešimo. Pranešimas gali būti paprastas tekstas, šifruotas tekstas arba kita vaizdo rinkmena. Sujungus slepiantį vaizdą ir slepiamą pranešimą gaunamas stegovaizdas. Pranešimui paslėpti ir atidengti dažniausiai naudojamas pasirinktas stegoraktas, kuris įvedamas specialia steganografijos programine įranga. Steganografinis pranešimas dažniausiai iš pradžių užšifruojamas, o paskui įkomponuojamas į slepiantį vaizdą. Tik tas, kuris žino, kaip buvo įkomponuotas pranešimas, gali jį rasti ir, jei reikia, iššifruoti.

Pagal slepiamų duomenų nešlius steganografijos tipai yra skirstomi į:

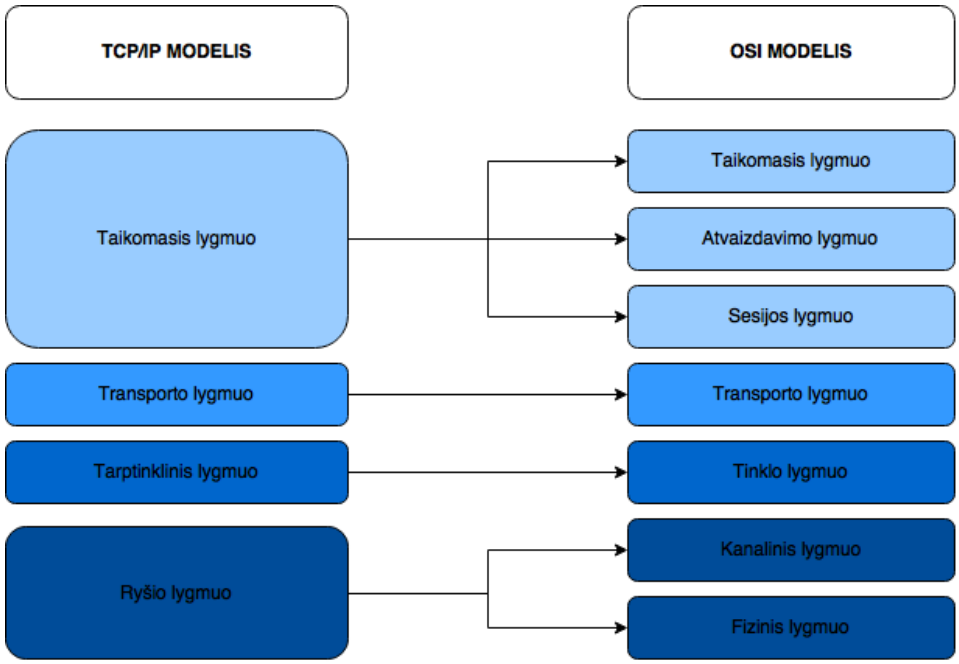
1. **Teksto steganografiją**: sunkiausiai įgyvendinama technika dėl teksto vientisumo palyginus su vaizdo įrašų ar protokolų. Tačiau jos įgyvendinimui reikia mažiau atminties. Vienas iš metodų, naudojamų teksto steganografijoje, yra duomenų suspaudimas: jis užkoduoja informaciją kitu formatu, kuris gali užimti mažiau atminties. Plačiausiai žinomas kodavimas yra Huffmano duomenų kompresijos algoritmas, dar kitaip vadinamas Huffmano kodavimo medžiu.
2. **Tinklo protokolų steganografiją**. Tai steganografijos metodai, skirti pakeisti esamus komunikacijos protokolus, nepažeidžiant jų veikimo, kad komunikacija vyktų įprastai ir visos tinklo dalys neatpažintų anomalijų siunčiamuose paketuose.
3. **Grafinių elementų steganografiją**. Ši technika plačiausiai naudojama paslėpti slaptos žinutės bitus tiesiogiai grafiniame elemente, nekeičiant jos formato. Pagrindinė idėja yra tokia: žinutės bitai yra slepiami triukšmingose grafinio elemento pozicijose, kur didelė spalvų variacija. Grafinių elementų steganografija yra skirstoma į:
    a) mažiausiai reikšmingo bito (angl. *Least significant bit*) integraciją;
    b) algoritmų ir transformacijų;
    c) šifravimo ir sklaidos (angl. *Encryption & scatter*);

d) perteklinio šablono kodavimo (angl. *Redundant pattern encoding*);
e) maskavimo ir filtravimo.

4. **Vaizdo įrašų steganografiją**: šis steganografijos tipas naudoja ir garso, ir grafinio elemento steganografijos metodų kombinacijas. Didžiausias šio metodo privalumas – didelis slepiamos informacijos kiekis dėl galimybės informaciją įterpti garso signale, ir vaizdo sraute.
5. **Garso įrašų steganografija**: garso steganografijos įgyvendinimas yra sudėtingesnis lyginant su grafinių elementų steganografija. Šio metodo esmė yra paslėpti informaciją skaitmeniniame garso signale. Slėpimo technika gali būti naudojama trims garso formatams (WAV, AU ir MP3) koduoti.

1.1. TCP/IP Modelis

Komunikacijai vykdyti tinkle tarp įrenginių yra naudojami tinklo protokolai. TCP/IP yra internete naudojama tinklo protokolų šeima. Tinklo protokolas – tai taisyklių visuma, apibrėžianti duomenų mainų būdą. Interneto protokolų specifikacija aprašyta *Requests for Comment* (RFC) puslapyje. Tarptautinė standartų organizacija (angl. *ISO*) yra pristačiusi standartizuotą būdą skirtingiems protokolų lygmenims atvaizduoti vadinamuoju ISO **OSI.** TCP/IP šeimos modelis yra sudarytas iš 4 lygmenų, o OSI modelis – iš 7. Nors TCP/IP ir OSI modeliai yra skirtingi, jų *tinklo* ir *perdavimo* lygmenys yra labai panašūs. TCP/IP modelyje nėra nagrinėjami *duomenų* ir *fiziniai lygmenys*. Taip pat OSI nėra suderinamas su TCP/IP modeliu (1 pav.).



1 pav. TCP/IP ir ISO/OSI modelių skirtumai

1.1.1. Kanalo lygis

Kanalo lygmuo jungia du OSI modelio lygius: 1 lygis – fizinis lygis ir 2 – ryšio lygis. Kanalo lygmens paskirtis yra duomenų perdavimas tarp sistemų. Prieš išsiunčiant duomenų bitai kartu su tokiais informaciniais bitais kaip kontrolinė suma yra sugrupuojami į fiksuoto dydžio grupes. Nors tai yra vienas paprasčiausių modelio lygių, tačiau jis yra vienas iš svarbiausių, kadangi nuo šio lygmens taisyklingo funkcionavimo priklauso visų aukštesnių lygių veikimas.

1.1.2. Tarptinklinis lygis

Tarptinklinis lygmuo (2 pav.) užtikrina duomenų perdavimą tarp skirtingų tinklų, taip pat ir internete. Duomenų perdavimas vykdomas sudalijant duomenis į mažesnių duomenų blokus, vadinamus paketais. Siuntėjas duomenų paketus vieną po kito išsiunčia, o gavėjas juos vieną po kito gauna. Gavėjas sulaukia, kol visi išsiųsti duomenų paketai gaunami ir tuomet juos perduoda į aukštesnį modelio lygmenį [1]. Interneto protokolas (IP) priskiriamas tarptinkliniam lygmeniui. Šiuo protokolu yra logiškai sujungiami tinklo įrenginiai. Adresatui pasiekti naudojamas gavėjo adresas, kuris nurodomas IP duomenų paketo antraštėje. Kiekvienas duomenų paketas perduodamas individualiai. Vienos duomenų perdavimo sesijos metu išsiųsti duomenų paketai gali gavėją pasiekti skirtingais keliais. Dėl šios priežasties gavėjas duomenis gaus kitokia eilės tvarka, nei jie buvo išsiųsti.

1.1.3. Perdavimo lygis

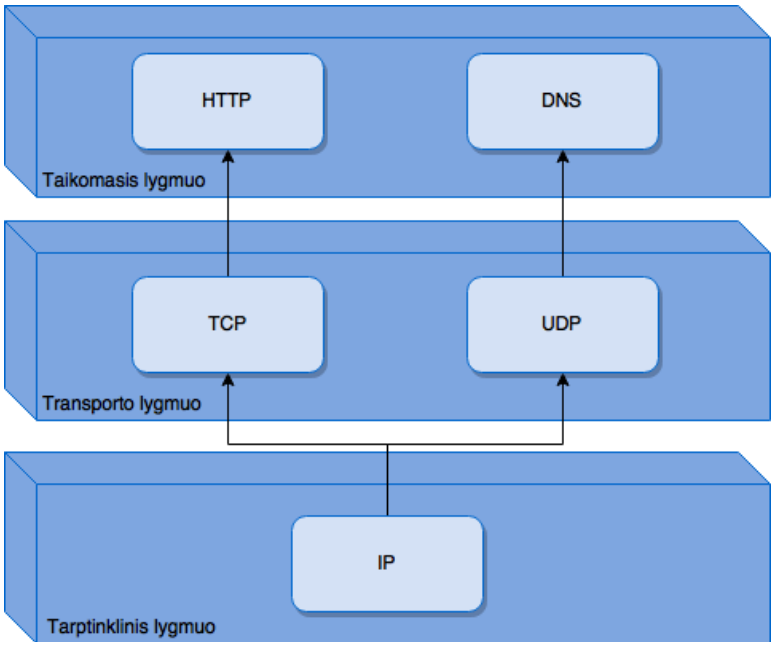
TCP ir UDP protokolai priskiriami TCP/IP modelio transporto lygmeniui (2 pav.). TCP protokolu perduodami sudalinti duomenys yra vadinami *segmentais*, tuo tarpu UDP duomenų blokai vadinami *datagramomis*. Pagrindinis skirtumas tarp TCP ir UDP yra tai, kad TCP užtikrina perduodamų duomenų pristatymo patvirtinimą, siuntimo metu prarastų duomenų pakartojimą gali inicijuoti gavėjas. UDP protokolu perduodami duomenys neturi jokio grįžtamojo ryšio, siuntėjas, išsiuntęs duomenis, neturi jokios galimybės sužinoti, ar jie sėkmingai pasiekė gavėją. Toks duomenų siuntimas be patvirtinimo leidžia duomenis siųsti greičiau.

1.1.4. Taikymo lygis

Taikomasis TCP/IP modelio lygmuo (2 pav.) jungia tris OSI modelio lygius: 7 – taikomąjį lygį, 6 – atvaizdavimo lygmenį ir 5 – sesijos lygį. Taikomojo lygmens protokolai gali būti išskirti į dvi grupes [2]:

- taikomųjų aplikacijų protokolai (HTTP, DNS, IMAP);
- paslaugų protokolai (maršrutizavimo protokolai).

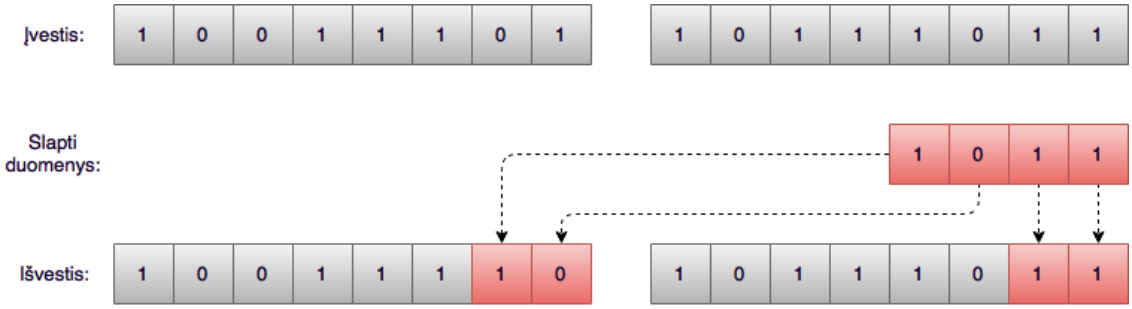
Standartiniams protokolams yra gana griežtai išskiriami prievadų numeriai – http:80, dns:53, ftp:21.



2 pav. Protokolų pasiskirstymas TCP/IP modelyje

1.2. Slapto kanalo steganografijos metodai

Atviras komunikacinis kanalas – tai komunikacinis kanalas, jungiantis įrenginius (pvz., kompiuterius, mobiliuosius telefonus) arba kompiuterinius tinklus ir yra skirtas autorizuotiems duomenims perduoti. Slapto informacinio kanalo sąvoką pirmasis pristatė B. W. Lampsonas [3]. Tai bet koks komunikacinis kanalas, kuris gali būti išnaudojamas slaptiems duomenims perduoti (3 pav.) siekiant apeiti sistemų saugos politiką. Kaip slapto informacijos perdavimo kanalas gali būti panaudota bet kokia vieša infrastruktūra. Slapti kanalai gali būti skirstomi į *laiko* ir *talpos* duomenų perdavimo kanalus. Talpos atveju siuntėjas įrašo slaptus duomenis, o gavėjas skaito. Laiko atveju, informacija perduodama laiko įvykiais (pvz., sutartais laiko intervalais tarp siunčiamų duomenų paketų). Laiko metodams taip pat priskiriami įvykių skaičiavimo metodai, kai duomenys perduodami sekant tam tikrų įvykių sekas. Laiko perdavimo kanalai skirstomi į aktyvius ir pasyvius. Aktyvūs – kai slaptiems duomenis perduodi generuojamas papildomas duomenų srautas, pasyvūs – kada duomenims perduoti naudojamas natūraliai tinkle esantis srautas.



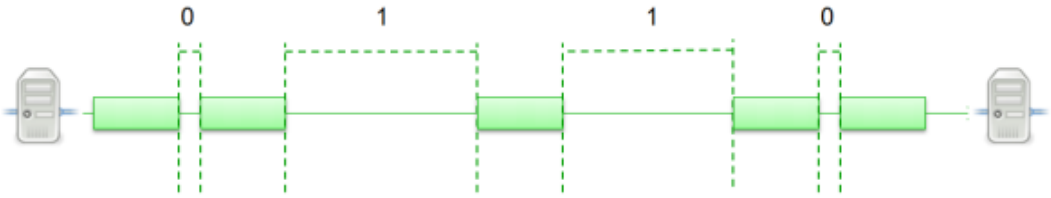
3 pav. Slaptos informacijos įterpimas

Kaip ir tradiciniuose komunikaciniuose kanaluose, taip ir slaptame informacijos kanale gali susidaryti triukšmo (angl. *noise*). Pagal tai, ar tarp komunikuojančių pusių komunikacinis kanalas užmezgamas per tarpinį komunikacijos tašką, kanalai skirstomi į tiesioginius ir netiesioginius. Informaciniai kanalai, tunelio principu besitęsiantys per kelis skirtingus skirtingų lygmenų protokolus, vadinami užpildo tuneliais (angl. *payload tunnel*). Slapti informacijos kanalai steganografijos srityje yra nagrinėjami plačiąja prasme. Šis darbas yra orientuotas į steganografinius metodus, naudojamus telekomunikaciniuose tinkluose, kurie yra geriau žinomi tinklo steganografijos pavadinimu.

Slaptų informacijos kanalų pralaidumui įvertinti naudojamas slaptų informacijos bitų persiuntimo skaičius per sekundę (angl. *bps*) [4, 5] arba slaptų informacijos bitų skaičius viename duomenų pakete (angl. *bpp*). Tuo atveju, kada duomenų paketų praradimas yra nulinis, talpos kanalų pralaidumas skaičiuojamas $bpp * n$; n – duomenų paketų skaičius per sekundę.

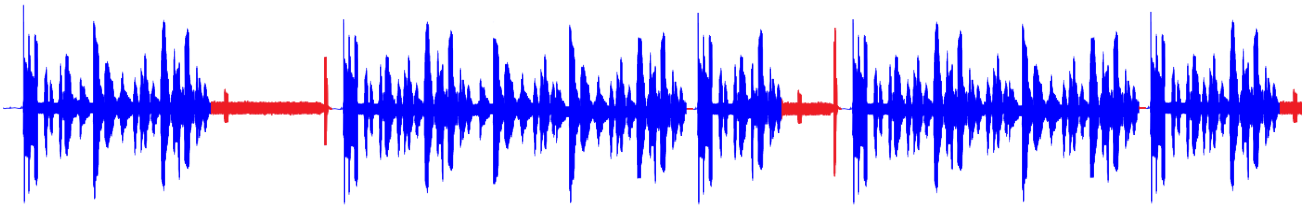
Tinklo protokolai yra puikiai tinkami slaptiems duomenims perduoti. Daugelis steganografinių metodų išnaudoja retai naudojamas protokolų antraštes [6]. Kiti išnaudojami antraščių laukai yra tokios atsitiktinių reikšmių antraštės kaip *IP Identification* ir *IP Initial sequence number*. Šiems laukams išnaudoti neužtenka aklai jų reikšmes pakeisti norima perduoti informacija. Atsitiktinių reikšmių laukų formavimas pasižymi savita specifika, kurią pažeidus galima nesudėtingai atskirti modifikuotus paketus.

Pirmasis mokslininkas, pateikęs mokslinių įrodymų dėl TCP/IP protokolo išnaudojimo steganografiniais metodais, buvo C. H. Rowlandas [5]. Talpos steganografiniai metodai yra lengviau įgyvendinami ir turi didesnį pralaidumo lygį, lyginant su laiko (4 pav.). Pastarųjų įgyvendinimas sinchronizuojant duomenų paketų srautą ir eliminuojant tinkle esantį triukšmą yra kur kas sudėtingesnis procesas.



4 pav. Steganografinis laiko kanalas

Nekorektišku atveju steganografiniai metodai yra naudojami paskirstytosios paslaugos nutraukimo atakoms (angl. *DDOS*) [7], platinant kompiuterių virusus ir kirminus, slaptai komunikacijai tarp teroristinių vienetų ir nusikaltėlių bei siekiant apeiti įmonės ugniasienę. Korektišku atveju steganografiniai metodai naudingi užtikrinant komunikuojančių pusių privatumą, didinant VoIP saugumą ir kokybę (angl. *QoS*) [8]. Jie naudojami tinklo srautams žymėti, šifruotoms atakoms ir anoniminiams VoIP skambučiams sekti [9]. E. Joneso [10] pasiūlytas metodas leidžia nustatyti IP įeigos tašką į tiriamo tinklo sritį naudojant steganografinį žymėjimo metodą IPv4 *TTL* antraštėje. Dalis šiuolaikinių lygiarangio ryšio (angl. *peer to peer; P2P*) paslaugų yra pažeidžiamos naudojant steganografinius metodus. Tai gali būti slaptų žinučių siuntimas „Skype“ [11], keičiant tylos metu



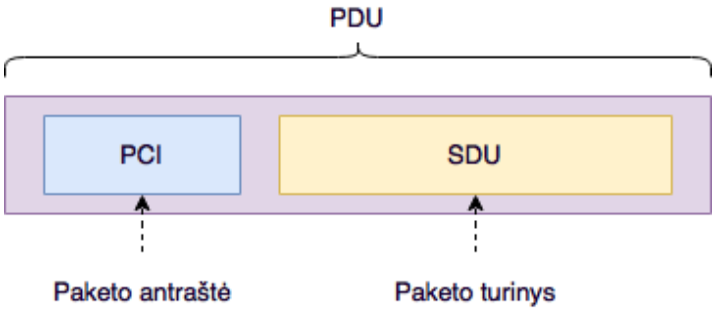
5 pav. Slaptų duomenų įterpimas balso sraute

perduodamus duomenis (5 pav.), „BitTorrent“ [12] atveju srauto tipas yra vienas-su-daugeliu, o jo naudojamo protokolo μTP antraščių laukai yra tinkami steganografiniams duomenims įterpti, „FreeWave“ [13] programine įranga moduliuojant interneto srautą į akustinius garsus ir juos perduodant VoIP kanalais (pvz., „Skype“, „Viber“, „WhatsApp“). Varšuvos universiteto mokslininkų buvo pristatytas steganografinis metodas kaip metodo dangą panaudojant „Google Suggest“ [14] paslaugą, o Hong Kongo universiteto mokslininkai įrodė galimybę slaptą informacijos kanalą užmegzti pasinaudojant tinklalapių lankytojų srauto skaitikliais [15].

1.2.1. TCP/IP steganografija

TCP/IP arba tinklo steganografiniai metodai skirstomi pagal PDU (angl. *protocol data unit*) modifikacijos būdą [16, 17]. Išskiriamos trys pagrindinės grupės (6 pav.): SDU (angl. *service data unit*) modifikavimas, PCI (angl. *protocol control information*) modifikavimas ir laiko intarpų modifikavimas tarp PDU. SDU modifikavimo atveju slapti duomenys įterpiami į perduodamų duomenų lauką. PCI modifikavimo atveju slapti duomenys įterpiami tarp OSI lygmenų perduodamų

SDU duomenų antraštėse. Modifikuojant PDU laiko intarpus, slapti duomenys yra išreiškiami sutartais laiko intarpais tarp duomenų paketų.



6 pav. PDU, PCI ir SDU

Kai kurie steganografiniai protokolai aptikimo atsparumui padidinti slaptus duomenis saugo keliuose skirtinguose TCP/IP lygiuose esančiuose protokoluose vienu metu. Pirmasis, pristatęs tarpsluoksninį steganografinį sprendimą, buvo B. Jankowskis [17]. Pasiūlytas metodas naudoja ARP, TCP, UDP ir ICMP, 1-ojo, 2-ojo ir 3-iojo lygmens TCP/IP modelio protokolus kartu išnaudojant *Etherneto* tinklo saugumo spragą, kuri dėl klaidos tinklo sąsajos tvarkyklėje (angl. *driver*) į specialiai sukonfigūruotą ICMP užklausą grąžina dalį prieš tai atakuojamoje mašinoje apdoroto tinklo srauto duomenų. Slaptos informacijos apsikeitimui suformuojamas nekorektiškai užpildytas *Etherneto* kadras. Užmegztas informacijos kanalas gali keisti tinklo protokolus, perduodančius slaptą informaciją, taip sumažindamas aptikimo tikimybę.

1.2.1.1. Kanalo lygis

Kanalo lygmenyje steganografinė duomenų slėpimo metodika išnaudoja fizines komunikacinio kanalo ypatybes ir imituoja jų netobulumus ar trikdžius. K. Szczypiorskis ir W. Mazurczykas pasiūlė metodą, skirtą belaidžiams IEEE 802.11 tinklams [18]. Jo veikimas paremtas slaptų duomenų įterpimu į perduodamų simbolių užpildo (angl. *padding*) dalį. Vėliau K. Szczypiorskis pasiūlė dar vieną metodą (taip pat belaidžiams tinklams) pavadinimu HICCUPS [19] (angl. *Hidden Communication System for Corrupted Networks*). Šio metodo idėja yra sugadinti perduodamų duomenų kadrų kontrolines sumas. Belaidžiuose tinkluose visi tinklo dalyviai girdi radijo bangomis perduodamus duomenis. Pagal RFC standartą kadrai su klaidingomis kontrolinėmis sumomis yra atmetami jau antrame OSI lygmenyje ir neperduodami į aukštesnius, dėl to apie atmestus kadrus galiniai vartotojai nieko neįtaria. Tačiau kada tokiu būdu siekiama perduoti slaptus duomenis, vartotojas nešalina ir skaito tokius duomenų kadrus, kurių kontrolinės sumos yra klaidingos.

Steganografiniai metodai taip pat gali būti realizuoti keičiant perduodamų duomenų formą (paketus ar kadrus). D. Kunduras ir K. Ahsanas [20] pasiūlė du būdus, tinkančius šiai tinklo steganografijos rūšiai. Pirmuoju atveju slaptųjų duomenų bitai rašomi į neišnaudotus arba rezervuotus duomenų paketų antraščių laukus. Tai yra įmanoma dėl daugelio populiarių tinklo protokolų nedokumentuotų atvejų, kaip turėtų būti užpildomi nenaudojami antraščių laukai. Konkrečiai pasiūlytame metode slaptai informacijai perduoti naudojami IP protokolo *DF* antraštė (angl. *Do not fragment*). Šiam metodui įgyvendinti būtina sąlyga – perduodamų duomenų paketų dydis, nesiekiantis maksimalaus paketų skaidymo dydžio (angl. *MTU – Maximum transfer unit*). *Etherneto* tinkluose šis dydis siekia 1500 B. Panašaus veikimo principo metodai taikomi ir kitoms skirtingų protokolų antraštėms – IP protokolo *ToS* antraštė [21], TCP protokolo *Flag* antraštė [22], IP protokolo eilės numerio antraštė [23] ir kt.

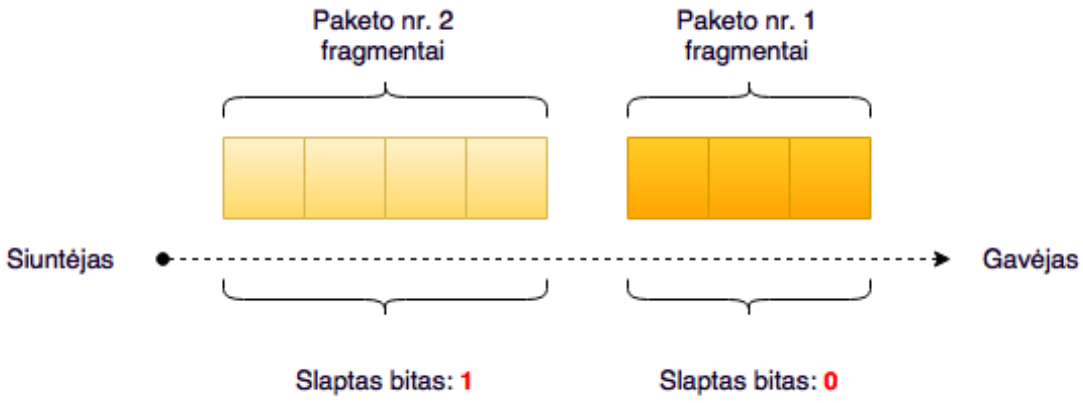
1.2.1.2. Tarptinklinis lygis (IP)

Dažniausiai IP protokolo steganografiniai metodai naudojasi tuo, kad IP antraštėse gausu laukų, kurie yra pertekliniai ar paprastai yra paliekami tušti perduodant duomenis. IP protokolo 4 versijoje tokie laukai yra *Identifiaction*, *Flags*, *Fragment Offset* ir *Options* (7 pav.). Pagrindinis šių metodų privalumas – didelė duomenų perdavimo talpa, tačiau trūkumas tai, kad slapti duomenys yra nesudėtingai eliminuojami. C. H. Rowlandas [24] slaptiems duomenis perduoti pasiūlė ASCII koduotės ženklus dauginti iš 256 ir talpinti 16-bitų *Identification* antraštės lauke. IP paketų fragmentavimo atveju gavėjas gaus pakartotines raides su kiekvienu duomenų fragmentu. Steganografiniai metodai, akcentuoti į IP fragmentavimą, duomenims perduoti naudoja *MF, DF ir FO* antraščių laukus. Šių laukų reikšmės reikalingos tik tuo atveju, jei duomenų paketai prieš perduodant yra suskaldomi, dėl to siuntėjui ir gavėjui žinant maksimalų duomenų paketo dydį, šiuos duomenų laukus galima panaudoti slaptiems duomenims perduoti. Duomenų perdavimo sparta 1-17 bpp. Kai komunikuojančios pusės neturi galimybės nustatyti maksimalaus duomenų paketo dydžio (angl. *maximum transmission unit*), jos gali naudotis metodu, duomenis įrašančiu į *identification* lauko reikšmės pirmuosius 8 bitus (formuojant paketą paskutiniai 8 bitai nustatomi atsitiktinai [25]) panaudojant XOR logiką tarp įrašomų duomenų ir jau esančios lauko reikšmės pirmųjų 8 bitų. Vienintelė sąlyga –duomenų paketo antraštės laukas *Options* turi būti tuščias.

<table>
<tr><td rowspan="2">versijos numeris</td><td rowspan="2">antraštės ilgis</td><td colspan="4">serviso tipas</td><td colspan="3" rowspan="2">bendras ilgis</td></tr>
<tr><td>PR</td><td>D</td><td>T</td><td>R</td></tr>
<tr><td colspan="5" rowspan="2">paketo identifikatorius</td><td colspan="3">požymiai</td><td rowspan="2">fragmento postūmis</td></tr>
<tr><td></td><td>D</td><td>M</td></tr>
<tr><td colspan="2">gyvavimo laikas</td><td colspan="3">aukštesnio lygmens protokolas</td><td colspan="3">kontrolinė suma</td></tr>
<tr><td colspan="8">IP siuntėjo adresas</td></tr>
<tr><td colspan="8">IP gavėjo adresas</td></tr>
<tr><td colspan="8">nustatymai ir užpildymas</td></tr>
<tr><td colspan="8">duomenys</td></tr>
</table>

7 pav. IP protokolo antraštės, tinkamos slaptiems duomenims įterpti

Steganografinių metodų autoriai – W. Mazurczykas ir K. Szczypiorskis – yra pristatę keletą metodų duomenims perduoti išnaudojant duomenų paketų dalijimo (angl. *segmentation*) savybę ir susijusius antraščių laukus. Duomenų paketas dalijamas į pasirinktą skaičių fragmentų (lyginis fragmentų skaičius reiškia dvejetainį skaičių nulį, o nelyginis – dvejetainį skaičių vienetą) (8 pav.); duomenų perdavimo sparta: 1 bpp; pritaikant reikšmes, esančias *Fragment Offset* lauke (lyginė reikšmė – dvejetainis skaičius vienetas, o nelyginė – nulis). Duomenų perdavimo sparta – 1 bpp; naudojant natūraliai susidariusius duomenų paketų fragmentus ir duomenis įterpiant į duomenų lauką. Duomenų perdavimo sparta – *fs*fd* bpp, kur *fs* – duomenų paketų fragmentų skaičius, *fd* – duomenų paketų fragmentų dydis. Šis metodas turi sąlyginai didelę duomenų perdavimo spartą ir yra sunkiau aptinkamas dėl naudojamų natūraliai susidariusių duomenų paketų fragmentų.



8 pav. Paketų segmentavimo metodas

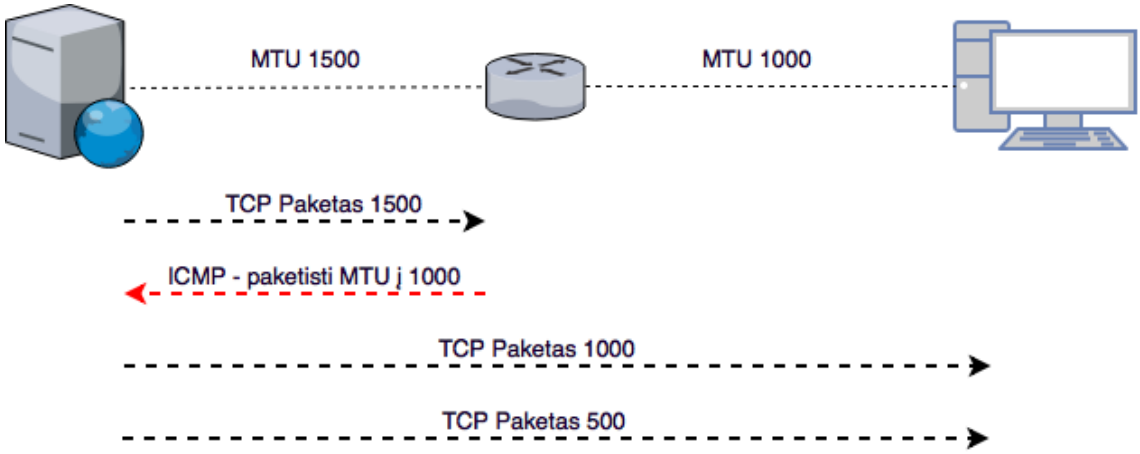
E. Cauicho pasiūlytame metode duomenims perduoti naudojami *Identification* ir *Fragment offset* laukai [26]. Duomenų perdavimo greitis nesudalintiems duomenų paketams – 29 bitai viename duomenų pakete, tačiau šis metodas veikia tik tarp greta esančių tinklo taškų. Pirmiausiai gavėjas pasitikrina, ar duomenų paketas yra nesuskaidytas (MF = 0), tuomet paketo antraštėje tikrina 3-jų rezervuotų bitų reikšmę, kuri gavėją informuoja apie slaptos informacijos egzistavimą. Slapti duomenys gaunami iš *Identification* bei *Fragment Offset* laukų.

Slaptai informacijai perduoti gali būti naudojami ir dinaminiai protokolo antraščių laukai, pavyzdžiui, *TTL* (angl. *time-to-live*). S. Zanderis pasiūlė patobulintą 1 bpp spartos steganografinį metodą [27]. Išanalizavęs standartines *TTL* reikšmes bei dažniausias reikšmes natūraliame duomenų sraute, šis mokslininkas pasiūlė naudoti dvi pradines *TTL* reikšmes – didelę ir mažą: viena reikšmė skirta atvaizduoti dvejetainiam vienetui, kita – dvejetainiam nuliui.

K. Ahsanas savo darbe pademonstravo, kaip įgyvendinti steganografinį *laiko* kanalą rūšiuojant duomenų paketus [28]. Šiam metodui įgyvendinti papildomai reikia žinoti pirminę paketų eiliškumo tvarką, pagal kurią būtų galimą korektiškai atkurti persiųstus duomenis. Pirminiam eiliškumui atkurti naudojama 32 bitų *Sequence Number* laukas *AH* antraštėje su *Option Data Lenght* ir *Option Data* laukų reikšmėmis, naudojamose *IPSec* protokole. Realiuose tinkluose, kur duomenų paketų pristatymo eiliškumas nėra garantuojamas, gali būti naudojamas eiliškumo apskaičiavimo algoritmas, pristatytas mokslininkų P. Pengo, P. Ningo ir D. Reeveso [29].

C. Abadas pademonstravo esminę IP protokolo dizaino klaidą [30], kai slapta informacija gali būti įrašoma į 16-bitų kontrolinės sumos antraštę pasinaudojant maišos funkcijos kolizijomis. Slaptos informacijos gavimui gavėjas turi žinoti pirminę TTL reikšmę.

Maksimalaus duomenų paketo dydžio (*MTU*) apskaičiavimui tarp skirtingų tinklų IPv4 tinkluose naudojamasi PMTUD (angl. *Path MTU Discovery*) (9 pav.) ir PLPMTUD (angl. *Packetization Layer MTU Discovery*) IPv6 tinkluose. Pirmuoju atveju yra siunčiami duomenų paketai su DF=1 ir tuo pat metu stebima ICMP protokolo teikiama informacija apie paketų pristatymą/praradimą. Antruoju atveju (IPv6) maksimalus duomenų paketo dydis apskaičiuojamas pradedant siųsti paketus nuo sąlyginai mažo dydžio ir jį palaipsniui didinant iki tol, kol duomenų paketai tampa per dideli ir nebepasiekia gavėjo. Gauti/negauti duomenų paketai analizuojami siuntimo lygmenyje be ICMP. Nustačius maksimalaus duomenų paketo dydį, IPv4 siuntėjas turi galimybę slaptą informaciją įterpti į pastarojo protokolo pranešimus, o gavėjui siųsti suklastotus ICMP pranešimus.



9 pav. Maksimalaus MTU apskaičiavimas

M. A. Padlipskis savo darbe pristatė *laiko* steganografinį kanalą, kuriame siuntėjas sutartais laiko intervalais siunčia arba nesiunčia duomenų paketų [31]. Sekundę padalinus į *n* laiko intervalų, gaunama slaptų duomenų perdavimo sparta *n* bps.

V. Berko metodas pagrįstas vėlavimais tarp siunčiamų duomenų paketų, kai laiko intarpai koduojami kaip slapta informacija [32]. Šioje sistemoje gavėjas seka laiko intervalus tarp gaunamų duomenų paketų. Ilgesnis nei vidutinis laiko intervalas yra interpretuojamas kaip dvejetainis skaičius vienetas, o trumpesnis – kaip dvejetainis skaičius nulis.

Aplikacijose kaip SSH kiekvienas paspaustas klaviatūros klavišas reiškia vieną išsiųstą duomenų paketą. Kontroliuodami paspaudimų greitį (intarpus tarp paspaudimų), gausime *pasyvų* slaptą informacijos kanalą.

S. Cabuko pristatytas metodas naudojasi realiais intervalų tarp duomenų paketų duomenimis. Šiuos rezultatus autorius padalija į dvi grupes: slaptos informacijos vienetas yra išreiškiamas kaip laiko intervalas, atitinkantis intervalą iš vienos grupės, ir informacijos nulis, kai laiko intervalas atitinka rezultatą iš kitos grupės [33].

H. Wango ir S. Floydo pristatyti modeliai atkartoja statistines realaus duomenų srauto charakteristikas, taigi dėl šios priežasties toks slaptų duomenų srautas yra neaptinkamas.

S. Gianvecchio metode aprašomas visas sprendimo karkasas (angl. *framework*): tai filtravimo, analizavimo, kodavimo bei siųstuvo moduliai [34]. Filtravimo modulis išrenka realų duomenų srautą iš turimų pavyzdžių duomenų bazės, analizavimo modulis pritaiko gautam duomenų srautui tinkamiausia modelį, tuomet, priklausomai nuo priskirto modelio kodavimo, modulis parenka deramą pasikarstymo funkciją iš turimų statistinių įrankių ir srauto generavimo bibliotekų, galiausiai siųstuvo modulis sugeneruoja slaptą duomenų srautą ir jį įterpia į natūralų srautą.

H. El-Sayedas pasiūlė išskirtinį steganografinį metodą kaip duomenų „nešėją“ naudojant *Traceroute* komandą su IP antraštėje nurodytu *Record Route* nustatymu [35]. Duomenų lauko ilgis – iki 40 baitų. I. Jawharo pasiūlytame metode aprašoma slapta FTP protokolo alternatyva – CFTP [36]. Šis metodas paremtas *Record Router* principu. CFTP protokolas teikia sesijos valdymo funkcionalumą, panašų į TCP, ir visa tai įgyvendinta IP protokolo *Options* antraštėje (10 pav.).

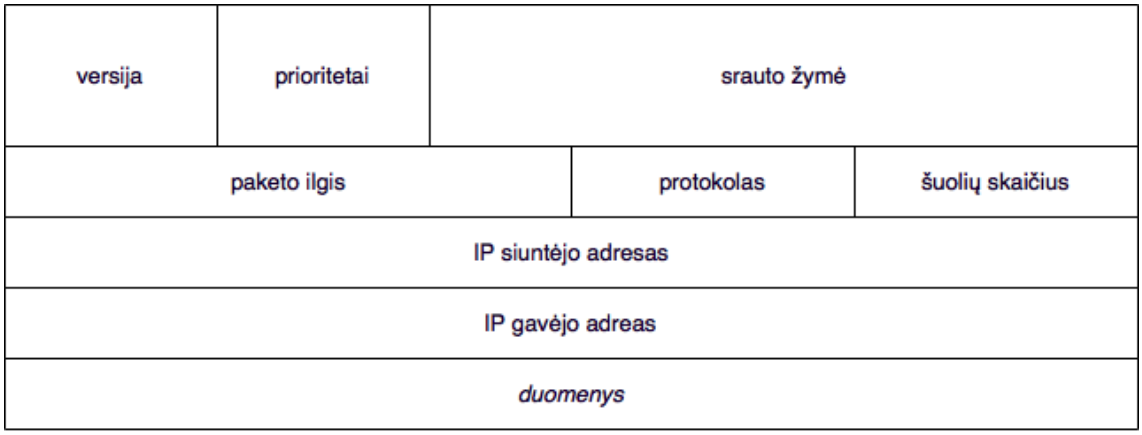
No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.16.3	172.16.16.20	ICMP	Information reply
2	0.002646	172.16.16.20	172.16.16.3	ICMP	Echo (ping) reply
3	0.002998	172.16.16.3	172.16.16.20	ICMP	Information reply
4	0.003357	172.16.16.20	172.16.16.3	ICMP	Timestamp reply
5	0.059727	172.16.16.3	172.16.16.20	ICMP	Echo (ping) reply
6	4.380337	172.16.16.3	172.16.16.20	ICMP	Echo (ping) reply
7	4.389121	172.16.16.20	172.16.16.3	ICMP	Echo (ping) reply
8	4.409872	172.16.16.20	172.16.16.3	ICMP	Information reply
9	4.410092	172.16.16.20	172.16.16.3	ICMP	Timestamp reply
10	4.410319	172.16.16.20	172.16.16.3	ICMP	Timestamp reply
11	4.410525	172.16.16.20	172.16.16.3	ICMP	Timestamp reply
12	4.410727	172.16.16.20	172.16.16.3	ICMP	Information reply
13	4.410927	172.16.16.20	172.16.16.3	ICMP	Information reply
14	4.411126	172.16.16.20	172.16.16.3	ICMP	Information reply
15	4.411326	172.16.16.20	172.16.16.3	ICMP	Echo (ping) reply

10 pav. CFTP programos slapto tinklo srauto vaizdas

T. Grafo metode analizuojamas duomenų kanalas realizuotas IPv6 protokolo *Destination Options* antraštėje [37]. Šiuo metodu perduodami duomenys yra koduojami *tipas-ilgis-reikšmė* principu (angl. *TLV, Type-Lenght-Value*) (11 pav.).

11 pav. TLV duomenų kodavimo būdas

N. B. Lucena savo darbe išanalizavo keletą steganografinių metodų, realizuotų IPv6 protokolo antraštėse [38]. Šiems metodams būdingas reikalavimas siuntėjui apskaičiuoti vientisumo patikrinimo vertę (angl. *ICV, Integrity Check Value*). Pagrindiniai IP protokolo duomenų paketų antraščių modifikavimo būdai: nustatant netikrą srauto tipą 8 bitų *Traffic Class* lauke; nustatant netikrą reikšmę 20 bitų *Flow Label* lauke; nustatant netikrą siuntėjo IP adresą 128 bitų *Source Address* lauke; nustatant *Hop Limit* reikšmę ir vėliau manipuliuojant kitais duomenų paketais. Šio metodo trūkumas – tai, kad duomenų paketai nebūtinai siunčiami tuo pačiu maršrutu, todėl tarpinių maršrutizatorių skaičius gali kisti. Duomenų perdavimo sparta 1 bpp, nustatant papildomą antraštę (12 pav.) 8 bitų *Next Header* lauke arba papildant persiunčiamus duomenis jų pabaigoje įterpiant slaptus duomenis.



12 pav. IPv6 antraštės struktūra

Nustačius netikrą maršrutizatoriaus pranešimą, perduodamų duomenų talpa – iki 2 baitų, kamšalo vietoje duomenų talpa – iki 256 baitų. Atveju, kai modifikuojami kiti *Option* antraštės laukai, perduodamų duomenų kiekis viename pakete gali siekti iki 2038 baitų; keičiant *Option Data Lenght* ir *Option Data* laukus *Hop-by-Hop Options* antraštėje; keičiant *Reserved* lauko reikšmę, perduodamų duomenų kiekis 4 bitai; keičiant adresus *Routing* antraštėje, nustačius *Routing Type* 0, perduodamų duomenų kiekis gali siekti iki 2048 bitų; nustatant 8 ir 2 bitų *Reserved* reikšmes, nustatant netikrą *Next Header* reikšmę arba įterpiant netikrą paketo fragmentą į *Fragment* antraštę. Pastaruoju atveju, norint nepažeisti siunčiamų duomenų ir kad netikras duomenų fragmentas nebūtų įtraukiamas į pirminius duomenis, autoriai siūlo du sprendimo būdus. Pirmasis – įterpiant netikrą *Identification* lauko reikšmę, dėl kurios netikras duomenų paketo fragmentas bus atmestas. Antrasis – nustatyti pasikartojančią *Fragment Offset* reikšmę. Tuomet duomenų rinkimo metu šis duomenų fragmentas bus perrašytas pirminio duomenų fragmento; nustatant 2 bitų *Reserved* lauko reikšmę arba sukuriant neegzistuojančią *Authentication Header* antraštę. Duomenų perdavimo sparta – iki 1022 bitų; sukuriant neegzistuojančią antraštę arba įrašant netikrą kamšalo reikšmę *Esp* antraštėje. Perduodamų duomenų sparta – nuo 255 iki 1022 bitų. Keičiant *Option Data Lenght* ir *Option data* laukų reikšmes bei sukuriant vieną ar kelis dirbtinius nustatymus *Options* antraštėje; nustatant netikrą kamšalo reikšmę *Destination Options* antraštėje.

P.Allix pristato pavyzdinę slapto kanalo schemą: duomenų siuntėjas kontroliuoja du tinklo įrenginius A ir B, kurie abu yra sujungti su bendru serveriu C. Atveju, kai A atsiunčia vieną duomenų paketą, o B atsiunčia du duomenų paketus, tai yra suprantama kaip slaptų duomenų dvejetainis nulis. Kai A atsiunčia du paketus, o B – vieną, tai suprantama kaip slaptos informacijos dvejetainis vienetas. P.Allix taip pat siūlo slaptų duomenų persiuntimą realizuoti nustatant sutartinį skaičių λ ir sekant persiųstų duomenų skaičių vienos sesijos metu. Kai persiųstų duomenų skaičius viršija arba yra lygus λ, tuomet tai interpretuojama kaip vienetas, jeigu persiųstų duomenų kiekis yra mažesnis nei λ, tuomet tai suprantama kaip dvejetainis nulis.

ICMP (angl. *Internet Control Message Protocol*) – interneto kontrolės žinučių protokolas. ICMP priklauso TCP/IP protokolų grupei ir priskiriamas OSI tinklo sluoksnio protokolams. Protokolas neturi jokio gavimo patvirtinimo funkcijos, todėl laikomas nepatikimu. Pagrindinė protokolo paskirtis – perduoti klaidos informaciją duomenų siuntėjui. Informacija dažniausiai perduodama tarp prie tinklo prijungto kompiuterio ir tinklinės įrangos (pvz., maršrutizatoriaus). ICMP pranešimai siunčiami apsupti (angl. *Encapsulated*) IP paketuose. ICMP turi 14 pranešimų tipų, kurių daugelis naudoja tik 4 bitus 8 bitų ICMP paketo antraštėje. *The Loki Project* atveju slaptas informacijos kanalas sudaromas

su *ICMP Echo Request* ir *ICMP Echo Reply* paketais. *Loki* kliento programa leidžia siuntėjui siųsti komandas į serverį per ICMP duomenų paketus, serveris savo ruožtu rezultatus gražina *ICMP Echo Reply* paketuose. Šio metodo įgyvendinimas yra sąlyginai paprastas ir veikia tinkluose, kuriuose nėra draudžiami ICMP pranešimai.

The Loki Project nėra vienintelis įgyvendintas duomenų tunelio per ICMP atvejis. *Skeeve* ir *ICMP-Chat* projektų duomenų perdavimo talpa – tarp 24-56 baitų. *Skeeve* atveju siuntėjas, kuris siekia išlikti anonimiškas, siunčia *ICMP Echo Request* duomenų paketą į tarpinį serverį, nustatydamas gavėjo IP adresą kaip siuntėjo. Tarpinis serveris, gavęs tokį pranešimą, *ICMP Echo Reply* duomenų paketą siunčia gavėjui. *ICMP-Chat* – tai terminalinė pokalbių sistema, naudojanti ICMP duomenų paketus komunikacijai. Duomenų konfidencialumui užtikrinti naudojamas AES šifravimo algoritmas. Slaptažodžių saugumui užtikrinti naudojamas SHA-256 maišos algoritmas. Papildomas funkcionalumas, leidžiantis keisti ICMP pranešimų kodus, sumažina aptikimo tikimybę.

D. Stodle pristatė *PING Tunnel* projektą, kuris leidžia patikimai užmegzti ir palaikyti TCP sesiją ICMP tunelio režime.

Įrankis *V00D00NET* IPv6 ir ICMPv6 skirtas slaptų duomenų kanalų sudarymui. Komunikacijai užmegzti siuntėjas ir gavėjas naudoja keturių skaitmenų PIN numerį – tai leidžia užmegzti kelis duomenų tunelius vienu metu.

Mokslininkai B.Ray ir S.Mishra savo moksliniame darbe pademonstravo galimybę slapta ir saugiai perduoti didelius kiekius informacijos pasinaudojant *ICMP Echo Request* duomenų paketais. Vienas iš didžiausių tinkamų naudoti laukų ICMP duomenų pakete yra 32 bitų talpos *Reserved* laukas *ICMP Router Solicitation* pranešime.

Interneto narystės grupėje protokolas (IGMP) – ryšio protokolas, kuris skirtas valdyti prisijungimą prie IP daugiaadresio transliavimo grupių. Taigi interneto narystės grupėje protokolas leidžia mazgui prisijungti prie bendros grupės, kuri gauna informaciją, arba nuo jos atsijungti. C.Scott siūlo duomenims perduoti naudoti *Reserved* 8 ir 16 bitų laukus *IGMPv3 Membership* pranešimuose. Šie laukai paprastai nėra užpildyti ir neturi įtakos gavėjui. Vienas iš įgyvendintų IGMP/IPv4 slaptų duomenų tunelių yra pavadinimu *B0CK*. Šis metodas naudoja IP paketo *Source Address* lauką slaptiems duomenis perduoti.

Nors *B0CK IP Protocol* lauke nustato IGMP reikšmę, tačiau pačiame pakete nėra apimtos (angl. *Encapsulated*) IGMP antraštės, paketo viduje vietoj jos yra įterpiamas 124 bitų kamšalas po 20 bitų IP antraštės.

DHCP protokolas realizuoja patikimą ir paprastą TCP/IP tinklo konfigūravimo būdą, garantuodamas adresų konfliktų nebuvimą dėl centralizuoto jų paskyrimo valdymo. Administratorius valdo adresų paskyrimo procesą parametru „nuomos trukmė“ (angl. *Lease Duration*). Pastarasis rodo, kiek ilgai kompiuteris gali naudotis paskirtu IP adresu iki kitos užklausos DHCP serveriui dėl IP adreso nuomos.

DHCP protokolo veikimo iliustracija gali būti situacija, kai kompiuteris, esantis DHCP klientu, yra pašalinamas iš potinklio. Tam įvykus, jam priskirtas IP adresas yra automatiškai atlaisvinamas. Kai kompiuteris yra prijungiamas prie kito potinklio, naujas adresas jam yra priskiriamas automatiškai. Nei vartotojas, nei tinklo administratorius nedalyvauja šiame procese. Šita savybė yra labai svarbi mobiliems vartotojams. DHCP protokolas naudoja kliento-serverio modelį. Sistemai startuojant, DHCP kompiuteris-klientas, esantis inicializavimo būsenoje, siunčia užklausą (angl. *discover*), kuri yra plačiai (angl. *broadcast*) išplatinama lokaliame tinkle, ir yra perduodama

visiems tinklo DHCP serveriams. Kiekvienas DHCP serveris, gavęs tokį pranešimą, atsako pranešimu (angl. *offer*), kuris suformuotas iš IP adreso ir konfigūracijos informacijos. R.Rios DHCP protokole išskyrė pozicijas, tinkamas steganografinių duomenų perdavimui:

- Nustatant 32 bitų *XID* lauko reikšmę. Ši reikšmė paprastai yra automatiškai atsitiktinai sugeneruojama. Šiuo lauku perduodama informacija priskiriama prie sunkiau aptinkamos, tačiau turi ribotą slaptų duomenų perdavimo spartą.

- Naudojantis *SECS* lauko reikšme. Šiuo atveju duomenų perdavimo sparta 1 bpp.

- Išnaudojant *CHADDR* lauko paskutinius 10 baitų, kai 48 bitų ilgio *Ethernet MAC* adreso laukas turi nustatytą reikšmę.

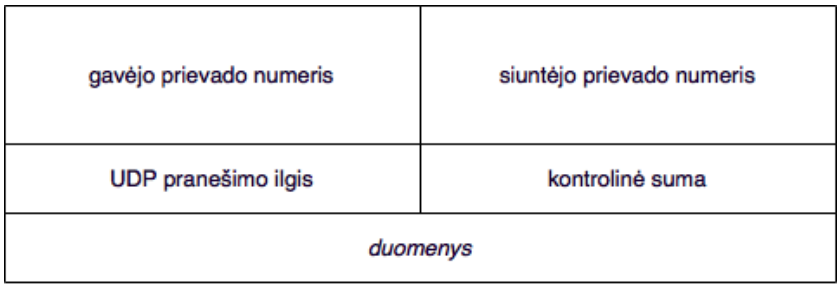
- 64 baitų *Sname* ir 128 baitų *File* laukuose įrašytos reikšmės eilutės pabaigai pažymėti naudoja nulinį *(*angl. *null)* baitą. Slapta informacija, įrašyta po nulinio baito, bus sėkmingai perduota gavėjui ir tai niekaip nepaveiks kitų klientų, gavusių tokius duomenų paketus, kadangi informacija po nulinio baito nebus skaitoma. Kai laukuose nerašoma informacija, operacinė sistema jos reikšmę pakeičia į nulį, tačiau atvejais, kai naudojama *Overload* reikšmė, *Options* antraštėje šis laukas gali būti panaudojamas perteklinės informacijos įrašymui.

- Naudojant nestandartinių ilgių laukų reikšmes ar nestandartinį kiekį *Options* antraštės įrašų, galima aptikti nesudėtingos slaptos informacijos.

ARP-tinklo lygmens protokolas, skirtas IP adresų (tinklo lygmens adresų) susiejimui su MAC adresais (kanalo lygmens adresais) TCP/IP tinkluose. ARP – labai paplitęs ir labai svarbus protokolas, užtikrinantis ryšį tarp siuntėjo ir gavėjo. Kiekvienas tinklo mazgas turi du adresus: fizinį (MAC adresą) ir loginį (IP adresą). *Ethernet* tinkluose ryšio užmezgimui turi būti identifikuoti siuntėjo ir gavėjo duomenys. Tam naudojami fizinis ir loginis adresai. Informacija siunčiama nuo vieno mazgo prie kito, kurie identifikuoja siuntėjo ir gavėjo fizinius (MAC) ir loginius (IP) adresus.

1.2.1.3. Perdavimo lygis (TCP/UDP)

UDP protokolas (13 pav.) – tai pats paprasčiausias TCP/IP modelio transporto lygmens protokolas. Skirtingai nei TCP, UDP nėra patikimas, neatlieka duomenų tėkmės kontrolės (angl. *flow-control*) ir neturi klaidų atitaisymo mechanizmų. UDP paketai (datagramos) gali būti atsiųsti ne ta tvarka, kuria buvo siunčiami. Kartą siųstas paketas gali ateiti du ir daugiau kartų arba ir neateiti visai. Datagramos ilgis yra ribotas. Tačiau trumpa UDP žinutė persiunčiama greičiau nei per TCP, nes nereikalingi paketai sesijai pradėti, taip pat siunčiama mažiau antraštės duomenų. UDP protokolas tinkamas duomenų perdavimui tada, kai dalis prarastų duomenų nedaro įtakos programos veikimui (pvz., internetinėje telefonijoje ar televizijoje).



13 pav. UDP protokolo antraštė

TCP yra vienas iš pagrindinių protokolų TCP/IP tinklo modelyje (14 pav.). Priešingai, nei UDP, šis protokolas užtikrina patikimą duomenų perdavimo kanalą. Tinkle prarasti duomenų paketai yra pakartotinai išsiunčiami. Patikimo duomenų kanalo užmezgimui ir palaikymui reikalingi papildomi duomenų paketai, o dėl to sulėtėja duomenų perdavimo greitis, lyginant su UDP.

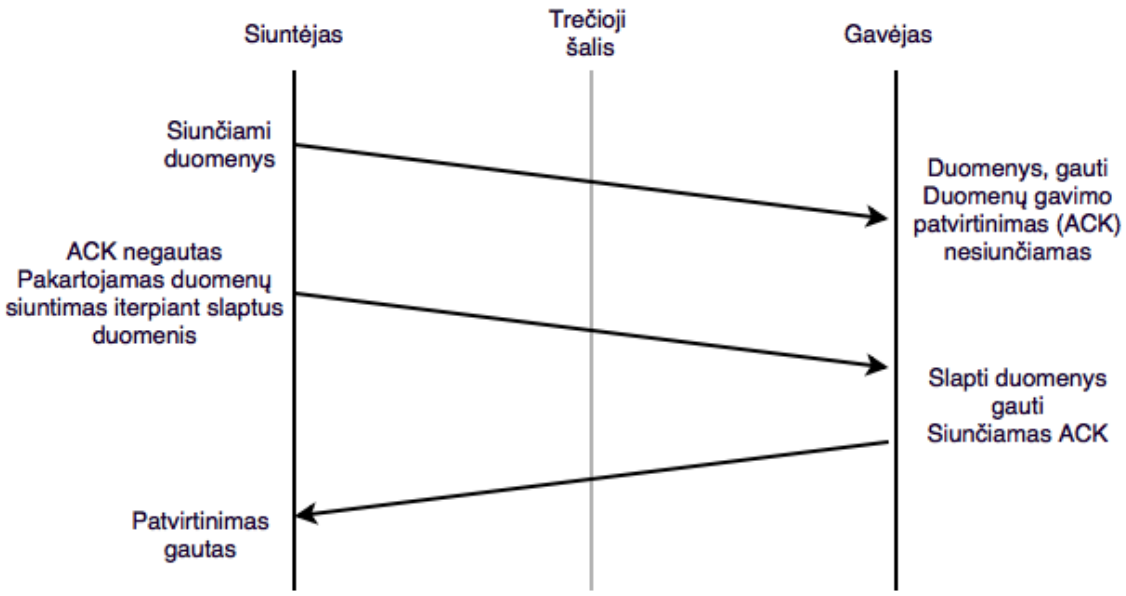
<table>
<tr><td colspan="5">siuntėjo kanalo numeris</td><td colspan="4">gavėjo kanalo numeris</td></tr>
<tr><td colspan="9">sekos numeris</td></tr>
<tr><td colspan="9">patvirtinimo numeris</td></tr>
<tr><td>antraštės ilgis</td><td>rezervas</td><td>URG</td><td>ACK</td><td>PSH</td><td>PST</td><td>SYN</td><td>FIN</td><td>langa dydis</td></tr>
<tr><td colspan="6">kontrolinė suma</td><td colspan="3">skubumas</td></tr>
<tr><td colspan="6">pasirinktys</td><td colspan="3">užpildymas</td></tr>
<tr><td colspan="9">duomenys</td></tr>
</table>

14 pav. TCP protokolo antraštė

W. Mazurczyko pasiūlė idėją slaptus duomenis siųsti standartiniame duomenų lauke duomenų paketuose, kuriuos gavėjas pareikalauja atsiųsti pakartotinai [39] (angl. *retransmision*). Pastarasis metodas skirtas tinklo protokolams, turintiems retransliavimo funkciją (15 pav.).

C. H. Rowland savo pristatytame metode slaptiems duomenims perduoti naudojo 32 bitų ilgio *Initila sequence number (ISN)* bei *acknolage sequence number* (ACK) laukus TCP protokolo antraštėje. J. Rutkowska, remdamasi C. H. Rowlando metodu, pristatė savo metodą, kuris duomenims perduoti negeneruoja papildomo srauto, o naudoja natūraliai susigeneravusį [40]. Siuntėjas keičia *ISN* ir *ACK SYN* laukų reikšmes, kurios buvo sugeneruotos ir tuose laukuose buvo įrašytos operacinės sistemos. S. Lewisas išvystė metodą *ISN* skaičių generavimui *Linux* operacinei sistemai, kuri būtų neatskiriama nuo natūraliai sugeneruotos operacinėje sistemoje [41].

J. Giffinas pristatė slaptą informacijos perdavimo būdą TCP protokolo laiko žymose (angl. *timestamps*) keičiant mažiausiai reikšmingą bitą [42]. Šis metodas visą duomenų srautą sulėtina tam, kad kai duomenų paketai su pakeistomis laiko žymomis pasieks gavėją, laiko žyma būtų galiojanti. Duomenų perdavimo sparta – 1 bpp. Dar vienas būdas informaciją perduoti lėtuose tinkluose yra lyginant perduodamos žinutės paskutinį nepersiųsta bitą su siunčiamos žinutės laiko žymos paskutiniuoju bitu. Jeigu bitai sutampa, duomenų paketas siunčiamas nemodifikuotas, jei ne – palaukiama, kol laiko žyma padidėja vienetu, ir tada išsiunčiama. Šio metodo aptikimas lėtuose tinkluose yra sudėtingas dėl atsitiktinių žemiausių bitų pasiskirstymo. R.C. Chakinalos ir kt. pasiūlytas modelis naudoja *laiko* slaptą informacijos kanalą keičiant duomenų paketų segmentų eiliškumą [43].



15 pav. Retransliacijos metodo veikimo schema

X. Luo įgyvendino *talpos* slaptą informacijos kanalą modifikuodamas *ACK* lauko reikšmę [44]. To paties autoriaus pristatytas *Tcpleaks* metodas slaptą informaciją įrašo į vienos arba kelių skirtingų sesijų *TCP ACK* duomenų paketus. Pastarasis metodas gali būti naudojamas su jau sudarytomis/veikiančiomis *TCP* sesijomis.

Cloak metodas priskiriamas prie sunkiai aptinkamų steganografinių metodų [45]. Duomenų kodavimui naudojami 10 skirtingų algoritmų bei pritaikoma atkartoti natūralaus TCP srauto charakteristikas. Siuntėjas gali užmegzti kelias HTTP sesijas su nutolusiu serveriu, taip pat tai gali daryti su keliais serveriais vienu metu. Tinklo stebėjimo įrangai slaptą informaciją tokio tipo sraute aptikti sudėtinga, kadangi dažnu atveju vartotojas natūraliai palaiko kelias TCP sesijas vienu metu.

TCPScript metode realizuojamas *laiko* slaptas informacijos kanalas, kuris slaptą informaciją įterpia į natūralų TCP srautą [46] bei išnaudoja TCP funkcijas, skirtas protokolo patikimumui užtikrinti (pavyzdžiui, ACK/SYN) siekiant padidinti koduotos informacijos tikslumą. Slapta žinutė išreiškiama kaip teigiamų sveikųjų skaičių masyvas *mi*, kur $mi \in [1,M]$ ir *M* yra prieš kodavimą siuntėjo ir gavėjo apsikeista konstanta. Šiuo metodu siunčiami duomenys nėra pažeidžiami tinklo trikdžių, paketų eilės pakeitimų ir paketų praradimų. Tačiau šio metodo perduodamų duomenų sparta yra žema ir tinklo srauto analizės metu steganografinis srautas išsiskiria iš natūralaus srauto.

W. Mazurczykas pritaikė *retransliacijos* metodą TCP protokolui [47]. *Retransliacijos* metodo idėja – atmesti gautą paketą siekiant iššaukti paketo pakartojimą. Į pakartotinai siunčiamą duomenų

paketo duomenų lauką vietoj esančių duomenų įrašoma slapta informacija. Siekiant išvengti slapto duomenų srauto aptikimo, pakartojimų santykis netūrėtų viršyti vidutinio pakartojimų skaičiaus tinkle. Autorius slaptų duomenų segmentų žymėjimui naudoja maišos funkciją. Šio metodo trūkumas yra tai, kad paketų retransliacijos santykis negali viršyti natūralaus srauto retransliacijų santykio, o tai gali būti labai sudėtinga vidiniuose tinkluose.

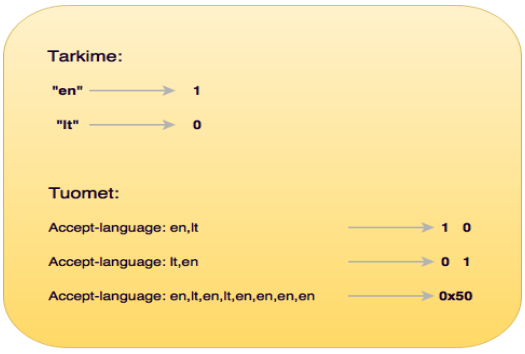
UDP slaptas informacijos kanalas gali būti sudaromas nustatant *Checksum* lauko reikšmę, taip pat kanalą, galimą užmegzti sutartinai nustatant tuščią lauko reikšmę, kadangi šio lauko užpildymas nėra privalomas. Duomenų perdavimo sparta – 1 bpp. Šiame metode aprašomi informacijos perdavimo principai gali būti panaudoti tiek UDP, tiek TCP protokoluose. Papildomai UDP protokole informacijai perduoti gali būti naudojami ir kiti laukai – *Source Address* ir *Source Port*. Visų aptartų laukų bendra duomenų perdavimo sparta gali siekti iki 6 baitų viename duomenų pakete.

O. I. Abdullazizo pasiūlytame metode slaptiems duomenims perduoti naudojami natūralūs realaus laiko komunikacijų (pvz., *VoIP Skype*) duomenys [48]. Slaptus duomenų bitus reprezentuoja UDP paketo dydis. Lyginis paketo dydis – slaptas duomenų bitas 1, nelyginis duomenų paketo dydis – slaptas duomenų bitas 0. Tokių slaptai perduodamų duomenų aptikimas ypač sudėtingas dėl naudojamo neišsiskiriančio duomenų srauto.

1.2.1.4. Taikomasis lygis (HTTP, DNS, FTP)

Keletas ar net keliolika TCP/IP aplikacinio lygmens protokolų yra tinkami slaptos informacijos perdavimui. Minimi šie protokolai: HTTP, DNS, FTP, VoIP, SIP, SMTP ir kt.

HTTP – pagrindinis metodas pasiekti informaciją pasauliniame tinkle (WWW). Pradinė protokolo paskirtis – pateikti standartinį būdą HTML puslapių skelbimui ir skaitymui. A. Dyatlovo darbe negarinėjama galimybė slaptą informaciją įterpti į HTTP užklausų antraštes (16 pav.) ar/ir kūno (angl. *body*) dalį [49]. Pats HTTP protokolas nelimituoja antraščių dydžio, tačiau visų protokolo antraščių suma negali viršyti platformos, kurioje yra realizuotas HTTP protokolas, nustatytos maksimalios reikšmės. *Apache* maksimalus antraščių dydis – 8 KB, *Microsoft IIS* – nuo 8 KB iki 16 KB priklausomai nuo naudojamos versijos.



16 pav. Steganografinis HTTP metodas

TCP/IP modelio aplikaciniame lygmenyje realizuotas W. Mazurczyko SIP protokolo steganografinis metodas, siūlantis duomenis siųsti per neišnaudotus protokolo laukus [50]. W. Benderis ir kt. siūlo steganografinio metodo slaptus duomenis įterpti į mažiausiai reikšmingus duomenų bitus garso ir vaizdo failuose [51]. M. V. Horenbeecko metodas slaptus duomenis talpina HTTP protokolo antraštėse [52].

M. Baueris pristatė savo protokolą, paremtą HTTP protokolu, skirtą slaptam informacijos kanalui užmegzti pasinaudojant įprastinių vartotojų naršymo srautu. Siūlomas protokolas naudoja penkis HTTP/HTML mechanizmus: peradresavimus, slapukus, nurodančias antraštes (angl. *referer headers*), HTML elementus bei aktyvų turinį (pvz., flash ar java) [53].

Z. Kweckos metode duomenims perduoti išnaudojami HTTP protokolo antraštėje esantys paeiliui einantys tarpai (angl. *white-space*), kurie interpretuojami kaip vienas tarpas. Tai gali būti *space* tarpas arba *tab* tarpas. Tokiu atveju informacijai atvaizduoti *space* gali būti pasitelkiamas kaip vienetas, o *tab* – kaip nulis, ir atvirkščiai. HTTP protokolo antraštėms nėra nustatytas specifinis eiliškumas, dėl to pakeistą antraščių išdėstymo tvarką galima panaudoti kaip informacijos perdavėją. Antraščių pavadinimų didžiosios ir mažosios raidės interpretuojamos kaip vienas ir tas pats, dėl to nustatant skirtingus raidžių dydžius galima perduoti slaptą informaciją [54].

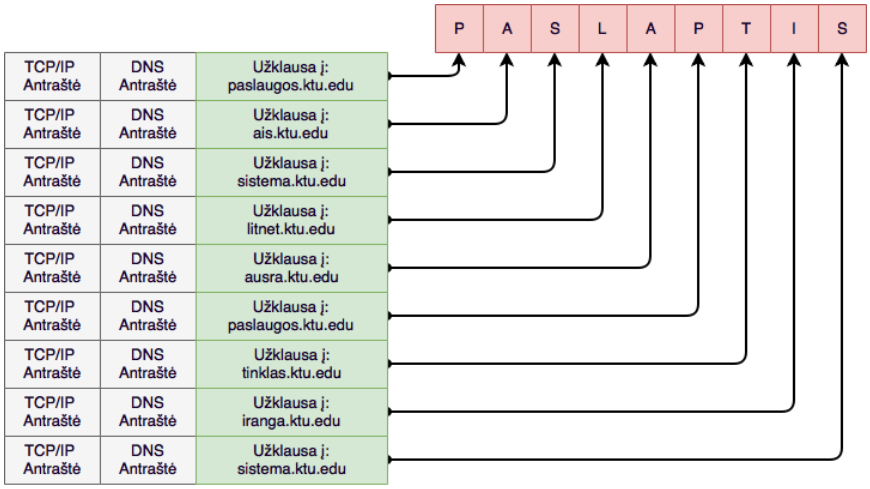
M.V. Horenbeecko įgyvendintame metode, sudarančiame komunikacinį tunelį, panaudojamos *Http Entity* žymų *Etag* ir *If-non-match* antraštės. Šios antraštės naudojamos kliento naršyklei patikrinti, t. y. ar vietinė tinklalapio kopija vis dar galioja. *Content-md5* antraštė gali būti naudojama duomenims su 128 bpp duomenų perdavimo sparta perduoti. Panašiu principu duomenys, perduodami panaudojant *Access-control-allow-origin* ir *Content-location* antraštes, nagrinėjami kituose darbuose. Kitas būdas informacijai įterpti galimas moduliuojant laiko antraščių reikšmių mažiausiai reikšmingus bitus. Šiam metodui realizuoti tinkamos antraštės *Date* ar *Last-modified*.

Dėl interneto turinio cenzūravimo ir naršymo istorijos stebėjimo kai kuriose šalyse buvo pristatyta *Infranet* infrastruktūra [55]. Šios paslaugos serveriai priima HTTP užklausas su cenzūruojamų puslapių adresais, užkoduotais įtarimo nesukeliančiose antraštėse. Serveriai savo ruožtu cenzūruotą turinį grąžina stenografiniais metodais įterptą į įtarimo nesukeliančius grafinius elementus.

FTP – tai failų perdavimo standartas, kuriam reikalinga speciali programinė įranga, vadinama *FTP klientu*. FTP veikia klientas–serveris principu, t. y. klientas siunčia tam tikras užklausas, o serveris jas interpretuoja ir atlieka tai, kas paprašyta. Visa tai vyksta naudojant TCP/IP ryšio protokolą. Failų perdavimą iš vienos sistemos į kitą realizuoja dvi programos, naudojančios FTP protokolą: programa–serveris, kuri veikia nutolusioje sistemoje, ir programa–klientas, kuri aptarnauja vartotoją, norintį atsisiųsti/patalpinti bylas nutolusioje sistemoje. FTP protokolas veikia dažniausiai per 20-21 prievadus. Pirmasis naudojamas duomenų siuntimui, o antrasis – komandų perdavimui į serverį.

X. Zou savo apžvalgoje pristatė du metodus, skirtus duomenų kanalo sudarymui per FTP [56]. Pirmasis slaptus duomenis koduoja komandas siųsdamas į FTP serverį. Duomenų perdavimo sparta *log2n*, kur *n* – komandų skaičius. Antrasis naudojasi FTP *Noop* komanda, kuri siunčiama, kai serveryje nevyksta komunikacija (angl. *idle*). Persiųstų *Noop* komandų kiekis interpretuojamas kaip sveikasis slaptos žinutės skaičius. Skirtingoms skaitinėms reikšmėms atskirti siunčiama *Abort* komanda. FTP protokolo sesijos palaikymui toks *Noop* ir *Abort* komandų siuntinėjimas yra įprastas.

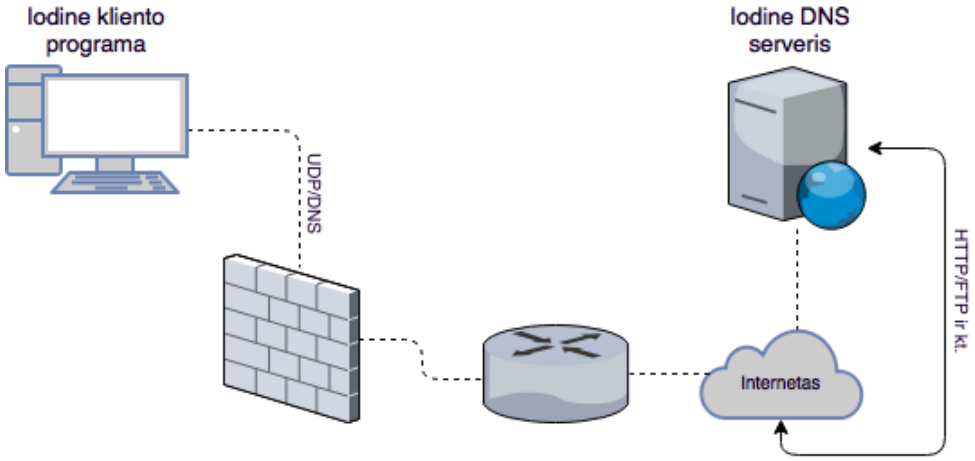
DNS – tai protokolas, skirtas simbolinių vardų vertimui į skaitinius. DNS leidžia kreiptis į tinklo resursus lengviau įsimenamu simboliniu vardu. DNS protokolo architektūra yra labai tinkama slapto informacijos kanalo sudarymui tuneliavimo režime tokiais protokolais kaip IP/TCP/UDP (17 pav.). Ypač didele talpa pasižymi DNS protokolo *NS*, *Cname* ir *TXT* įrašai, kurių ilgis siekia iki 255 baitų. Eksperimentinė *NULL* įrašo reikšmė gali siekti iki 65536 baitų.



17 pav. DNS protokolu perduodami slapti duomenys

C kalboje yra realizuoti du IPv4 per DNS metodai: tai *Nstx* [57] ir *Iodine* [58] (18 pav.). Abiem atvejais IP paketai yra suskaldomi ir išsiunčiami, galiniame tinklo taške surenkami atgal į pilną duomenų paketą. Duomenų kodavimui į užklausas *Nstx* metode naudojamas *Base64* algoritmas.

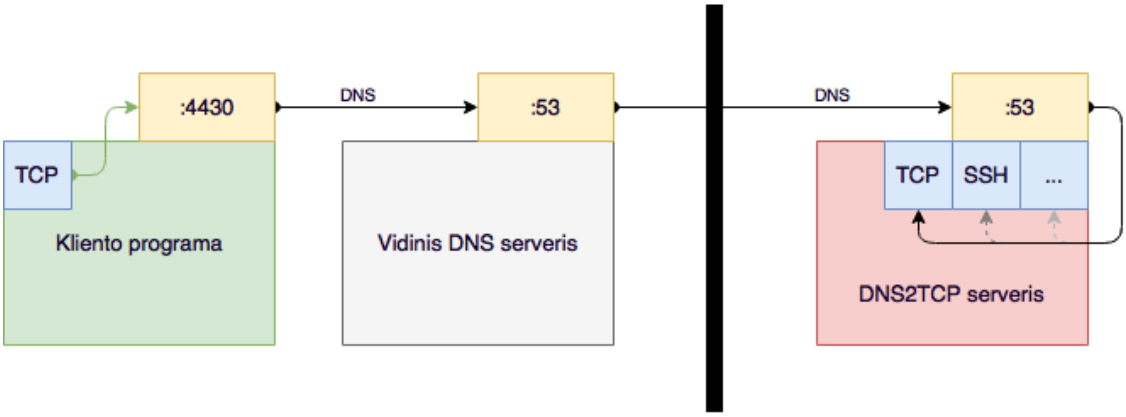
Užklausos perduodamos *TXT* įrašuose. *Iodine* metodas palaiko DNS protokolo *Edns0* išplėtimą, kuris leidžia siųsti didesnius DNS paketus. Išplėstinio DNS paketų ilgis iki 512 baitų. Duomenų atsiuntimui šis metodas naudoja *Null* įrašus, o išsiunčiamus duomenis suspaudžia *Gzip* formatu, koduoja *Base32* arba *Base64* algoritmais ir perduoda juos *Domain name* lauke.



18 pav. Iodine įrankio sujungimo schema

Java programavimo kalboje realizuotas *Dnscat* įrankis skirtas transliuoti IPv4 srautą per DNS. Duomenų perdavimui naudojami *Cname* įrašai [59].

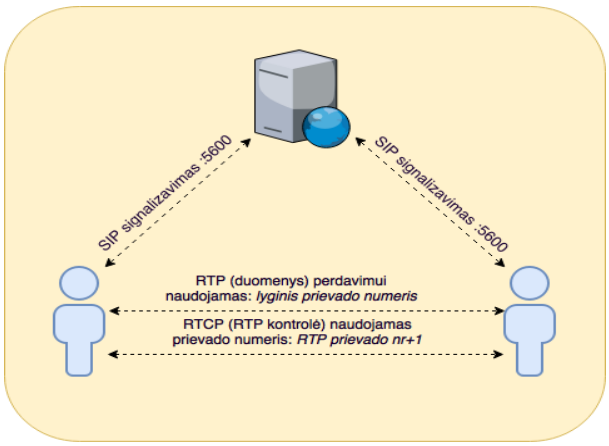
Ruby programavimo kalboje realizuotas duomenų paketų neskaidantis IPv4 per DNS įrankis, naudojantis *Cname* įrašus, koduotus *Base32* algoritmu [60]. Toks duomenų kodavimo algoritmas yra įprastas DNS siunčiamuose duomenyse, todėl toks slaptas turinys yra sunkiau aptinkamas. Pastarasis metodas pasikartojančių užklausų problemai išspręsti naudoja spartinimą (angl. *caching*).



19 pav. Dns2tcp įrankio veikimo schema

Dns2tcp įrankyje realizuotas metodas leidžia siųsti TCP ar SSH duomenų srautą per DNS panaudojant *Txt* įrašus [61] (19 pav.). Šiuo metodu perduodamų duomenų iššūkis yra patikimo duomenų kanalo užtikrinimas naudojant nepatikimą duomenų perdavimo protokolą (UDP).

Dar viena įdomi slaptų duomenų kanalų sritis yra realaus laiko aplikacijų duomenų perdavimas per IP protokolą: *Voice over IP* (VoIP), aukštos raiškos vaizdų transliacijos ir kt. Šiose aplikacijose vaizdo ir garso perdavimas paprastai realizuotas dviem skirtingais kanalais: naudojant RTP protokolą kartu su RTCP kontroliniu protokolu (20 pav.). Viena RTP protokolo dalis veikia aplikaciniame lygmenyje, tuo tarpu kita – perdavimo virš UDP. VoIP protokolo signalizavimo fazėje (prieš pradedant transliuoti garsą) naudojamas SIP protokolas. SIP savo ruožtu naudoja protokolus (pvz., SDP) sesijų valdymui [62].

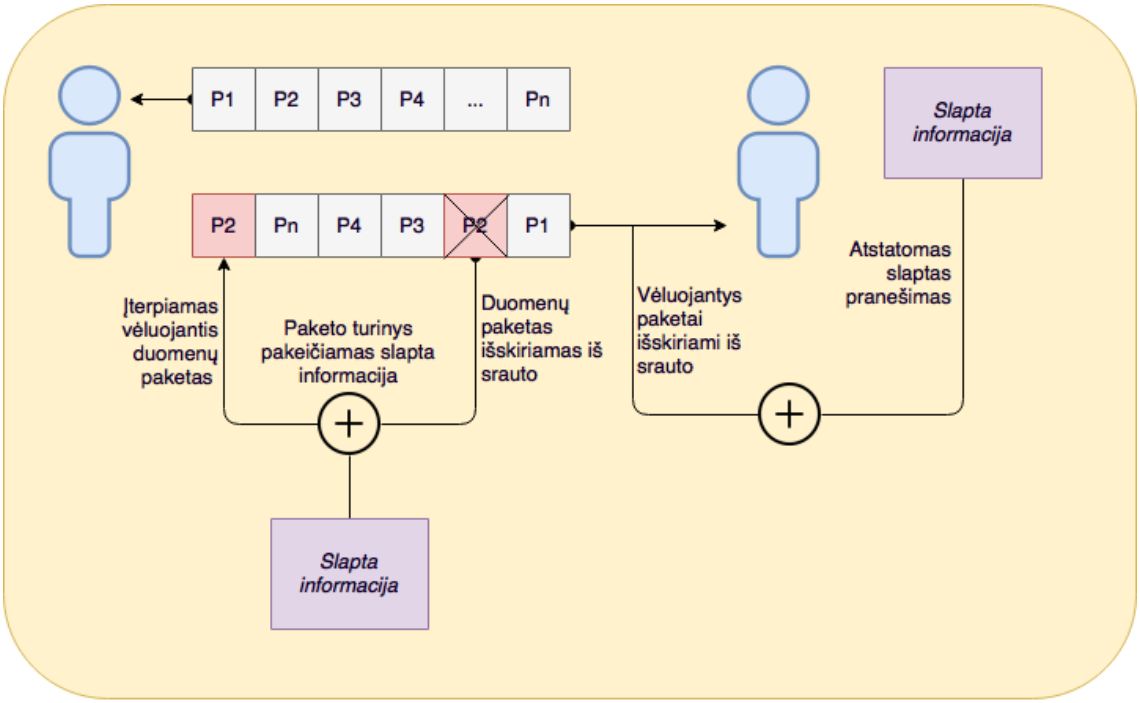


20 pav. VoIP sujungimo schema

RTP – realaus laiko sistemų protokolas. Tai *IETF* standartizuotas protokolas (RFC 1889), skirtas optimaliam garso ir vaizdo perdavimui paketinės komutacijos kanalais, negarantuojančiais informacijos pristatymo. Su šiuo standartu susijęs informacijos, perduodamos realaus laiko ryšio sistemomis, kontrolės protokolas RTCP. RTCP garantuoja grįžtamąjį ryšį tarp siuntėjo ir gavėjo. RTCP atlieka kelias pagrindines funkcijas: pateikia informaciją apie duomenų perdavimo terpės

kokybę; organizuoja papildomą RTP sesijos identifikavimo mechanizmą; reguliuoja generuojamus duomenų srautus ir minimaliai valdo RTP sesiją.

W. Mazurczykas detalizuoja slaptų informacijos kanalų kūrimą RTP protokole [62]. Duomenims perduoti naudojama 8 bitų ilgio *padding* lauko reikšmė, *extension* antraštė, atsitiktinai sugeneruota 16 bitų *sequence number* lauko reikšmė ir *timestamp* lauko 32 bitų reikšmė. Pastarasis



21 pav. Vėluojančio RTP paketo metodas

metodas gali būti taip pat naudojamas slaptos informacijos įterpimui RTCP protokolo *ntp timestamp* antraštės lauke. 160 bpp spartos informacijos kanalas gali būti sudaromas RTCP protokolo *receiver report* ir *sender report* duomenų paketuose. SRTP protokolo *authentication tag* talpinama informacija gali siekti iki 80 bitų. Taip pat šiame darbe siūlomas *lost audio packets* steganografinis metodas (21 pav.). Šis metodas veikia vėlinant paketus, o dėl to jie būna prarandami. Tradicinėje realaus laiko aplikacijų sistemoje vėluojantys paketai nėra analizuojami ir yra atmetami, tačiau šiame metode sistema šiuos paketus priima ir iš jų išgauna slaptą informaciją. Prarastų/žymiai vėluojančių paketų skaičius netūrėtų viršyti maksimalaus IP telefonijai nustatyto skaičiaus.

L. Y. Bai savo pristatytame metode naudoja RTCP protokolo 32 bitų *interarrival jitter* lauką slaptam informacijos perdavimui [63]. Principas veikia dviem aspektais: pirmoje dalyje renkama tiriamojo tinklo srauto statistika apie *jitter* lauko reikšmes. Antroje dalyje slapta informacija moduliuojama su esama *jitter* lauko reikšme, pasinaudojant statistika ir surinkta pirmoje dalyje informacija.

Y. Lizhi pristatytame *laiko* slaptame informacijos kanalo sudarymo metode naudojamos RTCP protokolo *run lenght code* ir *multi zero code* antraštės. Veikimo principas – jei siunčiamų slaptų duomenų bitas yra toks pat, kaip prieš tai išsiųstas bitas, tuomet siunčiamas tik RTP duomenų paketas. Priešingu atveju siunčiami RTCP ir RTP paketai.

SIP – protokolas, kuris palengvina ryšio formavimą, modifikavimą ir vykdymą tarp dviejų ar daugiau sesijos dalyvių. Vartotojų adresai (pvz., el. pašto adresas) leidžia identifikuoti ir nustatyti dalyvių paskirties vietą.

SDP – skirtas aprašyti duomenų transliavimo inicijavimo parametrus. Pats protokolas duomenų neperduoda, tačiau naudojamas sesijos paskelbimo, sesijos pakvietimų išsiuntimo, duomenų formato suderinimo ir kitoms funkcijoms.

Lenkijos mokslininkų darbe aprašomi keli informacijos slėpimo atvejai panaudojat SIP protokolą [64]. Pagrindinės informacijos talpinimo pozicijos: *from* lauko *tag* parametre (paprastai šiame parametre esanti reikšmė yra atsitiktinai sugeneruota ir naudojama SIP dialogo unikaliam identifikatoriui suformuoti), *via* lauko *branch* parametras, *call-id* laukas (unikali reikšmė skirta identifikuoti skirtingiems skambučiams), pirmoji *cseq* laiko dalis (pradinės sekos numeris skirtas atskirti skirtingoms sesijoms) ir *max-forwards* laukas. Šiame darbe taip apžvelgiamos kelios SDP protokolo antraštės, tinkamos slaptų duomenų įterpimui. Keli pagrindiniai laukai: *v* (protokolo versija), *o* (sesijos įkūrėjas), *s* (sesijos pavadinimas), *t* (laikas, kiek sesija yra aktyvi), *k* (šifravimo raktas).

Taip pat pastebėta, kad antraščių išdėstymo tvarka SIP/SDP pranešimuose priklauso nuo programuotojo, dėl to išdėsčius antraštes slapta tvarka, tai gali būti panaudota slaptos informacijos perdavimui. Pavyzdžiui, *call-id* laukas po *cseq* interpretuojamas kaip vienetas, o atvirkštinis variantas – kaip nulis. Laukų pavadinimų didžiosios ir mažosios raidės suprantamos kaip viena ir ta pati raidė, dėl to nustačius lauko pavadinimą didžiosiomis raidėmis, tai interpretuojama kaip slaptos informacijos dvejetainis skaičius vienetas, o mažosiomis raidėmis – kaip dvejetainis skaičius nulis.

Pastarojo metodo autoriai taip pat apžvelgė galimybę slaptą informaciją įrašyti pasinaudojant SIP/SDP protokolų saugos mechanizmais, skirtais užtikrinti duomenų konfidencialumą. SDP duomenų paketo turinys, įrašytas į *sip invite* pranešimą, gali būti šifruojamas ir pasirašytas skaitmeniniu parašu panaudojant *s/mime*. Šifruoti duomenys atskiriami atsitiktinai sugeneruoto skaičiaus riba. Šis skaičius gali būti naudojamas perduoti steganografinius duomenims. Tas pats galioja ir parašo duomenų atskyrimo ribos skaičiui – jis tinkamas slaptų duomenų nešimui.

SSH – protokolas, skirtas kliento prisijungimui prie serverio aplinkos (angl. *shell*). Standartinis TCP protokolų šeimos prievadas, naudojamas SSH, yra 22. SSH protokolas naudojasi SSL šifravimo ir duomenų perdavimo tinklu sistema. SSH protokolas labai plačiai naudojamas specialiose programose, skirtose saugiai prisijungti prie nutolusio kompiuterio (serverio). SSH yra labai plačiai naudojamas, nes apsaugo slaptažodžius nuo jų perėmimo tinkle. SSH naudoja šifravimą, todėl konfidenciali informacija neperduodama atviru tekstu. SSH taip pat padeda išvengti duomenų klastojimo, nes šifravimo mechanizmas paremtas viešo rakto metodika. Taip pat SSH naudojamas išvengti IP paketų nukreipimo, DNS klastojimo ir manipuliavimo duomenimis [65].

N. B. Lucenos aprašytame metode siūloma *mac* lauko reikšmę naudoti slaptų duomenų perdavimui [66]. Pastarojo lauko reikšmė gali siekti iki 160 bpp. *Mac* lauko reikšmės atsitiktinumui simuliuoti slapta informacija yra suspaudžiama ir užšifruojama. Esant galimybei perimti SSH duomenų srautą, galima įterpti papildomą šifruotą informaciją SSH duomenų paketo pradžioje. Duomenų paketo pradžioje esantis 4 baitų skaičius indukuoja apie šifruoto turinio buvimą arba nebuvimą. Slaptai informacijai perduoti taip pat gali būti naudojamas *random padding* laukas, kurio reikšmė gali siekti iki 255 baitų.

1.3. Analizės išvados

Apžvelgus steganografinius tinklo protokolus, galima pateikti šias išvadas:

1. Tinklu perduodami duomenys gali būti pažeisti tinkle susidarančio triukšmo;
2. UDP tinklo protokolas negarantuoja siunčiamos informacijos sėkmingo pristatymo;
3. UDP tinklo protokolas neužtikrina duomenų eiliškumo išlaikymo persiunčiant duomenis;
4. Perdavimo lygmens steganografiniai metodai, realizuoti UDP protokole, nenumato duomenų tėkmės kontrolės.
5. Aplikacinio lygmens steganografiniai metodai, transporto lygmenyje naudojantys UDP, taip pat nesprendžia sėkmingo informacijos pristatymo gavėjui problemos.

Remiantis šiais pastebėjimais, buvo suformuluotas darbo tikslas: **sukurti patobulintą steganografinį metodą, užtikrinantį prarastų duomenų atkūrimą bei persiunčiamų duomenų eiliškumo išlaikymą UDP protokole**.

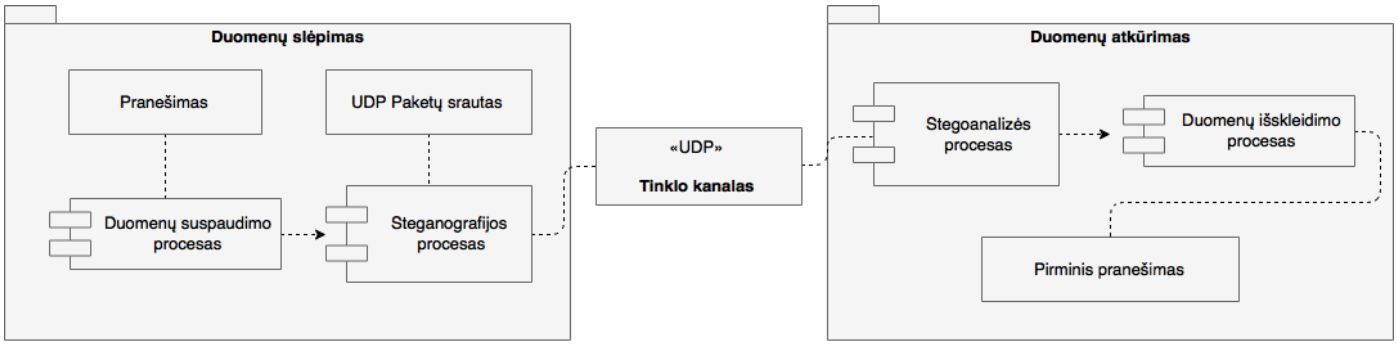
Tikslui pasiekti keliami šie uždaviniai:

- išanalizuoti esamus klaidų korekcijos metodus ir pasirinktus pritaikyti metodo kūrimo procese;
- pritaikyti metodą duomenų perdavimui UDP protokolu;
- atlikti sukurto sprendimo tyrimą ir rezultatų analizę.

2. UDP STEGANOGRAFINIS METODAS

Projektuojant steganografinį UDP metodą, buvo remiamasi analizės rezultatais, kurie parodė, kad steganografinių metodų, realizuotų UDP tinklo protokole, pagrindinis trūkumas yra informacijos perdavimo nepatikimumas. UDP siunčiamų duomenų pristatymas gavėjui nėra užtikrinimas jokiomis priemonėmis, persiuntimo metu duomenų paketai gali būti negrąžinamai prarasti. Pastarajai problemai spręsti bus naudojami skirtingi klaidų korekcijos algoritmai. Šios neigiamos UDP protokolo savybės eliminavimui yra skiriamas aukščiausias prioritetas kuriamame sprendime. Antras svarbus metodo aspektas yra gautų duomenų eiliškumo išlaikymas. IP tinkluose duomenis perduodant UDP protokolu, duomenų paketai į tą patį galinį tašką gali keliauti skirtingais tinklas. Dėl skirtingų vėlinimų (angl. *latency*) tarp tinklų, duomenų paketai gavėją pasiekia skirtinga tvarka nei buvo išsiųsti. Slaptų duomenų atkūrimas duomenų paketus gavus ne paeiliui dažnu atveju nėra įmanomas, tad dėl šios priežasties kuriamas sprendimas buvo papildytas duomenų eiliškumo atkūrimo funkcionalumu.

Visas darbo modelis sudarytas iš dviejų pagrindinių dalių: duomenų slėpimo dalyje pranešimas koduojamas ir įterpiamas į UDP paketų srautą steganografiniame procese, duomenų atkūrimo dalyje iš gauto UDP paketų srauto stegoanalizės procese išgaunamas pirminis pranešimas (22 pav.).



22 pav. Siūlomo metodo koncepcinis modelis

2.1. Klaidų korekcijos kodai

Skaitmeninius duomenis perduodant nepatikimu ryšio kanalu tikėtina, kad dalis duomenų gali būti sugadinti arba visiškai prarasti. Klaidų aptikimo ir klaidų korekcijos algoritmais galima aptikti ir ištaisyti klaidingus ar prarastus duomenis. Pagrindinė idėja prarastų duomenų atkūrimo algoritme yra papildomų duomenų pridėjimas prie perduodamų duomenų, kuriais remiantis gavėjas gali įsitikinti gautų duomenų vientisumu bei atkurti prarastus ar sugadintus duomenis. Skaitmeninių duomenų klaidų korekcijos kodai skirstomi į dvi pagrindines kategorijas: blokinius ir srautinius.

Blokiniai kodai duomenų seką padalina į k dydžio blokus. Duomenų blokas atvaizduojamas kaip dvejetainis k dydžio kortežas $u=(u_1,u_2, \ldots, u_k)$ ir yra vadinamas pranešimu. Kintamasis u skirtas atvaizduoti ne visą duomenų seką, o k bitų *pranešimą*. Galimų pranešimų skaičius išreiškiamas kaip 2^k. Kodavimo metu algoritmas koduoja kiekvieną pranešimą u atskirai į n dydžio kortežą $v=(v_1,v_2,\ldots,v_n)$, sudarytą iš diskrečių simbolių, vadinamų kodiniu žodžiu. Kintamuoju v atvaizduojamas n dydžio simbolių blokas. Remiantis taisykle, kad maksimalus galimų pranešimų kombinacijų skaičius lygus 2^k, egzistuoja 2^k skirtingų kodinių žodžių kombinacijų. Šis 2^k dydžio kodinių žodžių rinkinys, kur vieno bloko dydis nusakomas n, yra vadinamas (n,k) blokiniu kodu. Kodo

koeficientas *R* išreiškiamas *R=n-k*. Šis dydis nusako informacinių bitų skaičių, skirtą vienam duomenų simboliui [67].

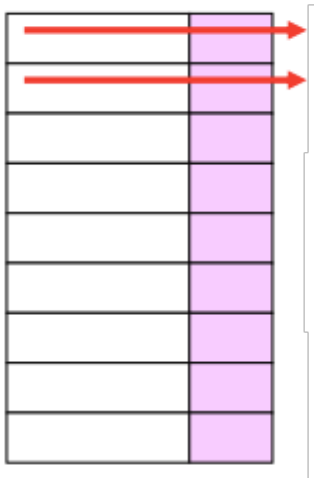
2.1.1. Klaidų pliūpsniai

Tinklu perduodamiems duomenims būdingi informacijos praradimai pliūpsniais. Klaidų pliūpsniais vadinami kelių paeiliui einančių simbolių praradimai duomenims keliaujant tinklu ar kitu

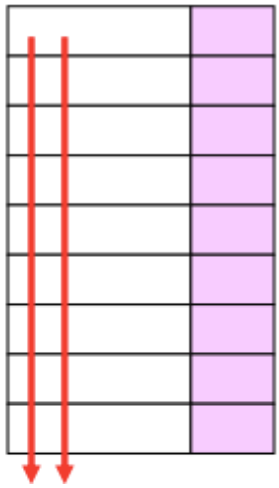
$$\mathbf{c}_1 = c_{11}c_{12}\dots c_{1n},\ \mathbf{c}_2 = c_{21}c_{22}\dots c_{2n},\ \dots\ ,\mathbf{c}_m = c_{m1}c_{m2}\dots c_{mn}$$

duomenų perdavimo kanalu. Vienas iš būdų spręsti klaidų pliūpsnių problemą, duomenis perduodant dvejetainės abėcėlės žodžiais, yra *simbolių išsibarstymo metodas*. Koduojant *m* srauto fragmentų, gauname *m* kodo žodžių [68] [189 psl.].

Į kanalą perduodama n žodžių po m simbolių. Gautus duomenis surašome eilute, tačiau nuskaitome stulpeliu (23, 24 pav.).



23 pav. Duomenų surašymas eilute



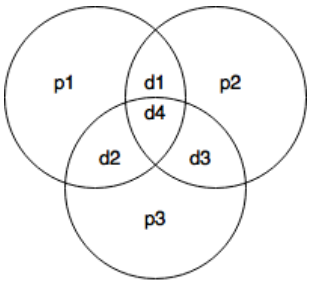
24 pav. Duomenų nuskaitymas stulpeliu

Kai gavėjas gaus *nm* simbolių, jis surašys juos į lentelę stulpelis po stulpelio ir, perskaitęs žodžius iš eilučių, turės *m* kodo *C* žodžių, kuriuose bus įvykę po vieną klaidą. Taigi klaidų pliūpsnio taisymas pavirsta pavienių klaidų taisymu keliuose žodžiuose [68] [190 psl.].

2.1.2. Hammingo klaidų korekcijos kodas

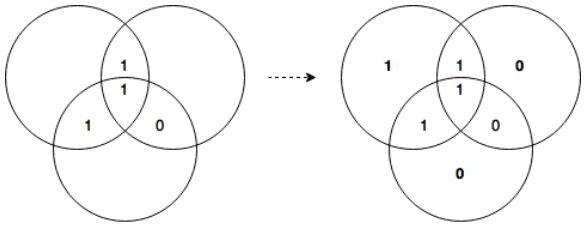
Hammingo kodų sistema buvo sukurta R. Hammingo 1950 m. Naudojant tradicinį Hammingo (7,4) kodą galima nustatyti pavienių klaidingų bitų pozicijas, taip pat dviejų paeiliui einančių klaidingų bitų buvimo faktą, tačiau įvardinti konkrečių jų pozicijų neįmanoma. Klaidingi bitai ištaisomi juos pakeičiant į atvirkštinį – 0 į 1 ir 1 į 0. Hammingo kodui pritaikius 1.4.1 skyrelyje „Klaidų pliūpsniai“ nagrinėtą simbolių išsibarstymo metodą, metodo efektyvumas gali būti pagerinamas ir net pasiekti

srautiniams kodams būdingą efektyvumo lygį. Vientisumo bitų apskaičiavimas geriausiai suprantamas atvaizduojant jį Venno diagrama (25 pav.)



25 pav. Grafinis keturių duomenų bitų ir jiems priklausančių vientisumo bitų vaizdavimas

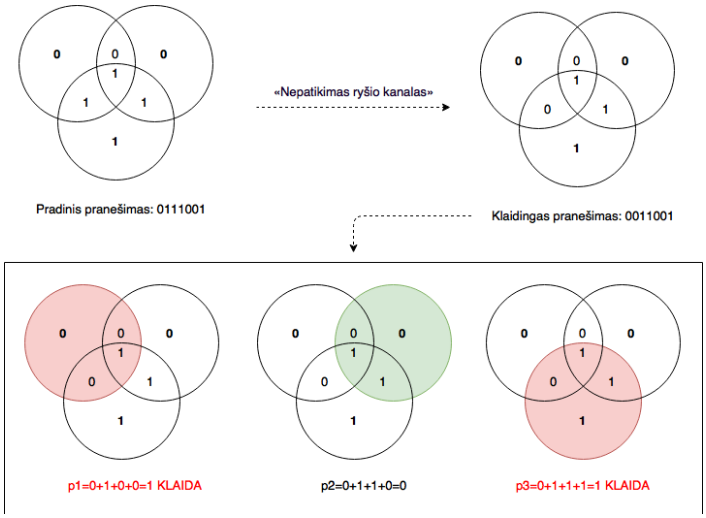
Kur p_n – apskaičiuoti vientisumo bitai, d_n – duomenų bitai. Kiekvieno apskritimo vienetų skaičius privalo būti lyginis. Tarkime, pranešimui užkoduoti $d_1d_2d_3d_4$=1101 pritaikomas Hammingo (7,4) algoritmas (26 pav.).



26 pav. Vientisumo bitų apskaičiavimas naudojant Venno diagramą

Gautas rezultatas p_1=*1*, p_2=*0*, p_3=*0*. Vientisumo bitai pridedami pradinių duomenų pabaigoje $d_1d_2d_3d_4p_1p_2p_3$=1101100. Klaidų patikra vykdoma perskaičiuojant gauto pranešimo vientisumo bitus (27 pav.).

Apskaičiavus informacinius bitus matyti, kad p_1 ir p_3 reikšmės neatitinka turimų duomenų. Remiantis turimais rezultatais galima daryti išvadą, kad klaidingas bitas yra p_1 ir p_3 skritulių susikirtimo vietoje. Tačiau susikirtimo vietoje yra du bitai – d_2 ir d_4. Kadangi p_2 reikšmė yra teisinga, tai galima daryti išvadą, kad d_4 bitas yra teisingas. Tai palieka vieną bitą, kuris yra neteisingas – d_2 [69].

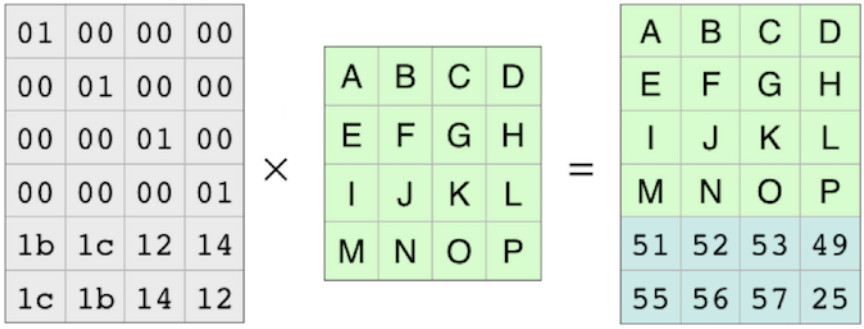


27 pav. Klaidingų bitų identifikavimas

2.1.3. Reed-Solomon klaidų korekcijos kodas

Reedo-Solomono klaidų korekcijos kodas sukurtas 1960 metais. Algoritmo autoriai – I. S. Reedas ir G. Solomonas. Reedo-Solomono (toliau – RS) blokinis klaidų korekcijos algoritmas dažnai naudojamas duomenis perduodant komunikaciniais kanalais bei duomenų saugojimo įrenginiuose. RAID sistemos, naudojamos *Linux* operacinėje sistemoje, naudoja RS algoritmą. Facebook *šaltoji duomenų saugykla* (angl. *cold storage*) duomenų saugojimui taip pat naudoja RS.

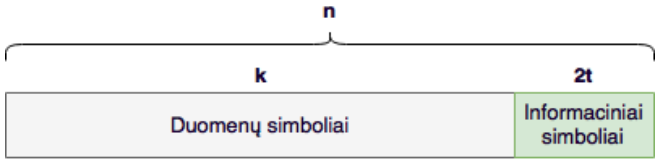
Šis metodas veikia įterpdamas papildomus informacinius simbolius duomenų, kurių reikšmė leidžia aptikti ir ištaisyti klaidingus duomenis, pabaigoje (28 pav.).



28 pav. Reedo-Solomono informacinių bitų apskaičiavimas. Kodavimo matrica sudauginama su duomenų matrica.

Kiekvieno RS įterpto simbolio reikšmė atitinka kelis *M* bitus. Pasikeitus bent vienam duomenų bitui, pasikeičia ir simbolio reikšmė, kuri nebeatitinka informacinio simbolio reikšmės. Šis klaidų korekcijos metodas priskiriamas prie srautinių klaidų korekcijos algoritmų. RS pasižymi galimybe koduoti neriboto ilgio pranešimus bei pridėti neribotą skaičių papildomų informacinių simbolių. RS algoritmas išreiškiamas formule *RS (n, n - k)*, kur n = bendras simbolių skaičius *kodiniame žodyje*; *k* = bendras vientisumo simbolių skaičius kodiniame žodyje; *n-k = 2t, t* = galimų ištaisyti simbolių skaičius, *s* = vieno simbolio dydis bitais.

Kodinis žodis (29 pav.) – tai informacijos ir informacinių simbolių junginys. Informaciniai simboliai – tai papildoma informacija, skirta klaidų korekcijai. RS algoritmas gali atkurti iki *(n-k)/2* klaidingų simbolių viename duomenų bloke [70] [258 psl.].



29 pav. Reedo-Solomono kodinio žodžio struktūra

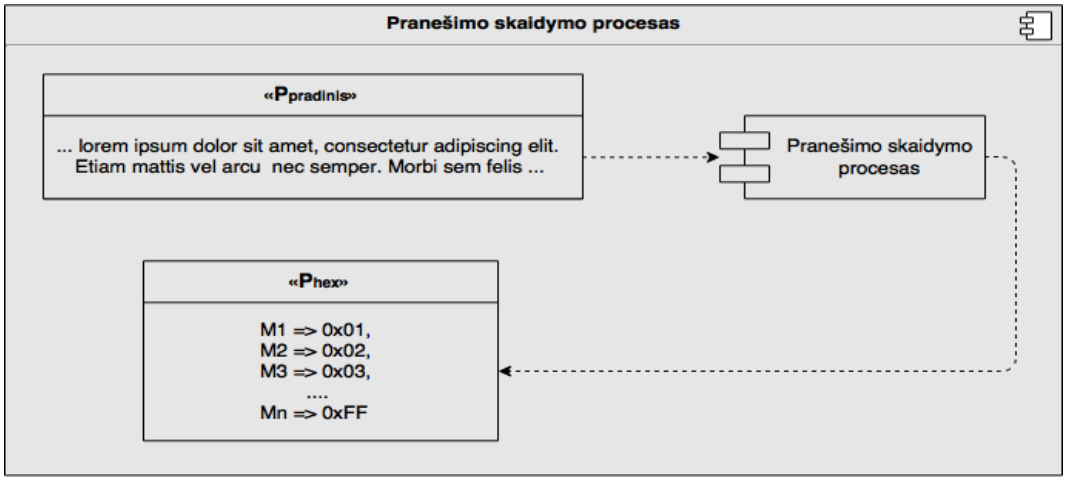
2.2. UDP steganografinis metodas

Kuriamas metodas paremtas dviem klaidų korekcijos algoritmais. Dėl skirtingo algoritmų veikimo bei skirtingos duomenų formavimo technikos kuriamo metodo procesų slėpimo ir atkūrimo schemos išskiriamos į dvi dalis – Hammingo ir Reedo-Solomono.

2.2.1. Reedo-Solomono klaidų korekcijos algoritmas

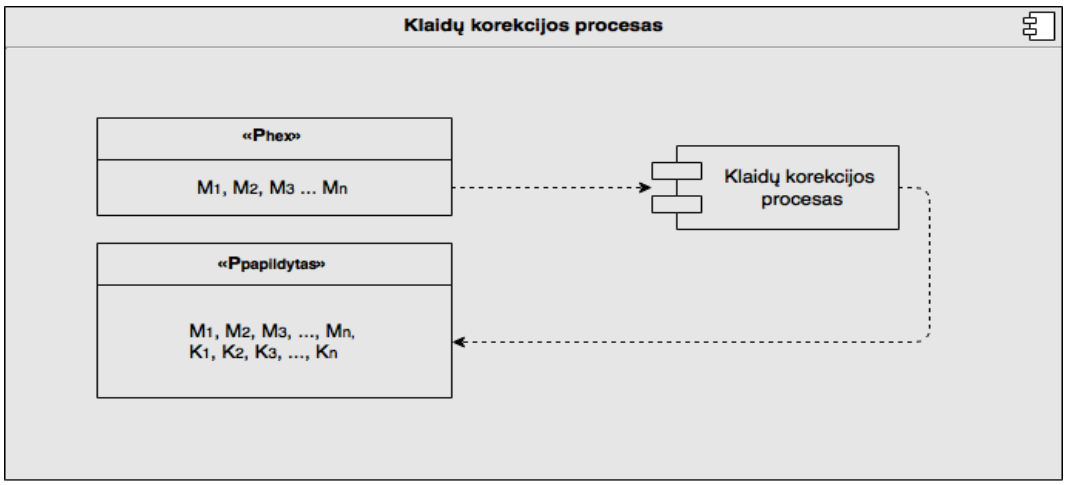
Duomenų slėpimo dalies procesas panaudojant Reedo-Solomono klaidų korekcijos algoritmą sudarytas iš žemesniame lygmenyje veikiančių procesų:

1. Pranešimo skaidymas į baitų masyvą. $M_1, M_2, \ldots, M_n$. Tekstiniai simboliai verčiami į skaitinius atitikmenis ASCII koduotėje (30 pav.). Vienas ASCII koduotės simbolis yra išreiškiamas šešioliktainiu skaitmenimi 8 bitų ribose.



30 pav. Pranešimo skaidymo proceso schema (Reedo-Solomono)

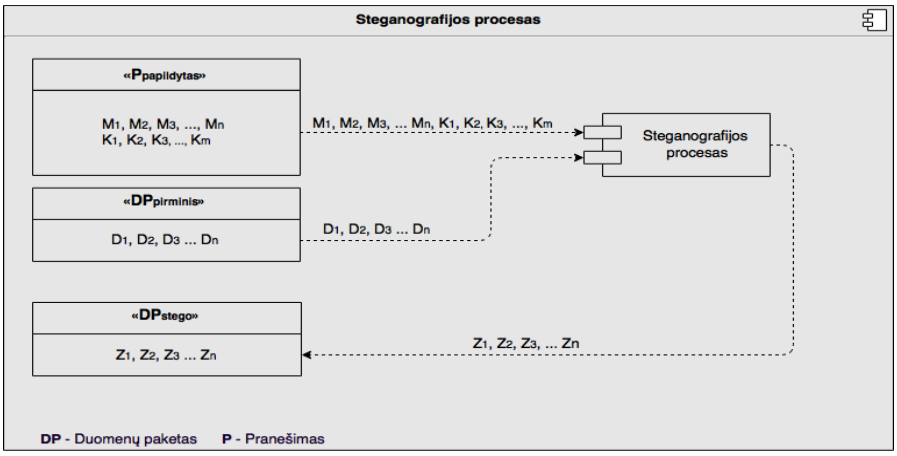
2. Klaidų korekcijos informacijos formavimas $M_1, M_2, M_3, \ldots M_n, K_1, K_2, K_3, \ldots K_n$. Prarastų duomenų atkūrimo metodo dalis paremta Reedo-Solomono algoritmu (31 pav.).



31 pav. Klaidų korekcijos proceso schema (Reed-Solomon)

Reedo-Solomono klaidų korekcijos algoritmas buvo pasirinktas dėl savybės, kuria pasižymi tinkle perduodami duomenys – tai srautiniai duomenų praradimai (n baitų) Klaidų korekcijos algoritmai paremti duomenų pertekliumi. Dėl šios priežasties didinant galimų atkurti bitų skaičių, taip pat didinamas siunčiamų duomenų kiekis. Siūlomame sprendime siekiama, kad vientisumo bitai neviršytų 100 % visų siunčiamų duomenų.

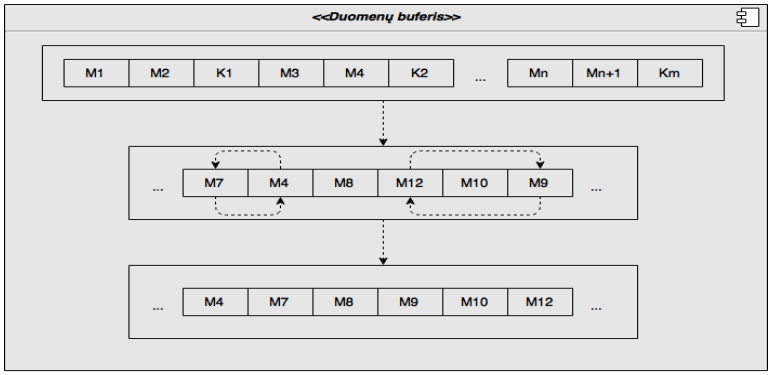
3. Duomenų ir numeracijos įterpimas į UDP paketų srautą. Duomenų eiliškumo atkūrimui užtikrinti naudojama siunčiamų duomenų paketų numeracija, kuri pridedama prie siunčiamų slaptų duomenų. Kuriamo metodo steganografijos dalis paremta K. Szczypiorskio (toliau – KS) darbe apžvelgta galimybe steganografinius duomenis įterpti UDP siuntėjo prievado lauke ir O. I. Abdullaziz (toliau – OIA) steganografiniu metodu slaptus duomenis išreiškiant kaip duomenų paketo dydį. Pastarieji metodai papildomi paketų numeravimo funkcionalumu (32 pav.).



32 pav. Steganografinio proceso schema (Reedo-Solomono)

4. Duomenų suspaudimo metodo dalis įgyvendinta pritaikius Lempelo–Zivo–Welcho duomenų suspaudimo algoritmą. Prarastų duomenų atkūrimo etape susidaręs siunčiamų duomenų pertekliškumas, pritaikius LZW suspaudimo algoritmą, gali būti sumažintas vidutiniškai 25–50 %.

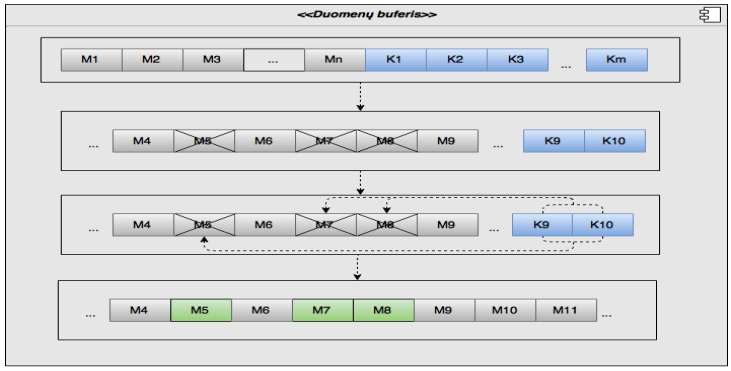
Duomenų atkūrimo dalies procesas panaudojant Reedo-Solomono klaidų korekcijos algoritmą sudarytas iš šių žemesniame lygmenyje veikiančių procesų:



33 pav. Duomenų eiliškumo atkūrimo proceso schema (Reedo-Solomono)

1. Duomenų eiliškumo atkūrimo proceso. Gauti duomenų paketai surašomi į duomenų masyvą (buferį) (33 pav.). Po paskutinio gauto duomenų paketo įvykdomas duomenų paketų rūšiavimas. Prarastų duomenų paketų pozicijose surašomi nuliai.

2. Prarastų duomenų paketų atkūrimo. Prarasti duomenų bitai atkuriami iš turimų informacinių duomenų bitų (34 pav.). Atvejais, kai visiškas duomenų atkūrimas neįmanomas, sekamas sugadintų bitų skaičius. Pasiekus vartotojo nustatytą maksimalią prarastų duomenų bitų ribą, pranešimas žymimas kaip neatkuriamas ir toliau netaisomas.



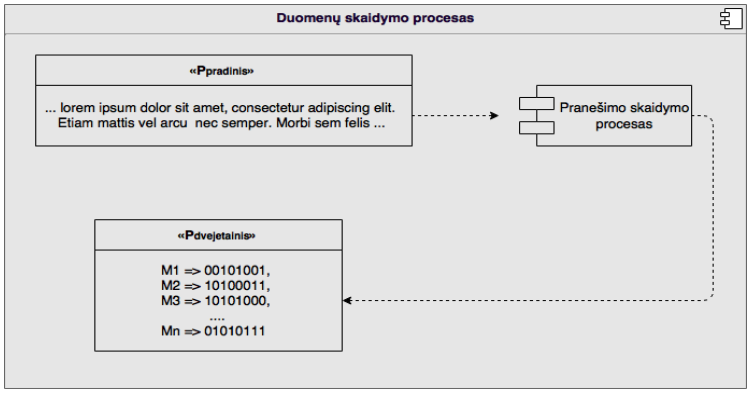
34 pav. Prarastų duomenų atkūrimo proceso schema (Reedo-Solomono)

3. Duomenų išskleidimo proceso. Po sėkmingo duomenų klaidų atkūrimo suspausti duomenys yra išskleidžiami naudojant *LZW* algoritmą, kuris buvo naudojamas duomenų suspaudimo etape.

2.2.2. Hammingo (8,4) klaidų korekcijos algoritmas

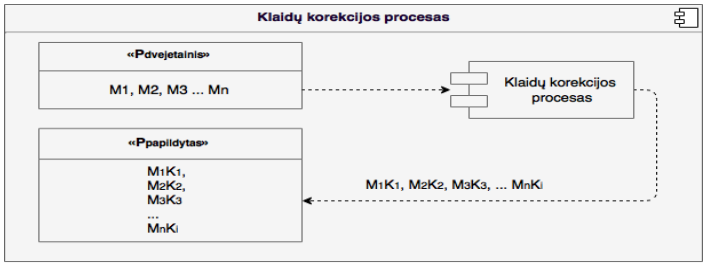
Duomenų slėpimo dalies procesas panaudojant Hammingo (8,4) klaidų korekcijos algoritmą sudarytas iš šių žemesniame lygmenyje veikiančių procesų:

1. Pranešimo skaidymo į 8 bitų blokus: $M_1, M_2, \ldots, M_n$. Tekstiniai simboliai verčiami į skaitinius atitikmenis ASCII koduotėje (35 pav.). Vienas ASCII koduotės simbolis yra išreiškiamas dvejetainiu skaitmenimi 8 bitų ribose.



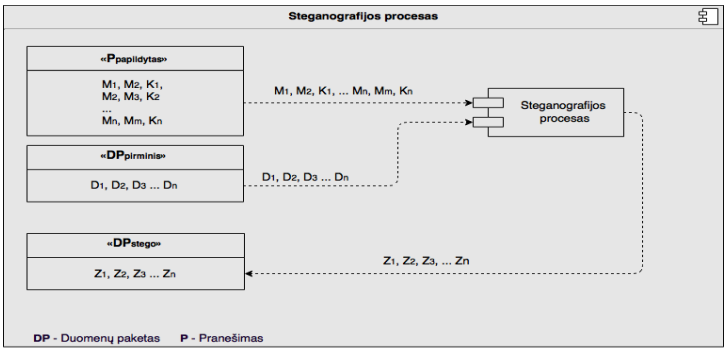
35 pav. Duomenų skaidymo proceso schema

2. Klaidų korekcijos informacijos formavimo $M_1, K_1, M_2, K_2, \ldots M_n, K_n$. Prarastų duomenų atkūrimo metodo dalis paremta Hammingo klaidų korekcijos algoritmais (36 pav.). Hammingo klaidų korekcijos algoritmas buvo pasirinktas dėl mažiau resursų naudojančio algoritmo įgyvendinimo. Klaidų korekcijos algoritmai paremti duomenų pertekliumi. Dėl šios priežasties didinant galimų atkurti bitų skaičių, taip pat didinamas siunčiamų duomenų kiekis. Siūlomu sprendimu siekiama, kad vientisumo bitai neviršytų 100 % visų siunčiamų duomenų.



36 pav. Klaidų korekcijos proceso schema

3. Duomenų ir numeracijos įterpimo į UDP paketų srautą. Duomenų eiliškumo atkūrimui užtikrinti naudojama siunčiamų duomenų paketų numeracija, kuri pridedama prie siunčiamų slaptų duomenų. Kuriamo metodo steganografijos dalis paremta *KS* darbe apžvelgta galimybe steganografinius duomenis įterpti UDP siuntėjo prievado lauke ir *OIA* steganografiniu metodu

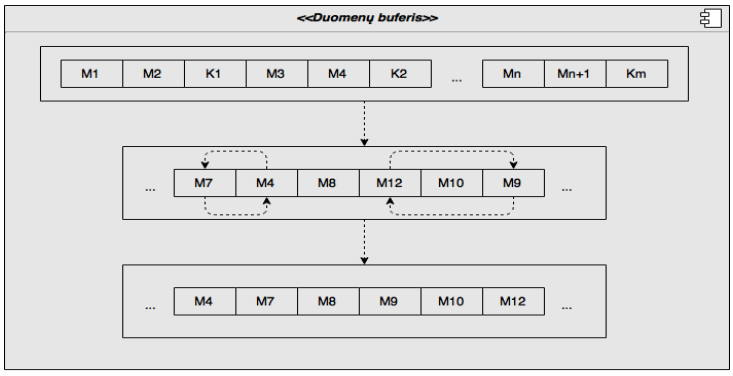


37 pav. Steganografijos proceso schema

slaptus duomenis išreiškiant kaip duomenų paketo dydį. Pastarieji metodai papildomi paketų numeravimo funkcionalumu (37 pav.).

4. Duomenų suspaudimo metodo dalis įgyvendinta pritaikius Lempelo–Zivo–Welcho duomenų suspaudimo algoritmą. Prarastų duomenų atkūrimo etape susidaręs siunčiamų duomenų perteklius gali būti sumažintas vidutiniškai 25–50 % pritaikius LZW suspaudimo algoritmą.

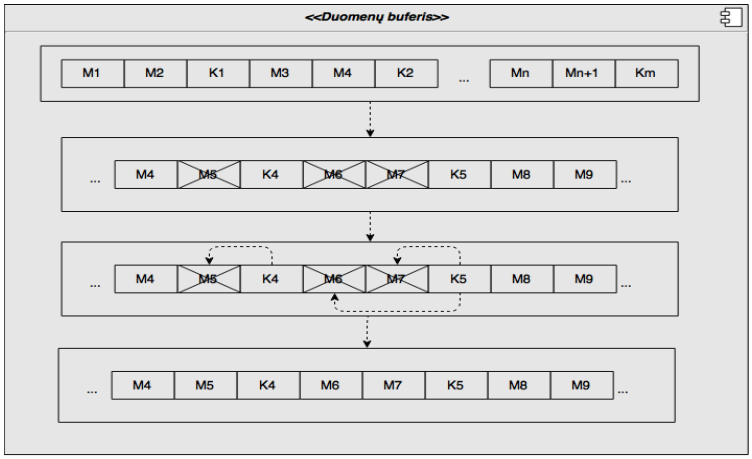
Duomenų atkūrimo dalies procesą sudaro šie žemesniame lygmenyje veikiantys procesai:



38 pav. Duomenų eiliškumo atkūrimo proceso schema

1. Duomenų eiliškumo atkūrimo procesas. Gauti duomenų paketai surašomi į duomenų masyvą (buferį) (38 pav.). Po paskutinio gauto duomenų paketo įvykdomas duomenų paketų rūšiavimas. Prarastų duomenų paketų pozicijose surašomi nuliai.

2. Prarastų duomenų paketų atkūrimas. Prarasti duomenų bitai atkuriami iš turimų informacinių duomenų bitų (39 pav.). Atvejais, kai visiškas duomenų atkūrimas neįmanomas, stebimas sugadintų bitų skaičius, pasiekus vartotojo nustatytą maksimalią prarastų duomenų bitų ribą, pranešimas žymimas kaip neatkuriamas ir toliau netaisomas.



39 pav. Prarastų duomenų atkūrimo proceso schema

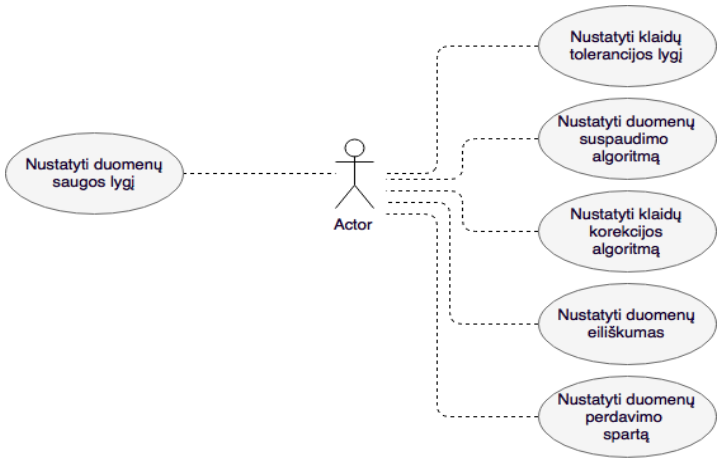
3. Duomenų išskleidimo procesas. Po sėkmingo duomenų klaidų atkūrimo suspausti duomenys yra išskleidžiami naudojant *LZW* algoritmą, kuris buvo naudojamas duomenų suspaudimo etape.

2.2.3. Patobulinto UDP steganografinio metodo panaudos atvejai

Pateikto pasiūlymo pagrindinės funkcijos yra pasirenkamos ir gali būti keičiamos galutinio vartotojo. Konfigūracijos metu vartotojas gali rinktis iš šių metodo nustatymų:

1. duomenų saugos lygis;
2. klaidų tolerancijos lygis;
3. duomenų suspaudimo algoritmas;
4. klaidų korekcijos kodas;
5. duomenų eiliškumo užtikrinimas;
6. duomenų perdavimo sparta.

Priklausomai nuo pasirinktų nustatymų yra didinama arba mažinama slaptų duomenų aptikimo rizika, persiunčiamų perteklinių duomenų kiekis, slaptų duomenų kiekio dalis persiunčiamuose duomenyse ir kt. Panaudos atvejai pateikiami 40 pav.



40 pav. Siūlomo metodo panaudos atvejai

1. Duomenų slaptumo lygis. Vartotojas gali rinktis vieną iš dviejų saugos lygių:
    a. Duomenų saugumo prioritetas – AUKŠTAS (žema duomenų perdavimo sparta).
    b. Duomenų saugumo prioritetas – ŽEMAS (aukšta duomenų perdavimo sparta).

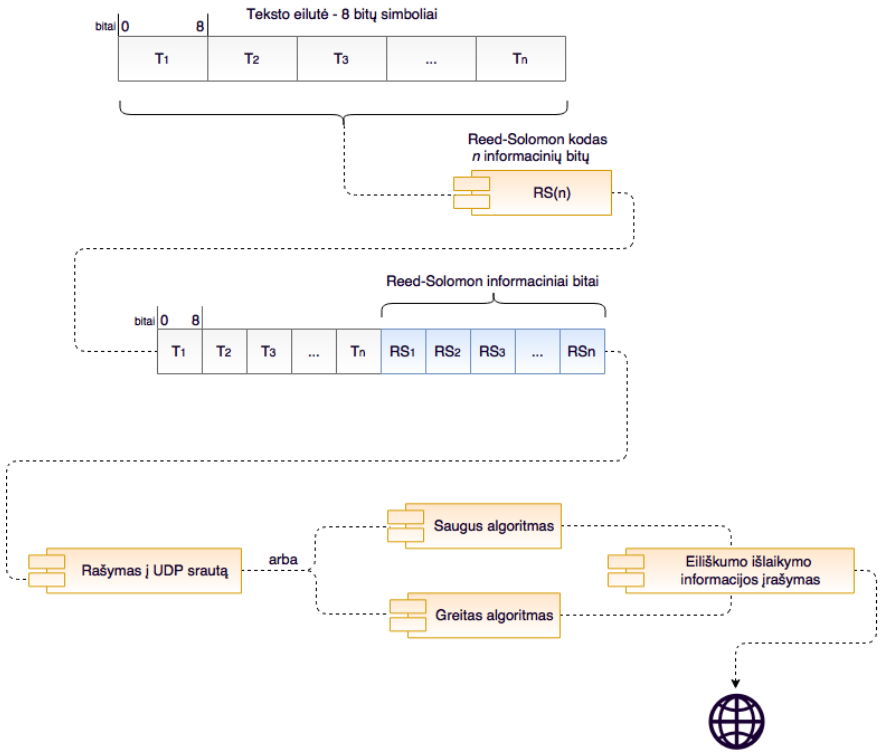
    Skirtingiems duomenų saugos lygiams realizuoti naudojami skirtingo aptikimo sudėtingumo steganografiniai metodai. Duomenų perdavimo sparta tarp skirtingų saugos lygių skiriasi iki 16-os kartų.

2. Klaidų tolerancijos lygis. **Siuntėjo pusėje** šiuo parametru nustatomas informacinių bitų kiekis atveju, kai pasirinktas Reedo-Solomono klaidų korekcijos algoritmas. Hammingo algoritmas naudoja fiksuotą informacinių bitų kiekį. **Gavėjo pusėje** šis parametras skirtas nustatyti maksimalų duomenų iškraipymo lygį, kai duomenų atkūrimo algoritmui nebeužtenka gautų duomenų atkurti pradiniams duomenims. Tai nutinka duomenų perdavimo metu praradus didelį kiekį duomenų. Pasiekus nustatytą lygmenį, nebandoma atkurti prarastų duomenų.
3. Duomenų suspaudimo algoritmas. Šis nustatymas leidžia vartotojui pasirinkti, naudoti ar nenaudoti duomenų suspaudimo.
4. Klaidų korekcijos kodas. Šiuo nustatymu nustatomas vienas iš dviejų klaidų korekcijų algoritmų, skirtų atkurti tranzito metu prarastus duomenų paketus. Vartotojas, siekdamas sumažinti perduodamų duomenų kiekį, taip pat gali pasirinkti nenaudoti klaidų korekcijos algoritmo. Toks pasirinkimas gali būti naudojamas, kai du galutinius taškus tinkle jungia sąlyginai nedidelis tarpinių stočių kiekis. Tai sumažina prarastų duomenų paketų tikimybę.
5. Duomenų eiliškumas. Šis nustatymas skirtas įjungti arba išjungti duomenų paketų numeracijos žymėjimą, skirtą išlaikyti duomenų eiliškumui atkūrimo metu. Kaip ir bet kokie papildomi duomenys, paketų numeracija didina persiunčiamų duomenų kiekį. Vartotojo pageidavimu ši numeracija gali būti panaikinta, kai duomenys perduodami reliatyviai nedideliu atstumu ir nedidele sparta.
6. Duomenų perdavimo sparta – tai nustatymas, leidžiantis didinti laiko intarpus tarp perduodamų duomenų paketų mili sekundžių (ms) intervalais. Atvejais, kai gavėjo techninė įranga negali susidoroti su dideliu įeinančių duomenų kiekiu, šis nustatymas leidžia riboti perdavimo greitį.

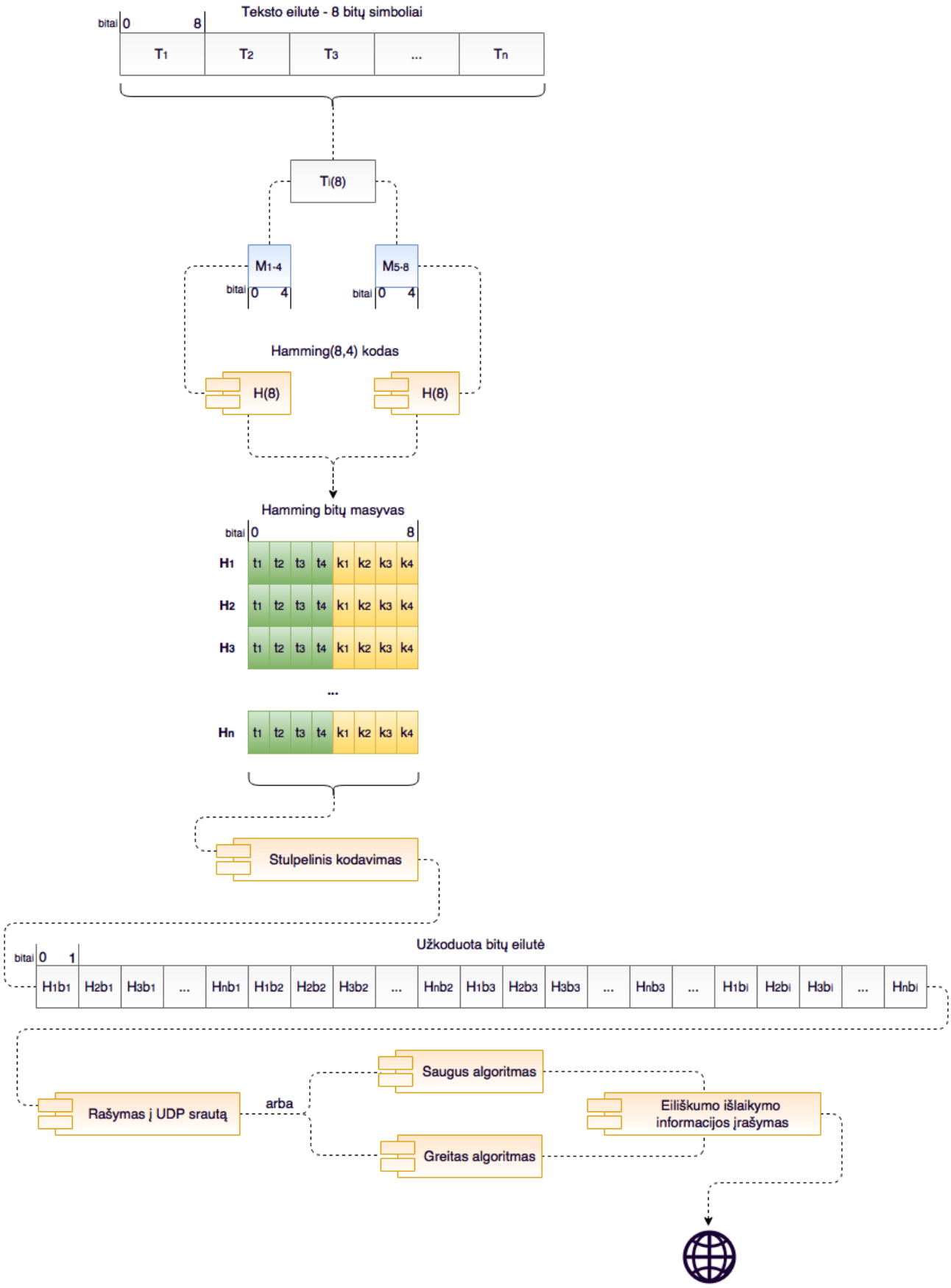
2.2.4. Duomenų patikimumo užtikrinimo metodas

Realizuojamas sprendimas (41, 42 pav.) sudarytas iš kelių paeiliui einančių dalių:

1. Perduodamų duomenų apdorojimas/kodavimas.
2. Įvykdomas duomenų suspaudimas.
3. Pritaikomas klaidų korekcijos kodas.
4. Pritaikomas duomenų blokavimo principas.
5. Pritaikomas steganografinis metodas.
6. Įvykdoma duomenų paketų numeracija.



41 pav. Duomenų patikimumo užtikrinimas naudojant Reedo-Solomono kodą



42 pav. Duomenų patikimumo užtikrinimas naudojant Hammingo (8,4) kodą

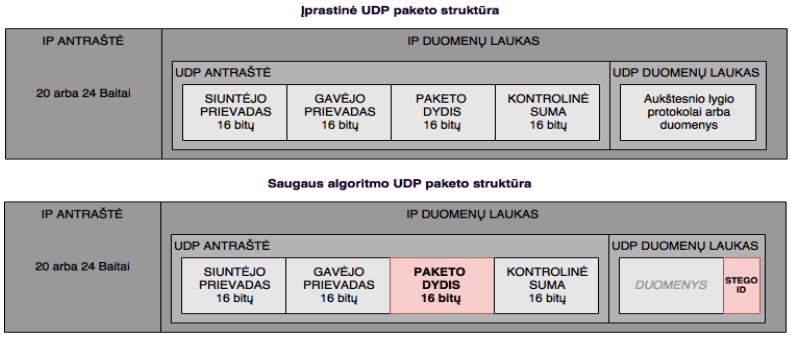
Metodas veikti pradeda nuo įvestos norimos perduoti tekstinės informacijos. Metodas pritaikytas darbui su standartine ASCII koduotės simbolių lentele (126 simboliai).

Kai vartotojas pasirinko naudoti duomenų suspaudimą įvestiems duomenims, pritaikomas pasirinktas duomenų suspaudimo algoritmas. Po suspaudimo Lempelo–Zivo–Welcho atveju prie duomenų yra pridedamas simbolių žodynas, kuriuo remiantis gavėjas įvykdys duomenų išskleidimą. Numatytoji nustatymo reikšmė: Lempel–Ziv–Welch.

Priklausomai nuo vartotojo pasirinkto klaidų korekcijos kodo, po suspaudimo duomenys yra padalinami į 8 (Reed-Solomono) arba 4 (Hammingo) bitų duomenų blokus. Hammingo atveju 4 stego bitai yra papildomi dar 4 vientisumo bitais pagal Hammingo (8,4) algoritmą. Nors Hammingo klaidų korekcijos kodas skirtas pavienėms klaidoms taisyti, tačiau ši problema išsprendžiama pritaikius stulpelinį duomenų kodavimo principą. Reedo-Solomono kodas iš prigimties yra sukurtas srautinių klaidų taisymui. Metode naudojamas statinis informacinių bitų skaičius siekiant, kad vientisumo bitai neviršytų projekte užsibrėžto tikslo – 100 % visų duomenų. Reedo-Solomono algoritmas gali aptikti *2t* klaidingų simbolių pranešime bei *t* klaidingų simbolių ištaisyti, kur *t* = informaciniai baitai, *s* = simbolių dydis bitais (8). Numatytoji nustatymo reikšmė: Reedo-Solomono.

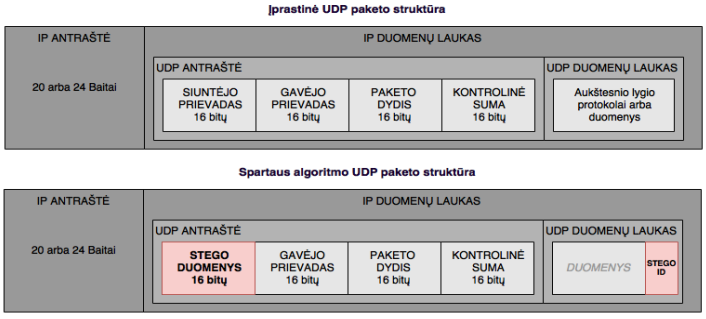
Pridėjus duomenų vientisumo bitus, pritaikomas stulpelinis duomenų blokavimo principas, jei naudojamas Hammingo klaidų korekcijos algoritmas. Reedo-Solomono atveju šis duomenų blokavimas duomenų patikimumo lygio nepadidina, todėl nėra taikomas. Numatytoji reikšmė: nenaudoti po duomenų sublokavimo duomenys yra paruošti talpinimui UDP pakete. Skirtingi steganografiniai metodai naudoja skirtingas duomenų perdavimo pozicijas ir technikas, todėl, priklausomai nuo pasirinkto saugos lygio, pritaikomas steaganografinis metodas:

a) Aukšto duomenų saugumo prioriteto atveju naudojamas *OIA* metodas (43 pav.).



43 pav. Saugaus stego metodo duomenų paketo struktūra

b) Žemo duomenų saugumo prioriteto atveju naudojamas *KS* metodas (44 pav.).



44 pav. Spartaus stego duomenų paketo struktūra

OIA algoritmui įgyvendinti naudojami natūraliai sugeneruoti tinklo VoIP duomenys. UDP duomenų paketo dydžio parametras yra interpretuojamas kaip slaptų duomenų vienetas, kai dydis yra lyginis skaičius, ir kaip slaptų duomenų nulis, kai UDP paketo dydis yra nelyginis skaičius. Siuntėjas savo siunčiamus duomenų bitus lygina su turimais natūraliais duomenimis ir atvejais, kai paketo ilgis neatitinka norimo persiųsti duomenų bito, natūralus duomenų paketas yra sutrumpinamas vienu bitu. Kuriamo metodo atveju natūralūs duomenys yra surinkti *Apple FaceTime* pokalbio metu tarp dviejų pašnekovų *WireShark* tinklo stebėjimo įranga – iš viso 10000 UDP duomenų paketų, saugomų faile.

Slaptų duomenų (*n* bitų) perdavimas įvykdomas per *n* duomenų paketų (1 lentelė).

1 lentelė. OIA metodo realizacija

Paketo Nr.	**Paketo dydis**	**Slaptas duomenų bitas**
Duomenų paketas 1	125	0
Duomenų paketas 2	44	1
Duomenų paketas 3	28	1
Duomenų paketas 4	29	0
Duomenų paketas 5	48	1
Duomenų paketas 6	99	0
Duomenų paketas 7	37	0
Duomenų paketas 8	87	0

KS metodas naudojamas, kai duomenų perdavimo sparta yra aukštesnio prioriteto, nei duomenų saugumas. Šis metodas slaptus duomenis koduoja UDP paketo siuntėjo prievado lauke (16 bitų). Gavėjas savo ruožtu gavėjo prievado lauko reikšmę padalina į dvi 8 bitų reikšmes, kurias vėliau atkoduoja į ASCII koduotės simbolius. Slaptų duomenų (*n* bitų) perdavimas įvykdomas per *n/16* duomenų paketų (2 lentelė).

2 lentelė. KS metodo realizacija

Paketo Nr.	**Siuntėjo prievado reikšmė**	**Slaptų duomenų reikšmė**
Duomenų paketas 1	01101011 01110100	KT
Duomenų paketas 2	01110101 00100000	U[*TARPAS*]
Duomenų paketas 3	01110101 01101110	UN
Duomenų paketas 4	01101001 01110110	IV
Duomenų paketas 5	01100101 01110010	ER
Duomenų paketas 6	01110011 01101001	SI
Duomenų paketas 7	01110100 01100101	TE
Duomenų paketas 8	01110100 01100001	TA
Duomenų paketas 9	01110011 00000000	S

Abiem atvejais duomenų paketų antraščių modifikavimui įgyvendinti reikalingas žemo lygio tinklo lizdas (angl. *raw socket*), kuris leidžia priimti/siųsti duomenų paketus tokius, kokie yra – be jokio transporto lygio protokolų duomenų formatavimo.

Duomenų paketų numeracijos pozicija UDP pakete priklauso nuo pasirikto steganografinio metodo. Pirmuoju atveju numeracijos duomenys siekiant užtikrinti siunčiamų duomenų neaptinkamumą gali būti išjungti. Kitu atveju, kai tinklo paketų eiliškumo praradimas yra aukšto lygio, numeracijos duomenys yra įrašomi į UDP paketo duomenų lauką.

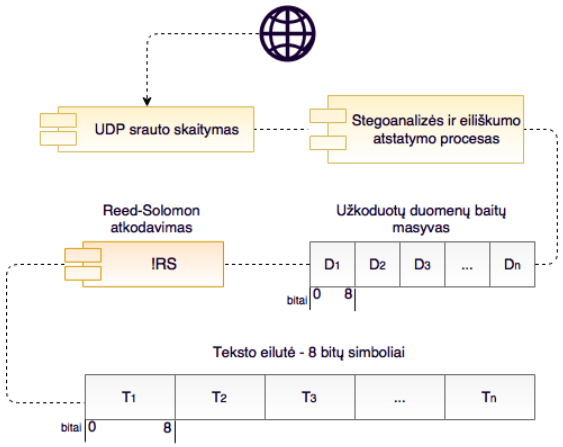
Vėliau nustatomas gavėjo IP adresas bei prievadas, kuriuo „klausosi“ gavėjas. Priklausomai nuo to, ar buvo pasirinktas vėlinimo laikas tarp duomenų paketų, duomenys yra siunčiami nustatytu laiko intarpu arba be laiko apribojimų. Duomenų įterpimo/siuntimo veiksmas kartojamas, kol pasiekiama įvestų slaptų duomenų masyvo pabaiga. Siunčiamų duomenų pabaiga gavėjui nurodoma išsiunčiant nulinį baitą (angl. *null byte*).

2.2.5. Duomenų atkūrimo metodas

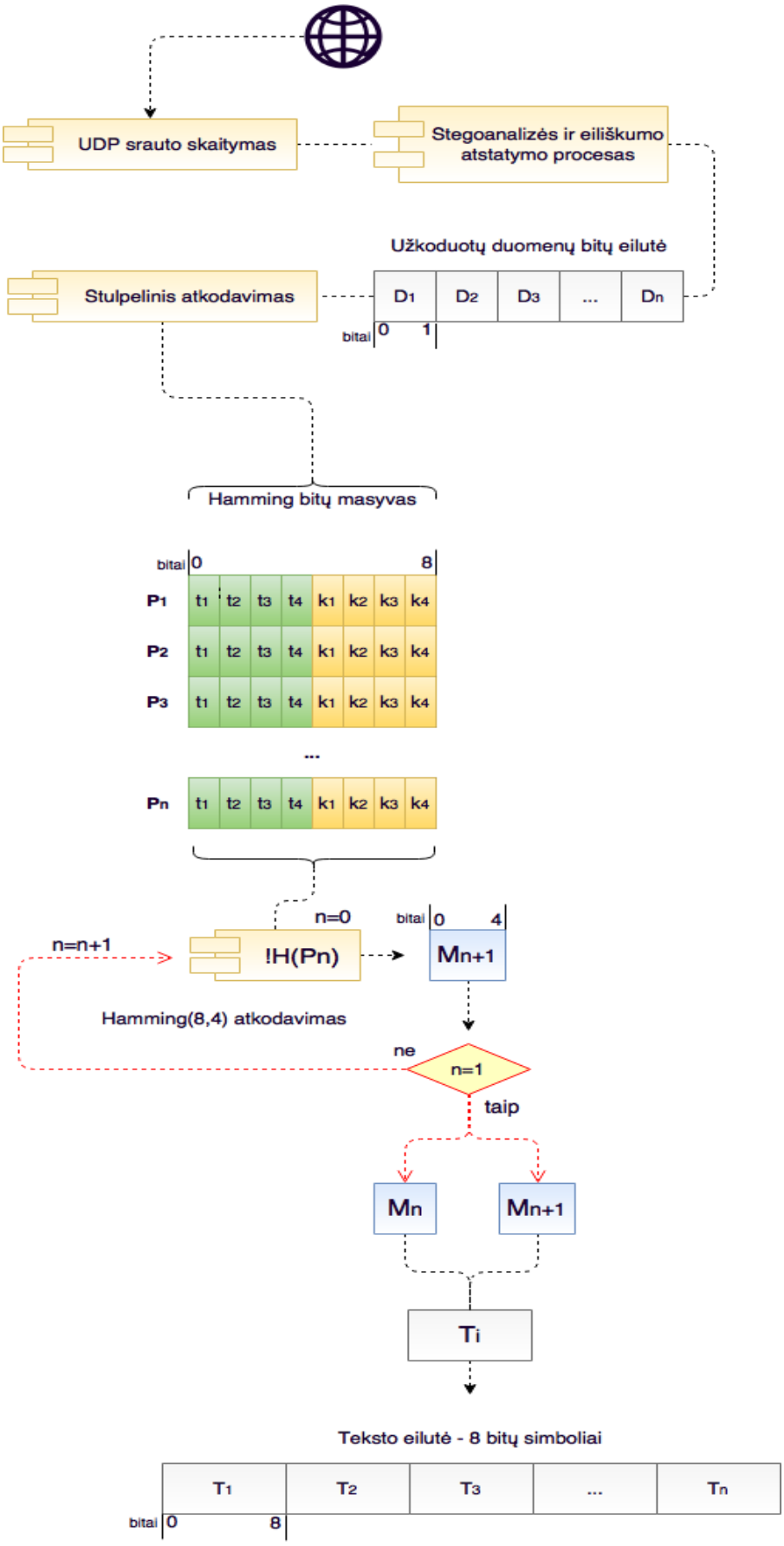
Duomenų atkūrimo metodo (45, 46 pav.) veikimo scenarijus:

1. Sukuriamas žemo lygmens tinklo lizdas (angl. *raw socket*).
2. Sukuriamas slaptų duomenų masyvas.
3. Pagal siuntėjo naudojamą steganografinį metodą nustatoma slaptų duomenų pozicija.
4. Slaptų duomenų masyvas pildomas gautais duomenimis, kol gaunamas nulinis baitas (angl. *null byte*) arba kai nuo paskutiniojo gauto duomenų paketo praeina 10 sekundžių.
5. Jei buvo naudojama duomenų paketų numeracija, iteruojama per gautų duomenų masyvą užpildant trūkstamų duomenų paketų pozicijas nuliniais bitais.
6. Uždaromas tinklo lizdas ir pradedama duomenų apdirbimo fazė.
7. Slaptų duomenų masyvas paverčiamas bitų eilute.
8. Jei siuntėjas naudojo duomenų blokavimo algoritmą, pritaikomas duomenų atblokavimo algoritmas.
9. Turima bitų eilė padalinama į bitų masyvą po 8 bitus.
10. Atliekama duomenų vientisumo patikra, jei įmanoma, atliekami pataisymai, jei ne, atsižvelgiant į klaidų tolerancijos lygio nustatymą, duomenys paliekami nekoreguoti.
11. Iš gautų duomenų pašalinami informaciniai bitai.
12. Jeigu buvo pritaikytas duomenų suspaudimo algoritmas, duomenys išskleidžiami.

Tokių būdu atkoduojami gauti slapti duomenys.



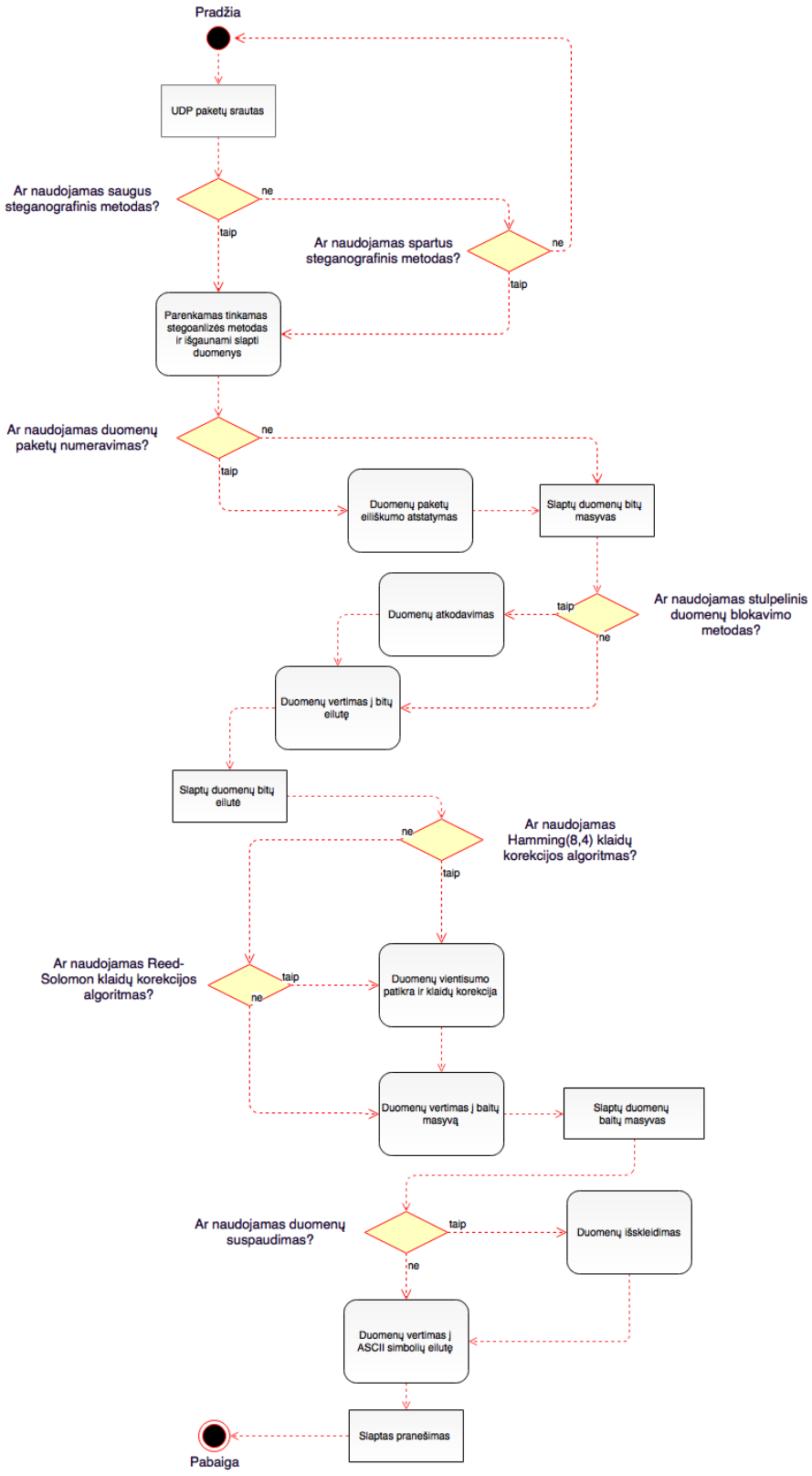
45 pav. Duomenų atkūrimo metodas naudojant Reedo-Solomono kodą



46 pav. Duomenų atkūrimo metodas naudojant Hammingo (8,4) kodą

2.3. UDP Steganografinio metodo veikimo simuliacija

Metodui (47 pav.) įgyvendinti buvo naudojami du virtualūs privatūs serveriai skirtinguose duomenų centruose (duomenų paketų praradimo simuliavimui).



47 pav. Siūlomo metodo veikimo schema (siuntėjo procesas)

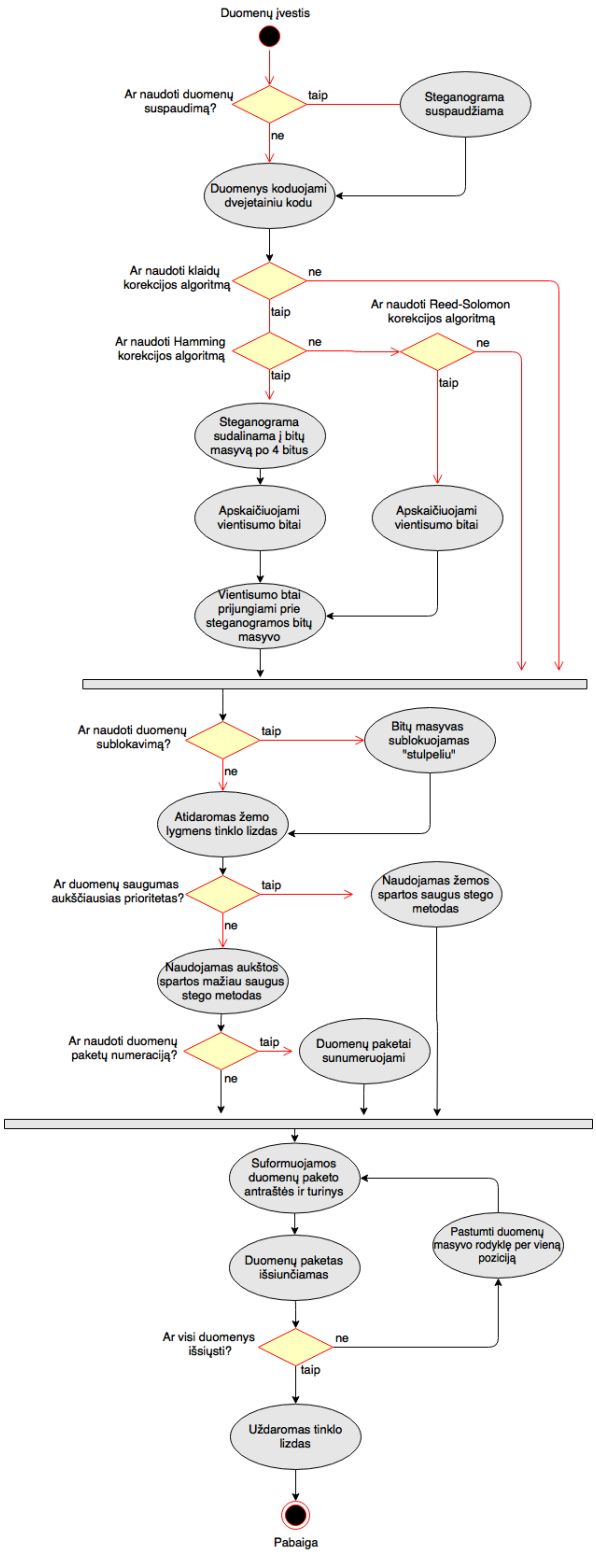
Serveriuose buvo įdiegta *Linux CentOS 7* operacinė sistema. Linux operacinė sistema buvo pasirinkta dėl galimybės programuoti naudojant žemo lygmens tinklo lizdą (angl. *raw socket*). Serveryje buvo įdiegtas SSH serveris, reikalingas nuotoliniam serverio konfigūravimui. Patogiam darbui su tinklo paketais naudojamas *Scapy* įrankis. Natūraliems tinklo paketams surinkti (naudojamiems saugiame stego metode) buvo naudojama *WireShark* programinė įranga. Reedo-Solomono algoritmo realizacijai naudojamas *reedsolo 0.3* programinis paketas. Programavimo kalba dėl aukštesnio programavimo efektyvumo, lyginant su žemesnio lygmens programavimo kalbomis, buvo pasirinkta *Python 2.7*.

Realizuotoje sistemoje (48 pav.) duomenų įvedimui naudojama terminalo eilutė prisijungus prie serverio per SSH kliento programą. Po duomenų įvedimo pasirenkami norimi konfigūraciniai nustatymai (gavėjo IP adresas, prievado Nr., duomenų suspaudimas, klaidų korekcijos algoritmas, duomenų blokavimas ir kt.). Konfigūracija baigiama iš tekstinio meniu pasirinkus punktą *siųsti duomenis*. Po sėkmingos konfigūracijos įvesti duomenis yra apdirbami taip, kad vėlesniuose etapuose jais būtų paranku manipuliuoti. Sėkmingai apdirbus duomenis, pradedamas kodavimo etapas.

48 pav. Siūlomo metodo klasių diagrama

Kodavimo metu įgyvendinamas duomenų vientisumo ir suspaudimo funkcionalumas. Duomenų įterpimo etape duomenys įrašomi į UDP paketų srautą pagal pasirinktą steganografinį metodą bei sunumeruojami. Paruošti duomenų paketai siunčiami tol, kol išsiunčiami visi įvesti slapti duomenys.

Gavėjo procese (49 pav.) konfigūraciniai sesijos nustatymai turi atitikti siuntėjo konfigūracinius nustatymus, priešingu atveju sėkmingo duomenų atkodavimo įvykdyti nepavyks. Slaptų duomenų paketams aptikti tinklo sraute naudojamas unikalus aukštas (>10000) lizdo prievadas,



49 pav. Siūlomo metodo veikimo schema (gavėjo procesas)

taip pat dar tikslesniam paketų aptikimui naudojamas filtravimas pagal siuntėjo IP adresą. Gautų duomenų atkodavimas pradedamas nuo slaptų duomenų išgavimo iš gautų duomenų paketų. Išgauti slapti duomenys atkoduojami pritaikant klaidų korekcijos algoritmą, eiliškumo atkūrimą bei išskleidimą. Slaptų duomenų atkūrimo etape duomenys suformuojami ASCII formatu ir atspausdinami gavėjo programos lange.

2.4. Projektavimo ir realizacijos išvados

Šiame skyriuje buvo atliktas patobulinto UDP steganografinio metodo projektavimas. Sudarinėjant metodą buvo vertinamas jo efektyvumas atkuriant prarastus duomenis bei išlaikant siunčiamų duomenų eiliškumą.

1. Atsižvelgiant į analizės dalies rezultatus perduodamų duomenų saugumui užtikrinti, buvo nuspręsta naudoti du skirtingus UDP steganografinius metodus, vieną – pasižymintį aukšta duomenų perdavimo sparta, kitą – užtikrinantį aukštesnį perduodamų duomenų saugumo lygį, tačiau pasižymintį žemesne duomenų perdavimo sparta.
2. Prarastų duomenų atkūrimo funkcionalumui įgyventini buvo nuspręsta naudoti du vartotojo laisvai pasirenkamus klaidų korekcijos algoritmus – patobulintą Hammingo (8,4) metodą, bei Reedo-Solomono algoritmą.
3. Hammingo (8,4) metodo patobulinimas įgyvendinamas panaudojant analizės dalyje aptartą stulpelinį bitų rikiavimo metodą.
4. Duomenų paketų eiliškumo užtikrinimas įgyvendinamas duomenų paketų numeracijos informaciją įterpiant UDP paketo duomenų lauko pabaigoje.
5. Dėl efektyvaus duomenų suspaudimo algoritmo veikimo duomenų kompresijai siūlomame projekte naudojamas Lempelo–Zivo–Welcho algoritmas.

3. UDP STEGANOGRAFINIO METODO TYRIMAS

3.1. Tyrimo metodikos

Tyrimas pradėtas nuo steganografinio UDP metodo atsparaus paketų praradimui sukūrimo ir realizacijos. Kuriamo metodo prototipas yra realizuotas Python programavimo kalba Linux operacinėje sistemoje.

Kuriamo sprendimo įvertinimui yra panaudoti šie pagrindiniai kriterijai:

- prarastų duomenų kiekis iki slaptų duomenų vientisumo praradimo (PDKVP);
- prarastų duomenų kiekis iki visiško slaptų duomenų praradimo (PDKVDP).

Tyrimas vykdomas dviem etapais:

1. Pirmame etape ištestuotas sukurto metodo veikimas pasinaudojant sukurta terminalo sąsaja bei panaudojant atskirų komponentų testavimą.
    a. Pirmojo bandymo metu gauti rezultatai (**kaip duomenų perdavimo patikimumui įtaką daro skirtingi duomenų kodavimo būdai**) palyginti pagal 4 scenarijus: Hammingo be suspaudimo, Hammingo panaudojant LZW suspaudimą, Reedo-Solomono be suspaudimo ir Reedo-Solomono su LZW suspaudimu. Visų scenarijų metu naudojamas blokavimas stulpeliu ir įprastai.
        i. scenarijus 1a – Hammingo algoritmas duomenis koduojant stulpeliu;
        ii. scenarijus 2b – Hammingo algoritmas duomenis koduojant eilute;
        iii. scenarijus 2a – Hammingo algoritmas su LZW algoritmu duomenis koduojant stulpeliu;
        iv. scenarijus 2b – Hammingo algoritmas su LZW algoritmu duomenis koduojant eilute;
        v. scenarijus 3a – Reedo-Solomono algoritmas duomenis koduojant stulpeliu;
        vi. scenarijus 3b – Reedo-Solomono algoritmas duomenis koduojant eilute;
        vii. scenarijus 4a – Reedo-Solomono algoritmas su LZW algoritmu duomenis koduojant stulpeliu;
        viii. scenarijus 4b – Reedo-Solomono algoritmas su LZW algoritmu duomenis koduojant eilute;
    b. Antrojo bandymo metu gauti rezultatai (**kokią įtaką duomenų perdavimo patikimumui daro prarastų duomenų paketų pozicija**) palyginti pagal 2 scenarijus: Hammingo panaudojant stulpelinį kodavimą, Reedo-Solomono duomenis koduojant įprastai. Abiejų scenarijų metu vertinami duomenų praradimai nuo pradžios paeiliui, duomenų praradimai nuo pabaigos paeiliui, duomenų praradimai atsitiktinėse pozicijose.
        i. scenarijus 1a – Hammingo algoritmas, duomenys prarandami nuo pranešimo pradžios paeiliui;
        ii. scenarijus 1b – Hammingo algoritmas, duomenys prarandami nuo pabaigos paeiliui;
        iii. scenarijus 1c – Hammingo algoritmas, duomenys prarandami atsitiktinėse pozicijose;
        iv. scenarijus 2a – Reedo-Solomono algoritmas, duomenys prarandami nuo pranešimo pradžios paeiliui;

v. scenarijus 2b – Reedo-Solomono algoritmas, duomenys prarandami nuo pranešimo pabaigos paeiliui;
vi. scenarijus 2c – Reedo-Solomono algoritmas, duomenys prarandami atsitiktinėse pranešimo pozicijose.

c. Trečiojo bandymo metu gauti rezultatai (**kokią įtaką duomenų perdavimo patikimumui daro informacinių bitų pertekliškumas**) palyginti pagal 3 scenarijus: Reedo-Solomono perduodamų duomenų dydis 250 B, 500 B, 1000 B. Visais trimis atvejais taikomi trys skirtingi vientisumo bitų pertekliškumo lygiai: 30 %, 50 %, 100 %.
   i. scenarijus 1 – Reedo-Solomono algoritmas tiriant 250 B dydžio pranešimus su trijų skirtingų dydžių informacinių bitų pertekliškumu;
   ii. scenarijus 2 – Reedo-Solomono algoritmas tiriant 500 B dydžio pranešimus su trijų skirtingų dydžių informacinių bitų pertekliškumu;
   iii. scenarijus 3 – Reedo-Solomono algoritmas tiriant 1000 B dydžio pranešimus su trijų skirtingų dydžių informacinių bitų pertekliškumu.

d. Ketvirtojo bandymo metu gauti rezultatai (**kokią įtaką duomenų perdavimo patikimumui daro duomenų paketų numeravimas**) palyginami pagal 2 scenarijus: Hammingo panaudojant duomenų blokavimą stulpeliu, Reedo-Solomono panaudojant įprastinį duomenų blokavimą. Abu scenarijai vertinami dviem atskirais atvejais: be numeravimo ir su numeravimu.
   i. scenarijus 1a – Hammingo algoritmas naudojant duomenų paketų numeravimą;
   ii. scenarijus 1b – Hammingo algoritmas nenaudojant duomenų paketų numeravimo;
   iii. scenarijus 2a – Reedo-Solomono algoritmas naudojant duomenų paketų numeravimą;
   iv. scenarijus 2b – Reedo-Solomono algoritmas nenaudojant duomenų paketų numeravimo;

2. Antrame etape lyginami rezultatai tarp realizuoto metodo ir nemodifikuoto UDP steganografinio metodo. Pagrindiniai palyginimo kriterijai – prarastų duomenų kiekis iki slaptų duomenų vientisumo praradimo ir prarastų duomenų kiekis iki slaptų duomenų visiško praradimo.

3.2. Sukurto atsparaus duomenų praradimams UDP steganografinio metodo tyrimas

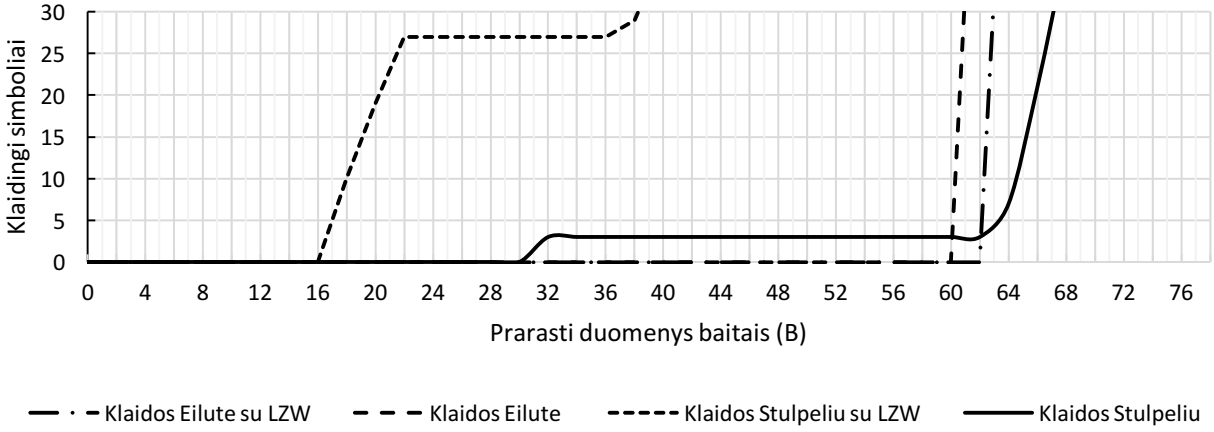
Pirmasis testas sukurtas norint išsiaiškinti, ar skirtingos duomenų kodavimo technikos turi įtakos slaptų duomenų atkūrimui po dalinio duomenų praradimo. Bandymo scenarijai: Hammingo su LZW, Reedo-Solomono, Reedo-Solomono su LZW. Vertinami atvejai: duomenų kodavimo stulpeliu ir duomenų kodavimas eilute (įprastai). Šiame bandyme visais keturiais atvejais naudojama ta pati tekstinė slapta žinute ASCII koduotės formatu. Naudojamas Lempelo–Zivo–Welcho duomenų suspaudimo algoritmas. Pirminiai testo parametrai: vientisumo bitų kiekis <= 50 %, perduodamos žinutės ilgis – 500 B, naudojant suspaudimą 429 B, duomenų praradimai vykdomi 2 B blokais. PDKVP rodiklio dydis nurodo metodo veikimo efektyvumą. Šiuo atveju didesnė reikšmė – didesnis prarastų duomenų kiekis ir efektyvesnis metodo veikimas. PDKVDP rodiklio reikšmė nurodo prarastų duomenų kiekį, kai nė vieno simbolio iš slapto duomenų pranešimo nebebuvo įmanoma atkurti.

3 lentelė. Skirtingų kodavimo metodų bei duomenų suspaudimo įtaka duomenų atkūrimo efektyvumui

Klaidų korekcijos algoritmo pavadinimas	Informacinių bitų kiekis	Slaptų duomenų dydis	Duomenų blokavimas	PDKVP	PDKVDP
Scenarijus 1a (Hammingo)	<101%	500B	Stulpeliu	12,8 % (64B)	74,8 % (374B)
Scenarijus 1b (Hammingo)		500B	Eilute (įprastai)	<0,4 % (2B)	>99 % (500B)
Scenarijus 2a (Hammingo su LZW)		429B	Stulpeliu	9,2 % (46B)	25,6 % (128B)
Scenarijus 2b (Hammingo su LZW)		429B	Eilute (įprastai)	<0,4 % (2B)	9,2 % (46B)
Scenarijus 3a (Reedo-Solomono)		500B	Stulpeliu	6,4 % (32B)	34,8 % (174B)
Scenarijus 3b (Reedo-Solomono)		500B	Eilute (įprastai)	12,8 % (64B)	75,6 % (378B)
Scenarijus 4a (Reedo-Solomono su LZW)		429B	Stulpeliu	4,20 % (18B)	95 % (408B)
Scenarijus 4b (Reedo-Solomono su LZW)		429B	Eilute (įprastai)	14,9 % (64B)	74,6 % (320B)

Pirmojo bandymo rezultatai.

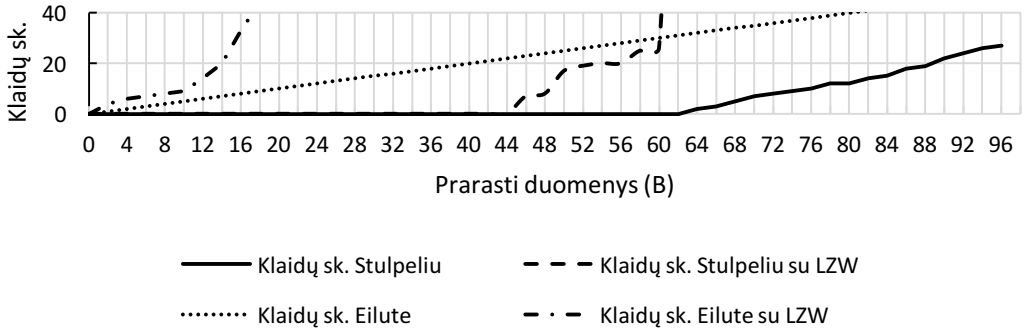
Nors Hammingo klaidų korekcijos algoritmas sukurtas pavienių klaidų korekcijai, tačiau pritaikius duomenų blokavimo techniką *stulpeliu*, bandymo metu pavyko sėkmingai atkurti slaptus duomenis gavėjo pusėje praradus 64 baitus (12,8 %) duomenų paeiliui. Tai 12,4 % geresnis rezultatas, lyginant su tradiciniu duomenų blokavimo metodu (50 pav.). Visiškas duomenų sunaikinimas naudojant šį blokavimo būdą įvyko praradus 74,8 % visų duomenų. Tai 24 % prastesnis rezultatas, lyginant su įprastine duomenų blokavimo technika, tačiau šiame darbe prioritetas teikiamas visiškam duomenų integralumo išlaikymui.



50 pav. Hammingo (8,4) skirtingų duomenų kodavimo metodų efektyvumas

Antrojo scenarijaus metu, kai buvo bandomas Hammingo algoritmo efektyvumas kartu su Lempelo–Zivo–Welcho algoritmu, pastebėta, kad abu vertinami kriterijai suprastėjo, lyginant su Hammingo algoritmo efektyvumu be LZW duomenų kompresijos.

Trečiojo scenarijaus metu Reedo-Solomono blokinis duomenų korekcijos algoritmas, atvirkščiai nei Hammingo algoritmas, sublokuotas *stulpeliu*, pademonstravimo prastesnį rezultatą, lyginant su tradiciniu blokavimo metodu (51 pav.). Tai atitinka 6,4 % ir 12,8 % prarastų duomenų iki slaptų duomenų integralumo praradimo. Visiško duomenų praradimo aspektu taip pat geriau pasirodė tradicinis duomenų blokavimo metodas.



51 pav. Reedo-Solomono skirtingų duomenų kodavimo metodų efektyvumas

Ketvirtojo scenarijaus metu, bandant Reedo-Solomono algoritmo efektyvumą su ir be LZW duomenų suspaudimo, pastebėta, kad efektyviausias algoritmo veikimas gaunamas duomenis blokuojant eilute ir naudojant LZW suspaudimą. Tai 2,1 % geresnis rezultatas, lyginant tą patį blokavimo metodą be suspaudimo ir 10,7 % geresnis rezultatas, lyginant su blokavimu stulpeliu su LZW suspaudimu.

Antrojo bandymo metu siekiama išsiaiškinti, kokią įtaką duomenų perdavimo patikimumui daro prarastų duomenų paketų pozicija visame duomenų bloke. Bandymo scenarijai: Hammingo naudojant stulpelinį duomenų kodavimą ir Reedo-Solomono algoritmas naudojant įprastinį kodavimo metodą. Vertinami atvejai: duomenų praradimai nuo pradžios paeiliui, duomenų praradimai nuo pabaigos paeiliui, duomenų praradimai atsitiktinėse pozicijose. Bandyme naudojama tekstinė slapta žinutė ASCII koduotės formatu. Pirminiai testo parametrai: vientisumo bitų kiekis <= 100 %, perduodamos žinutės ilgis – 500 B, duomenų praradimai vykdomi 2 B blokais. PDKVP rodiklio dydis nurodo metodo veikimo efektyvumą. Šiuo atveju didesnė reikšmė – didesnis prarastų duomenų kiekis ir efektyvesnis metodo veikimas. PDKVDP rodiklio reikšmė nurodo prarastų duomenų kiekį, kai nė vieno simbolio iš slapto duomenų pranešimo nebebuvo įmanoma atkurti.

4 lentelė. Klaidingų duomenų bitų pozicijos įtaka duomenų atkūrimo efektyvumui

Klaidų korekcijos algoritmo pavadinimas	Vientisumo bitų kiekis	Slaptų duomenų dydis	Pozicija	PDKVP	PDKVDP
Scenarijus 1a (Hamming)	**<51%**	**500B**	**Pradžia**	**64B**	**374B**
Scenarijus 1b (Hamming)			**Pabaiga**	**124B**	**372B**
Scenarijus 1c (Hamming)			**Atsitiktinė**	**10B**	**42B**
Scenarijus 2a Reed-Solomon			**Pradžia**	**64B**	**378B**
Scenarijus 2b Reed-Solomon			**Pabaiga**	**78B**	**442B**
Scenarijus 2c Reed-Solomon			**Atsitiktinė**	**128B**	**380B**

Antrojo bandymo rezultatai.

Duomenų praradimų nuo pradžios scenarijaus metu gauti rezultatai bandomais algoritmais neišsiskyrė (po 64 B).

Duomenų praradimų paeiliui nuo pabaigos scenarijaus metu Hammingo algoritmas parodė 59 % geresnį rezultatą lyginant su Reedo-Solomono, atitinkamai 124 B ir 78 B prarasti duomenų baitai iki pirmo neatkuriamo duomenų simbolio. Iki visiško duomenų praradimo Hammingo atveju buvo prarasti 74 % visų duomenų. Tai buvo 18 % prastesnis rezultatas.

Atsitiktinių duomenų praradimų atveju Reedo-Solomono algoritmas, lyginant su Hammingo, pademonstravo 11 kartų geresnį (1180 %) PDKVP. Visiškas duomenų praradimas įvyko Hammingo

atveju praradus 42 B visų duomenų. Tai buvo 9 kartais (804 %) prastesnis rezultatas, lyginant su Reedo-Solomono.

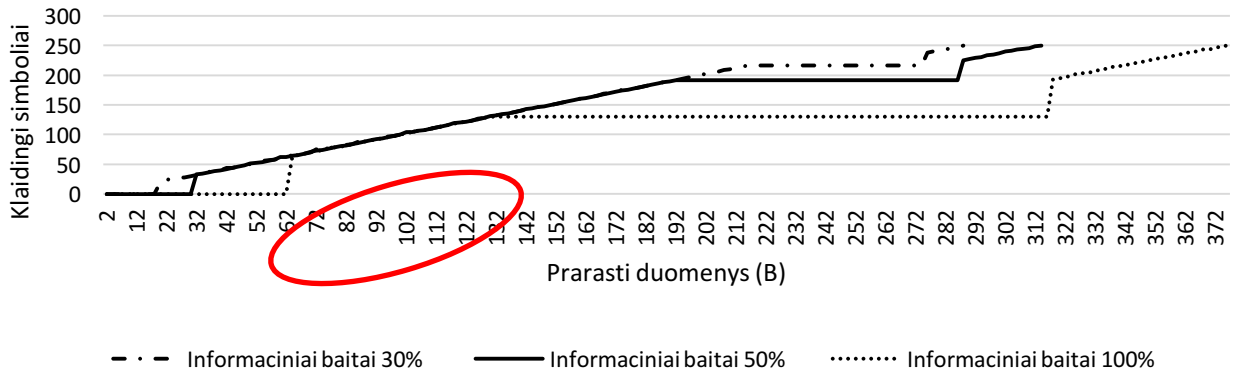
Trečiojo bandymo metu siekiama išsiaiškinti, kaip kokią įtaką duomenų perdavimo patikimumui daro informacinių bitų perteklius. Bandymo scenarijai: Reedo-Solomono perduodamų duomenų dydis 250 B, 500 B, 1000 B. Visais trimis atvejais taikomi trys skirtingi informacinių bitų pertekliškumo lygiai: 30 %, 50 %, 100 %.

5 lentelė. Informacinių bitų kiekio įtaka duomenų atkūrimo efektyvumui

Algoritmo pavadinimas ir scenarijaus Nr.	Duomenų dydis	Informacinių bitų pertekliškumas	PDKVP (B)	PDKV DP (B)	PDKVP (%)	PDKVDP (%)
Reedo-Solomono scenarijus 1	250 B	30% (325 B)	18	288	7.2 (5.5)	115.2 (88.6)
		50% (375 B)	30	314	12 (8.0)	125.6 (83.7)
		100% (500 B)	62	376	24.8 (12.4)	150.4 (75.2)
Reedo-Solomono scenarijus 2	500 B	30% (650 B)	24	600	4.8 (3.7)	120 (92.3)
		50%(750 B)	42	668	8.4 (5.6)	133.6 (89.0)
		100%(1000 B)	62	876	12.4 (6.2)	175.2 (87.6)
Reedo-Solomono scenarijus 3	1000 B	34.8% (1348 B)	28	1304	2.8 (2.1)	130.4 (96.7)
		50% (1500 B)	42	1420	4.2 (2.8)	142.0 (94.7)
		100% (2000 B)	62	1876	6.2 (3.1)	187.6 (93.8)

Trečiojo bandymo rezultatai.

Visų trijų scenarijų metu pastebėta, kad informacinių bitų kiekio didinimas nėra proporcingas duomenų atkūrimo efektyvumo didėjimui. Bandant skirtingo dydžio pranešimus ir proporcingai didinant informacinių bitų skaičių pastebėta, kad, nepaisant didesnio pranešimo dydžio, su 100 % pertekliškumu pirmas neatkuriamas simbolis įvyksta po paeiliui prarastų 62 baitų duomenų.



52 pav. Informacinių bitų pertekliaus įtaka duomenų perdavimo patikimumui

Taip pat pastebėta Reedo-Solomono algoritmo savybė, kai, nepaisant skirtingo informacinių bitų pertekliškumo lygio, algoritmo efektyvumas 70 B atkarpoje buvo vienodas (52 pav.). Atlikus bandymus su trijų skirtingų dydžių pranešimais nustatyta, kad ši atkarpa išsilaiko toje pačioje pozicijoje nepaisant didėjančio duomenų kiekio. Remiantis šiais duomenimis daroma išvada, kad sąlyginai mažesnio dydžio pranešimams taikomas didesnis informacinių bitų kiekis neturi įtakos duomenų atkūrimo efektyvumui.

Ketvirtojo bandymo metu bandoma išsiaiškinti, kokią įtaką duomenų perdavimo patikimumui daro duomenų paketų numeravimas. Bandymas vykdomas dviem scenarijais: Hammingo (8,4) algoritmas panaudojant duomenų blokavimą stulpeliu bei Reedo-Solomono algoritmą – tradiciniu duomenų kodavimo metodu. Vertinami rezultatai: be numeravimo ir su numeravimu. Simuliacijos parametrai: duomenų išsimaišymo lygis 30 %, maišomų duomenų blokų dydis 2 B, visas pranešimo ilgis – 250 B.

6 lentelė. Duomenų paketų numeravimo įtaka duomenų atkūrimo efektyvumui

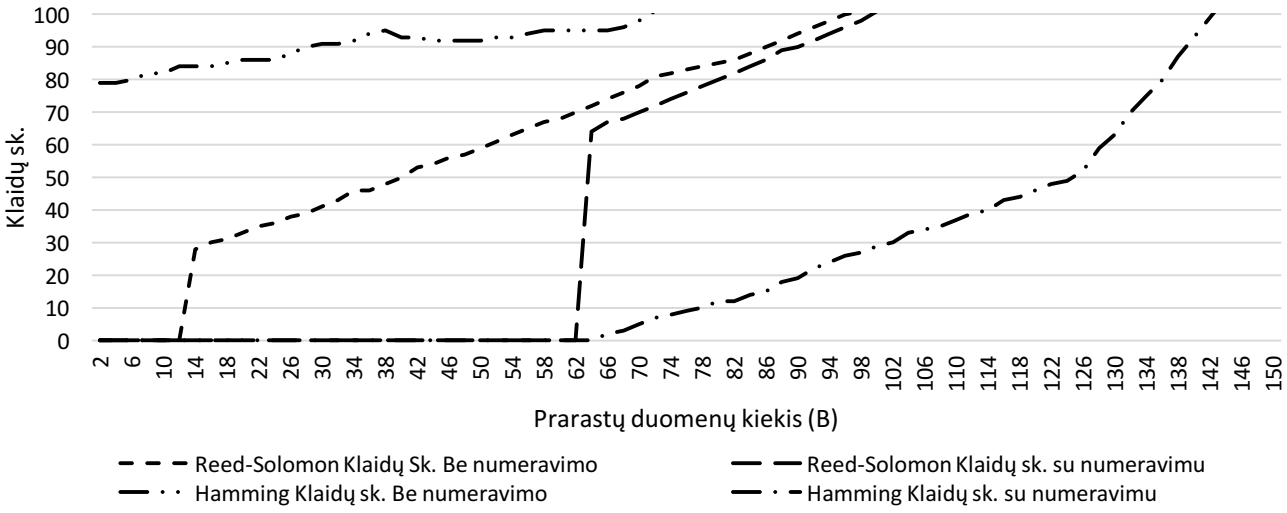
Scenarijus	Duomenų dydis	Informacinių bitų pertekliškumas	Duomenų išmaišymo lygis	PDKVP (B)	PDKVDP (B)
Scenarijus 1a (Hammingo)	250 B	100 %	30 %	66	368
Scenarijus 1b (Hammingo)				2	372
Scenarijus 2a (Reedo-Solomono)				64	378
Scenarijus 2b (Reedo-Solomono)				14	378

Ketvirtojo bandymo rezultatai.

Hammingo algoritmo bandymo metu nustatyta, kad nenaudojant duomenų numeravimo visiškai atkurti pranešimo nebuvo įmanoma nuo pirmųjų prarastų duomenų baitų, tuo tarpu naudojant duomenų numeravimą ir atstačius gautų duomenų eiliškumą, visiškas duomenų atkūrimas buvo galimas net ir praradus <66 B duomenų. Tai buvo geriausias rezultatas iš dviejų lyginamų algoritmų (53 pav.).

Reedo-Solomono algoritmo bandymo metu nustatyta, kad be duomenų numeravimo pirmas neatkuriamas duomenų baitas užfiksuotas po >14 B prarastų duomenų. Naudojant duomenų numeravimą tokiomis pat sąlygomis, pirmasis neatkuriamas duomenų baitas užfiksuotas po >64 B prarastų duomenų.

PDKVDP visais keturiais bandymų atvejais matavimo rezultatai skyrėsi nežymiai, todėl daroma išvada, kad šio rodiklio duomenų numeravimas įtakos nedaro.

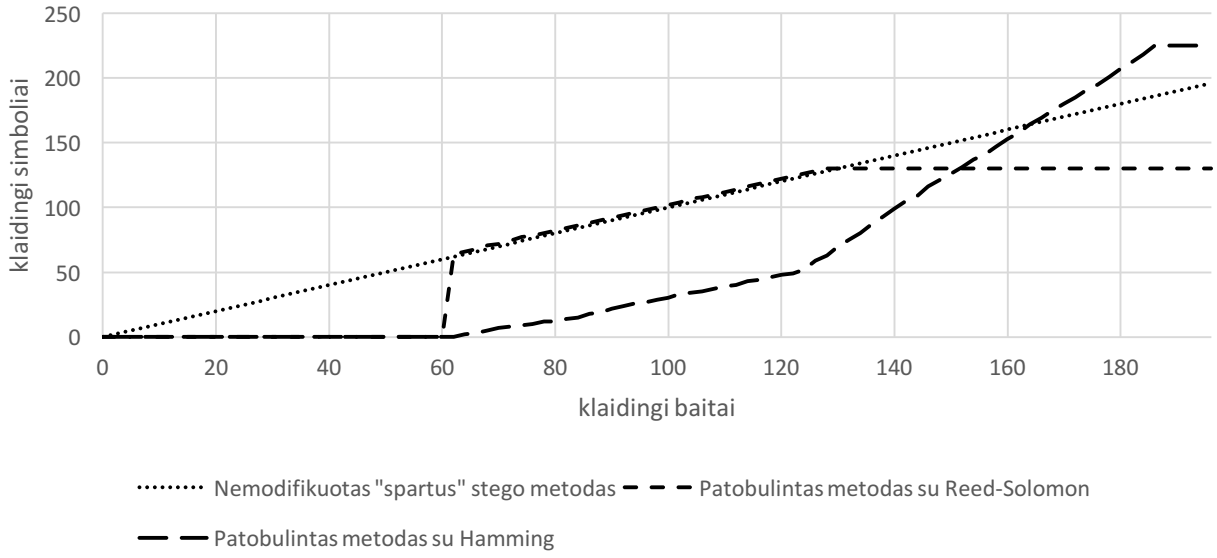


53 pav. Duomenų paketų numeravimo įtaka duomenų atkūrimui

3.3. Modifikuoto UDP steganografinio metodo palyginimas su KS ir OIA metodais

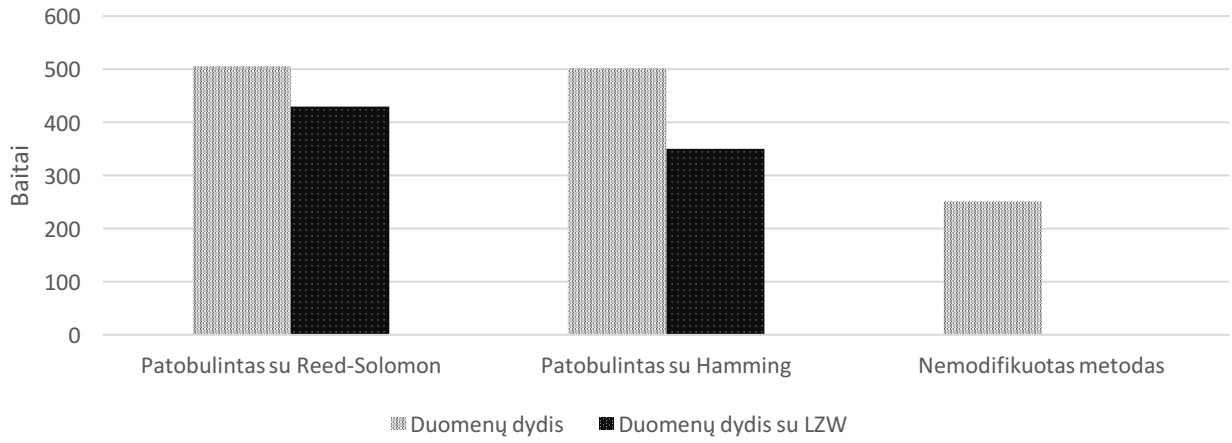
Antrame etape buvo lyginami modifikuoto KS ir OIA UDP steganografinų metodų rezultatai. Pagrindiniai palyginimo kriterijai – prarastų duomenų kiekis iki slaptų duomenų vientisumo praradimo ir prarastų duomenų kiekis iki slaptų duomenų visiško praradimo.

Palyginus modifikuotą algoritmą, *KS* ir *OIA* UDP steganografinius metodus (54, 55 pav.) matyti, kad klaidingų simbolių skaičius tiesiogiai priklausomas nuo prarastų baitų skaičiaus.



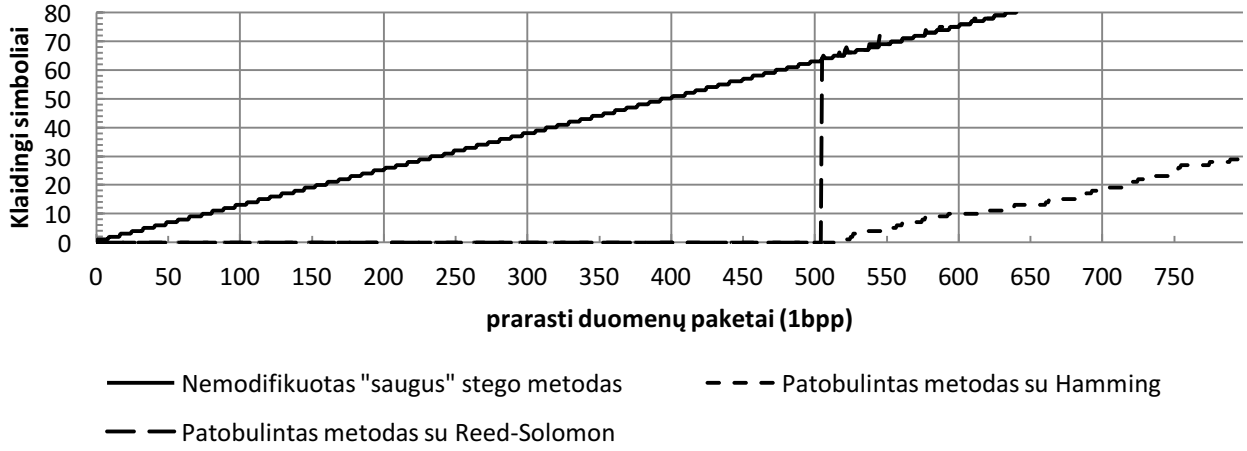
54 pav. Spartaus steganografinio metodo atsparumo trikdžiams palyginimas

Modifikuotu metodu pirmasis sugadintas simbolis, kurio atkurti nebebuvo įmanoma, užfiksuotas praradus apie 500 bitų visų duomenų (250 B). Tačiau reikia paminėti, kad, didėjant prarastų duomenų paketų skaičiui, duomenų atkūrimo efektyvumas mažėja. Kai kuriais atvejais metodas savo efektyvumu gali nusileisti nemodifikuotam metodui, kaip nutiko spartaus steganografinio metodo bandymo metu, kai, pasiekus ~60 % klaidingų simbolių ribą, patobulinto metodo (Hammingo) kreivė klaidingų simbolių atžvilgiu viršijo nemodifikuotą metodą (54 pav.)



55 pav. Saugaus steganografinio metodo atsparumo trikdžiams palyginimas

Palyginus patobulintą ir nemodifikuotą metodus matyti, kad, naudojant 100 % informacinių baitų perteklių, padvigubinamas perduodamų duomenų skaičius (56 pav.). Šis skaičius gali būti sumažintas panaudojant LZW suspaudimo algoritmą, tačiau ankstesni bandymai (1 lentelė) parodė, kad duomenų suspaudimo algoritmo naudojimas gali sumažinti klaidų korekcijos algoritmų veikimo efektyvumą. Siekiant užtikrinti aukščiausią metodo veikimo efektyvumą, būtina koreguoti sistemos parametrus (informacinių bitų pertekliškumą, suspaudimo algoritmą, klaidų korekcijos bei steganografinį algoritmus), pritaikant juos kiekvienu konkrečiu atveju prie esamų tinklo veikimo sąlygų.



56 pav. Metodų palyginimas pagal duomenų dydį siunčiant vienodo ilgio pranešimą

3.4. Tyrimo išvados

1. Palyginus rezultatus su standartiniais steganografiniais metodais, nustatyta, kad realizuotu metodu yra perduodama 30 –100 % daugiau duomenų nei įprastai.
2. Palyginus rezultatus su standartiniais steganografiniais metodais, nustatyta, kad nemodifikuotų metodų atveju klaidingų simbolių skaičius tiesiogiai priklauso nuo prarastų baitų skaičiaus.
3. Nustatyta, kad, naudojant Reedo-Solomono algoritmą, efektyviausias veikimo rezultatas pasiekiamas (vertinant PDKVP rodiklį) duomenis koduojant eilute (įprastai) ir duomenų suspaudimui naudojant LZW algoritmą.
4. Nustatyta, kad, naudojant Hammingo (8,4) algoritmą, efektyviausias veikimo rezultatas pasiekiamas (vertinant PDKVP rodiklį) duomenis koduojant stulpeliu, be duomenų suspaudimo.
5. Duomenų praradimų nuo pabaigos paeiliui atveju geriausią rezultatą PDKVP rodiklio atžvilgiu parodė Hammingo algoritmas, kai duomenys buvo koduojami stulpeliu.
6. Duomenų praradimų atsitiktinių pozicijų atveju geriausią rezultatą PDKVP rodiklio atžvilgiu parodė Reedo-Solomono algoritmas, kai duomenys buvo koduojami eilute.
7. Nustatyta, kad Hammingo algoritmas ištaisė vidutiniškai 66 baitus (~13 %) klaidingos informacijos iš 500 baitų dydžio pranešimo, naudojant 50 % informacinį pertekliškumo lygį.
8. Nustatyta, kad Reedo-Solomono algoritmas vidutiniškai ištaisė 90 baitų (~18 %) klaidingos informacijos iš 500 baitų dydžio pranešimo, naudojant 50 % informacinį pertekliškumo lygį.

4. IŠVADOS

Baigiamojo darbo tikslas – sukurti steganografinį metodą, užtikrinantį prarastų duomenų atkūrimą bei išlaikyti persiunčiamų duomenų eiliškumą UDP protokole. Panaudojant UDP protokolą, realizuotas steganografinis metodas bei ištirtas jo atsparumas tinklo trukdžiams. Perduodamų duomenų slaptumui užtikrinti naudojami *OIA* ir *KS* steganografiniai metodai. Duomenų užtikrinimui naudojami *Hammingo* ir *Reedo-Solomono* klaidų korekcijos algoritmai, duomenų suspaudimui – *Lempelo–Zivo–Welcho* algoritmas.

Projektuojant sistemą, metodo veikimui apibūdinti, algoritmo panaudos atvejams pavaizduoti bei komponentų diagramai nubraižyti panaudota UML modeliavimo kalba. Metodo realizacija atlikta Python programavimo kalba. Modifikuoto metodo palyginimui su *KS* ir *OIA* metodais nuspręsta panaudoti du vertinimo kriterijus: PDKVDP rodiklį (duomenų kiekis iki slaptų duomenų vientisumo praradimo) ir PDKVDP rodiklį (prarastų duomenų kiekis iki slaptų duomenų visiško praradimo).

1. Atlikus literatūros analizę nustatyta, kad UDP tinklo protokolas negarantuoja siunčiamos informacijos sėkmingo pristatymo. Dalis siunčiamos informacijos gali būti prarasta.
2. Išanalizavus tinklo steganografinius metodus nustatyta, kad perdavimo lygmens steganografiniai metodai, realizuoti UDP protokole, nenumato duomenų tėkmės kontrolės, t. y. siunčiama informacija iškraipyta tinkle, gavėjo pusėje, negali būti atkurta į pradinę būseną.
3. Palyginus rezultatus su standartiniais steganografiniais metodais nustatyta, kad modifikuoto metodo perteklinės informacijos kiekis sudaro 30–100 % persiunčiamų duomenų.
4. Gautus rezultatus palyginus su *KS* ir *OIA* steganografiniais metodais, nustatyta, kad nemodifikuotų metodų atveju klaidingų simbolių skaičius tiesiogiai priklauso nuo prarastų baitų skaičiaus.
5. Simuliuojant sukurto metodo veikimą skirtingų duomenų kodavimo technikų bandyme nustatyta, kad, naudojant Reedo-Solomono algoritmą, geriausias rezultatas vertinant PDKVP rodiklį gaunamas duomenis koduojant eilute (įprastai) ir duomenų suspaudimui naudojant LZW algoritmą (14,9 % PDKVP). Hammingo algoritmo atveju efektyviausias duomenų atkūrimas pasiekiamas duomenis koduojant stulpeliu, be duomenų suspaudimo (12,8 % PDKVP).
6. Tyrimo metu pastebėta, kad duomenų atkūrimo efektyvumas priklauso nuo to, kurioje pozicijoje duomenys prarandami. Duomenų praradimą fiksuojant paeiliui nuo perduodamų duomenų pradžios, PDKVP rodiklio geriausias rezultatas pastebėtas naudojant Hammingo algoritmą, kai duomenys koduojami stulpeliu. Atsitiktinių pozicijų atveju efektyviausiai metodas veikia naudojant Reedo-Solomono algoritmą.
7. Simuliacijos metu pastebėta, kad, didėjant prarastų duomenų paketų skaičiui, duomenų atkūrimo efektyvumas mažėja.

Siekiant užtikrinti aukščiausią metodo veikimo efektyvumo koeficientą, visi metodo parametrai turi būti suderinti su esamomis ir numatomomis tinklo sąlygomis, kuriose šis metodas yra taikomas.

5. LITERATŪRA

[1] B. A. Forouzan, TCP/IP Protocol Suite, New York: McGraw-Hill, 2010.

[2] L. Dostalek ir A. Kabelova, Understanding TCP/IP, Birmingham: Packt Publishing, 2006.

[3] B. W. Lampson, „A note on the confinement problem," *Communications of the ACM,* t. 16, nr. 10, pp. 613-615, 1973.

[4] W. Mazurczyk ir K. Szczypiorski, „Steganography of VoIP Streams," Warsaw University of Technology, Warsaw, 2008.

[5] C. H. Rowland, „Covert Channels in the TCP/IP Protocol Suite," *First Monday,* t. 2, nr. 5, 1997.

[6] T. G. Handel ir M. T. Sandford II, „Hidding Data in the OSI Network Model," Weapon Design Technology Group, Los Alamos, 2005.

[7] M. Mehic, J. Slachta ir M. Voznak, „Whispering through DDoS attack," *Perspectives in Science,* t. 7, pp. 95-100, 2016.

[8] W. Mazurczyk ir Z. Kotulski, „New security and control protocol for VoIP based on steganography and digital watermarking," Warsaw University of Technology, Warsaw, 2006.

[9] S. Chen, X. Wang ir S. Jajodia, „Tracking anonymous peer-to-peer VoIP calls on the internet," įtraukta *CCS '05 Proceedings of the 12th ACM conference on Computer and communications security*, New York, 2005.

[10] E. Jones, O. L. Moigne ir J.-M. Robert, „IP Traceback Solutions Based on Time to Live Cover Channel," IEEE, Ottawa, 2004.

[11] W. Mazurczyk, M. Karas ir K. Szczypiorski, „SkyDe: a Skype-based Steganographic Method," *nternational Journal of Computers, Communications & Control ,* t. 8, nr. 3, pp. 389-400, 2013.

[12] P. Kopiczko, W. Mazurczyk ir K. Szczypiorski, „StegTorrent: a Steganographic Method for the P2P File Sharing Service," įtraukta *Security and Privacy Workshops (SPW)*, San Francisco, 2013.

[13] A. Houmansadr, T. Riedl, N. Borisov ir A. Singer, „I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention," įtraukta *NDSS Symposium 2013*, Austin, 2013.

[14] P. Bialczak, W. Mazurczyk ir K. Szczypiorski, „Sending Hidden Data via Google Suggest," Warsaw University of Technology, Warsaw, 2011.

[15] H. Venter, M. Eloff, L. Labuschagne ir J. E. v. Solms, „ New Approaches for Security, Privacy and Trust in Complex Environments," įtraukta *IFIP TC-11 22nd International Information Security Conference (SEC 2007)*, Sandton, 2007.

[16] J. Lubacz, W. Mazurczyk ir K. Szczypiorski, „Principles and Overview of Network Steganography," *IEEE Communications Magazine,* t. 52, nr. 5, pp. 225 - 229, 2014.

[17] B. Jankowski, W. Mazurczyk ir K. Szczypiorski, „PadSteg: introducing inter-protocol steganography," *Telecommunication Systems,* t. 52, nr. 2, pp. 1101-1111, 2013.

[18] K. Szczypiorski ir W. Mazurczyk, „Hiding Data in OFDM Symbols of IEEE 802.11 Networks," įtraukta *International Conference on Multimedia Information Networking and Security*, Kathmandu, 2010.

[19] K. Szczypiorski, „HICCUPS: Hidden Communication System for Corrupted Networks," Warsaw University of Technology, Warsaw, 2003.

[20] D. Kundur ir K. Ahsan, „Practical Internet Steganography: Data Hiding in IP," Texas A&M University, Texas, 2003.

[21] T. G. Handel ir M. T. Sandford II, Hiding Data in the OSI Network Model, Los Alamos: Weapon Design Technology Group, 2005, p. 31.

[22] T. G. Handel ir M. T. Sandford II, Hidding Data in the OSI Network Model, Los Almos: Weapon Design Technology Group, 2005, p. 32.

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

[24] C. H. Rowland, „Covert Channels in the TCP/IP Protocol Suite: Method One: Manipulation of the IP Identification Field," *First Monday,* 1997.

[25] J. Touch, „Updated Specification of the IPv4 ID Field (RFC:6864)," Internet Engineering Task Force (IETF), 2013.

[26] E. Cauich, R. Gómez ir R. Watanabe, „Data Hiding in Identification and Offset IP fields," įtraukta *International Symposium and School on Advancex Distributed Systems* , Berlin, 2005.

[27] S. Zander, G. Armitage ir P. Branch, „Covert Channels in the IP Time To Live Field," Swinburne University of Technology, Melbourne, 2007.

[28] K. Ahsan, „Covert Channel Analysis and Data Hiding in TCP/IP," University of Toronto, Torronto, 2002.

[29] P. Peng, P. Ning ir D. S. Reeves, „On the Secrecy of Timing-Based Active Watermarking Trace-Back Techniques*," IEEE, Raleigh, 2006.

[30] C. Abad, „IP Checksum Covert Channels and Selected Hash Collision," University of California, California, 2001.

[31] M. Padlipsky, D. Snow ir P. Karger, „Limitations of end-to-end encryption is secure computer networks," United States Air Force, Massasuchesetts, 1978.

[32] V. Berk, A. Giani ir G. Cybenko, „Detection of Covert Channel Encoding in Network Packet Delays," Dartmouth College, Hanover, 2005.

[33] S. Cabuk, C. E. Brodley ir C. Shields, „IP Covert Timing Channels: Design and Detection," ACM, Washington, 2004.

[34] S. Gianvecchio, H. Wang, D. Wijesekera ir S. Jajodia, „Model-Based Covert Timing Channels: Automated Modeling and Evasion," įtraukta *RAID '08 Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection* , Cambridge, 2008.

[35] Z. Trabelsi, H. El-Sayed ir L. F. Rabie, „Traceroute Based IP Channel for Sending Hidden Short Messages," įtraukta *IWSEC 2006: Advances in Information and Computer Security*, Berlin, 2006.

[36] Z. Trabelsi ir I. Jawhar, „Covert File Transfer Protocol Based on the IP Record Route Option," *Journal of Information Assurance and Security,* nr. 5, pp. 064-073, 2010.

[37] T. Graf, „Messaging over IPv6 Destination Options," 2003.

[38] N. B. Lucena, G. Lewandowski ir S. J. Chapin, „Covert Channels in IPv6," įtraukta *PET 2005: Privacy Enhancing Technologies* , Berlin, 2005.

[39] W. Mazurczyk, M. Smolarczyk ir K. Szczypiorski, „Retransmission Steganography Applied," įtraukta *Multimedia Information Networking and Security (MINES)*, Jiangsu, 2010.

[40] J. Rutkowska, „The Implementation of Passive Covert Channels in the Linux Kernel," įtraukta *Chaos Communication Congress*, 2004.

[41] S. J. Murdoch ir S. Lewis, „Embedding Covert Channels into TCP/IP," įtraukta *IH 2005: Information Hiding* , Berlin.

[42] J. Giffin, R. Greenstadt, P. Litwack ir R. Tibbetts, „Covert Messaging Through TCP Timestamps," įtraukta *PET 2002: Privacy Enhancing Technologies*, Berlin, 2003.

[43] R. C. Chakinala, A. Kumarasubramanian, R. Manokaran, G. Noubir, C. P. Rangan ir R. Sundaram, „Steganographic Communication in Ordered Channels," įtraukta *IH 2006: Information Hiding*, Berlin, 2006.

[44] X. Luo, E. W. W. Chan ir R. K. C. Chang, „CLACK: A Network Covert Channel Based on Partial Acknowledgment Encoding," įtraukta *Communications, 2009. ICC '09*, Dresden, 2009.

[45] X. Luo, E. W. W. Chan ir R. K. C. Chang, „Cloak: A Ten-Fold Way for Reliable Covert Communications," įtraukta *ESORICS 2007: Computer Security – ESORICS 2007*, Berlin, 2007.

[46] X. Luo, E. W. W. Chan ir R. K. C. Chang, „TCP covert timing channels: Design and detection," įtraukta *Dependable Systems and Networks With FTCS and DCC, 2008*, Anchorage, 2008.

[47] W. Mazurczyk, M. Smolarczyk ir K. Szczypiorski, „On information hiding in retransmissions," *Telecommunication Systems ,* t. 52, nr. 2, p. 1113–1121, 2011.

[48] O. I. Abdullaziz, V. T. Goh, H.-C. Ling ir K. Wong, „Network Packet Payload Parity Based Steganography," įtraukta *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, Selangor, 2013.

[49] 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,“ 2003.

[50] W. Mazurczyk ir K. Szczypiorski, „Covert Channels in SIP for VoIP signalling,“ *Global E-Security. Communications in Computer and Information Science,* t. 12, pp. 65-72, 2008.

[51] W. Bender, D. Gruhl, N. Morimoto ir A. Lu, „Techniques for data hiding,“ *IBM SYSTEMS JOURNAL,* t. 35, nr. 3-4, pp. 313-336, 1996.

[52] M. V. Horenbeeck, „Deception on the network: thinking di erently about covert channels,“ įtraukta *7th Australian Information Warfare and Security Conference*, Edith, 2006.

[53] M. Bauer, „New covert channels in HTTP: adding unwitting Web browsers to anonymity sets,“ įtraukta *WPES '03 Proceedings of the 2003 ACM workshop on Privacy in the electronic society*, Washington, 2003.

[54] Z. Kwecka, „Application Layer Covert Channel Analysis and Detection,“ Napier University, Edinburgh, 2006.

[55] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan ir D. Karger, „Infranet: Circumventing Web Censorship and Surveillance,“ įtraukta *11th USENIX Security Symposium*, Berkeley.

[56] X.-g. Zou, Q. Li, S.-H. Sun ir X. Niu, „The research on information hiding based on command sequence of FTP protocol,“ įtraukta *KES'05 Proceedings of the 9th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part III*, Berlin, 2005.

[57] T. M. Gil, „NSTX (IP-over-DNS) HOWTO,“ 2010. [Tinkle]. Available: http://thomer.com/howtos/nstx.html. [Kreiptasi 09 05 2017].

[58] L. P. Deutsch, E. Ekman ir A. Bezemer, „Official git repo for iodine dns tunnel,“ Aladdin Enterprises, 2014. [Tinkle]. Available: https://github.com/yarrick/iodine. [Kreiptasi 09 05 2017].

[59] T. Pietraszek, „DNScat,“ 31 10 2004. [Tinkle]. Available: http://tadek.pietraszek.org/projects/DNScat/index.html. [Kreiptasi 09 05 2017].

[60] R. Bowes, „dnscat2,“ 2015. [Tinkle]. Available: https://github.com/iagox86/dnscat2. [Kreiptasi 09 05 2017].

[61] O. Dembour ir N. Collignon, „Dns2tcp,“ 16 06 2010. [Tinkle]. Available: http://www.hsc.fr/ressources/outils/dns2tcp/. [Kreiptasi 09 05 2017].

[62] Cisco Systems, Understanding Voice over IP Protocols, 2002.

[63] L. Y. Bai, Y. Huang, G. Hou ir B. Xiao, „Covert Channels Based on Jitter Field of the RTCP Header,“ įtraukta *Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, 2008.

[64] W. Mazurczyk ir K. Szczypiorski, „Covert Channels in SIP for VoIP signalling,“ p. 68, 2008.

[65] D. J. Barrett, R. E. Silverman ir R. G. Byres, SSH The Secure Shell. The Definitive Guide, Sebastopol: O'Reilly Media, Inc., 2001.

[66] N. B. Lucena, J. Pease, P. Yadollahpour ir S. J. Chapin, „Syntax and Semantics-Preserving Application-Layer Protocol Steganography," įtraukta *Information Hiding. IH 2004. Lecture Notes in Computer Science*, Berling, 2004.

[67] V. Guruswami, „Introduction to Coding Theory: Notes 1: Introduction, linear codes," Carnegie Mellon University, Pittsburgh, 2010.

[68] V. Stakėnas, Kodai ir šifrai, Vilnius: Naujoji Vilnia, 2006.

[69] J. I. Hall, Notes on Coding Theory. Chapter 4: Hamming Codes, Michigan: Michigan State University, 2001.

[70] S. Lin ir J. Daniel J. Costello, Error Control Coding: Fundamentals and Applications, New Jersey: Prentice-Hall, Inc., 1983.