KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS IR INFORMACINIŲ TECHNOLOGIJŲ SAUGOS STUDIJŲ PROGRAMA

MANTAS LASAUSKAS

P2P (PEER TO PEER) TINKLŲ ANONIMIŠKUMO UŽTIKRINIMO METODŲ ĮVERTINIMAS

Magistro baigiamasis darbas

Darbo vadovas
prof. dr. R. Plėštys

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS IR INFORMACINIŲ TECHNOLOGIJŲ SAUGOS STUDIJŲ PROGRAMA

MANTAS LASAUSKAS

P2P (PEER TO PEER) TINKLŲ ANONIMIŠKUMO UŽTIKRINIMO METODŲ ĮVERTINIMAS

Magistro baigiamasis darbas

Recenzentas
doc. dr. N. Morkevičius
2013-05-27

Darbo vadovas
prof. dr. R. Plėštys
2013-05-27

Darbą atliko:
Mantas Lasauskas
2013-05-27

KAUNAS, 2013

AUTORIŲ GARANTINIS RAŠTAS

DĖL PATEIKIAMO KŪRINIO

2013 - 05 - 27 d.
Kaunas

Autoriai, **Mantas Lasauskas**
(vardas, pavardė)

_____,

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama – Kūrinys) **„P2P (Peer to peer) tinklų anonimiškumo užtikrinimo metodų įvertinimas“**
(kūrinio pavadinimas)

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autoriai

Mantas Lasauskas	_____
(vardas, pavardė)	(parašas)
_____	_____
(vardas, pavardė)	(parašas)
_____	_____
(vardas, pavardė)	(parašas)
_____	_____
(vardas, pavardė)	(parašas)

SANTRAUKA

Anonimiškumas tapo labai svarbiu faktoriumi, saugant verslo ar privačią informaciją. Per pastaruosius keletą metų, taikomosios P2P programos tapo nemaža mūsų virtualios veiklos dalimi bendraujant balsu, susirašinėjant ar apsikeičiant duomenimis. Dauguma P2P tinklų vartotojų susirūpinę, kad visi jų veiksmai tinkle ir tapatybė gali būti nesunkiai atskleidžiama kitų tinklo vartotojų, todėl, šiais laikais, yra pasiūlyta daug įvairių priemonių, skirtų užtikrinti anonimiškumą P2P tinkle. Šios priemonės ir metodai yra suprojektuoti siekiant trijų pagrindinių tikslų: apsaugoti informacijos siuntėjo tapatybę, apsaugoti duomenų paieškos iniciatorių ir apsaugoti siunčiamos informacijos turinį.

Išanalizavus anoniminių tinklų ir sistemų veikimo principus buvo nustatyta, kad įvairiose publikacijose pateikiami galutiniai anonimiškumo analizės rezultatai be detalaus matematinio modelio. Buvo suformuluota pagrindinė problema – kaip matematiškai pamatuoti įvairių P2P tinklų anonimiškumą.

Siekiant išmatuoti anonimiškumą, buvo sukurta ir pritaikyta P2P tinklų anonimiškumo įvertinimo metodika, įgalinanti apskaičiuoti anonimiškumo laipsnį tiek struktūrizuotos, tiek ir nestruktūrizuotos P2P sistemos atveju. Struktūrizuotos P2P sistemos anonimiškumo laipsnis visada yra mažesnis už nestruktūrizuotos P2P anonimiškumo laipsnį, nes informaciją apie galimus maršrutus turi nemažai tinklo mazgų.

Pasiūlytas modelis, įgalinantis įvertinti P2P sistemos entropiją, kai tunelio ilgis yra atsitiktinis. Šis modelis leidžia tiksliau apskaičiuoti visos P2P sistemos anonimiškumą.

Anonimiškumo įvertinimo metodas ir jo panaudojimo metu gauti rezultatai yra svarbūs kuriant anonimines P2P sistemas, pagrįstas atsitiktinių maršrutų sudarymo principais.

SUMMARY

Anonymity has become a very important factor in protecting business or private information. Over the past few years P2P applications have become a significant part of our virtual activities such as voice communication, messaging or data exchange. Most of the P2P network users are concerned that all their actions in the network and their identity can be easily revealed by other network users. For that reason a variety of tools have been proposed to ensure the anonymity of P2P network. These tools and techniques are designed with three main purposes: to protect the identity of the sender, to protect the identity of data retrieval initiator and to protect data content which is being sent across the network.

After analyzing the principles of anonymous networks and systems, it has been found that in various publications the final results of analysis are included without explicit mathematical model. The main problem was formed – how to measure the anonymity of various P2P networks mathematically.

Anonymity evaluation methodology was developed and adapted to P2P networks in order to measure anonymity. This methodology enables to calculate anonymity degree of both structured and unstructured P2P systems.

Anonymity degree of structured P2P system is always lower than in unstructured system because in structured system nodes know the topology of the whole network and can simply get information about the possible routes.

The proposed mathematical model allows to evaluate entropy of P2P system with random tunnel length. This model allows to calculate the total anonymity of P2P systems more precisely.

Anonymity evaluation method and results obtained by it are important for the development of anonymous P2P systems based on random routing principles.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

1. ĮVADAS

Didėjantys perduodamos informacijos mastai turi lemiamos įtakos visuomenės ekonominiam vystymuisi, bet kartu iškelia naujų problemų informacijos saugumo užtikrinimo srityje. Pastaraisiais metais asmens duomenų privatumo užtikrinimas yra viena iš aktualiausių informacijos saugos uždavinių.

Šio uždavinio sprendimui esamų tinklų pagrindu kuriami taip vadinami perdangos tinklai, kurie pasižymi naujomis savybėmis, lyginant su esamais kompiuterių tinklais. Viena iš tokių tinklų atmainų yra taip vadinami P2P (peer-to-peer) tinklai. Jų pagalba vyksta informacijos mainai ne tarp visų IP tinklo mazgų, bet tarp tų tinklo mazgų, kurie registruojasi į šiuos tinklus.

Esamo IP tinklo mazgai gali būti piktavaliai, kurie įvairiais būdais stengiasi atskleisti tinklo vartotojų asmeninę arba atskiroms verslo įmonėms skirtą informaciją. Kartais pati informacija gali būti neslapta, bet turi būti paslėptas tos informacijos siuntėjas arba gavėjas. Tokiu atveju turi būti paslėpti informacijos siuntimo ir informacijos gavimo mazgai – t.y. tiek informacijos siuntėjas, tiek ir informacijos gavėjas turi išlikti anonimiški. Tai gali užtikrinti anoniminiai P2P tinklai.

Sukurta gana daug tipų tinklų, turinčių anonimiškumo užtikrinimo savybių (MIX-NET, OPENNET, DARKNET, FREENET, MUTE, MORPHMIX, TARZAN, AP3), tačiau jų anonimiškumas dar nėra pakankamai įvertintas. Anonimiškumą celinio tipo tinkle nagrinėjo darbo vadovas Rimantas Plėštys [1]. Anonimiškumo klausimais daug dirba Shandong Institute of Business & Technology (Dapeng Cheng), Ball State University Muncie, Indiana, USA (Baijian Yang), Rice University, Houston, TX, USA, University of Washington, Seattle, WA, USA, University of Texas at Arlington, Tsinghua University, Beijing, China ir kiti. Jų publikacijose pateikiami galutiniai analizės rezultatai be detalaus matematinio modelio, išskyrus paskutiniąją nuorodą. Kuriant anoniminius tinklus, bei rengiant mokomąją medžiagą pasigendama gilesnės tokių tinklų veikimo analizės. Remiantis atlikta anoniminių P2P tinklų analize suformuluotas darbo tikslas.

Šio darbo tikslas:

Išanalizuoti anoniminių P2P tinklų sudarymo principus ir įvertinti jų anonimiškumo užtikrinimo metodus.

Tikslui pasiekti sprendžiami šie uždaviniai:

1. Išnagrinėti P2P tinklų sudarymo principus
2. Išanalizuoti P2P tinklų veikimo principus
3. Išanalizuoti esamas P2P tinklų anonimiškumo įvertinimo metodus
4. Sudaryti P2P tinklų anonimiškumo įvertinimo matematinį modelį
5. Naudojantis matematiniu modeliu atlikti P2P tinklų anonimiškumo įvertinimą

6. Suformuluoti rekomendacijas reikiamo lygio anonimiškumo užtikrinimui

Darbo rezultatai: sukurtas P2P tinklų anonimiškumo įvertinimo matematinis modelis nuo žinomų modelių skiriasi tuo, kad įgalina įvertinti P2P tinklo, sudaryto iš atsitiktinio skaičiaus ir kintamo ilgio tunelių, anonimiškumą. Esamuose darbuose vertinamas P2P tinklo, sudaryto tik iš vieno pastovaus ilgio tunelio, anonimiškumas.

2. ANALIZĖ

2.1. P2P tinklai [2] [3] [4]

P2P tinklų samprata .

P2P (angl. *Peer To Peer*) tinklas remiasi paskirstytų tinklų technologija, kurios pagrindas – tinklu apjungti lygiaverčiai kompiuteriai, kurių kiekvienas teikia kitiems tinklo dalyviams tam tikrą dalį resursų (priešingai nei centralizuotos architektūros atveju , kuomet resursus teikia tik itin mažas serverių skaičius). Tikrame P2P architektūros modelyje neegzistuoja nei serverių, nei klientų – tik lygiaverčiai mazgai, atliekantys tiek serverio, tiek kliento funkcijas. Visgi kai kurie tinklai naudoja serverių sąvoką: pvz. Gnutella (failų apsikeitimo) tinkle serveryje saugoma informacija apie kitus tinkle esančius vartotojus.

P2P sistema yra abstraktus persidengiantis tinklas, veikiantis taikymo sluoksnyje, virš fizinio tinklo topologijos. Tokie sluoksniai yra naudojami taškų indeksacijai ir vienas kito atradimui, taip pat jie suteikia P2P sistemai nepriklausomybę nuo fizinio tinklo topologijos. P2P tinkle turinio apsikeitimas vykdomas tiesiogiai per pagrindinį interneto protokolo (IP) tinklą. Anoniminės P2P sistemos yra išimtis. Jos turi keletą papildomų maršrutizavimo sluoksnių, kad paslėptų šaltinio tapatybę ar užklausų tikslą.

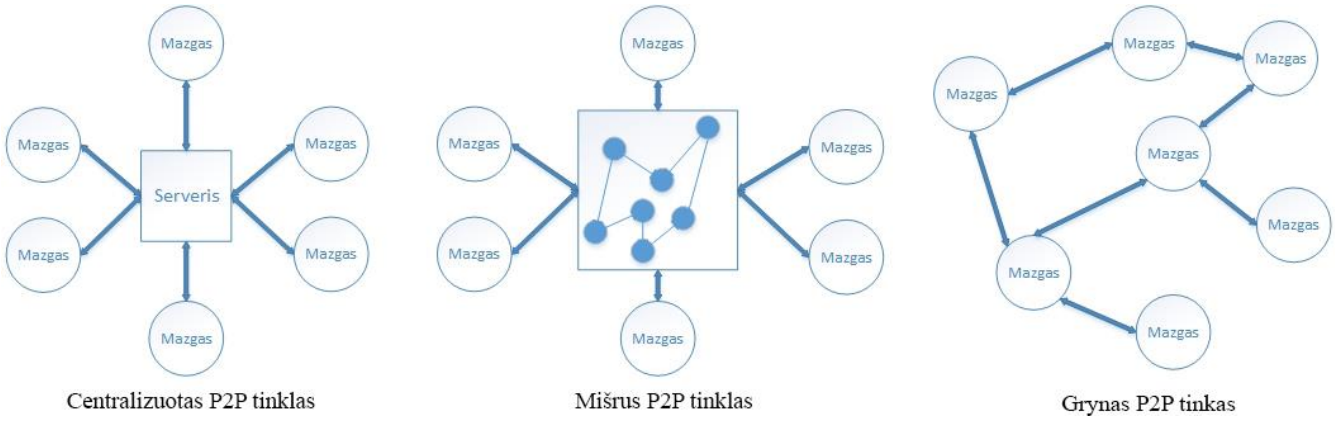
Stephena‘s ir Diomidis siūlo labiau detalizuotą P2P apibrėžimą:

P2P sistemos yra paskirstytos sistemos, sudarytos iš sujungtų mazgų, galinčių savarankiškai organizuotis į tinklo topologijas su tikslu dalintis ištekliais tokiais kaip turinys, procesoriaus ciklai, duomenų laikymo talpa ir tinklo pralaidumas.

P2P tinklų klasifikacija

Klasifikacija – pagal tinklo centralizacijos laipsnį:

- *Centralizuoti P2P tinklai:* Tokie tinklai turi centrinį serverį, kuris stebi visą informacijos judėjimą tame tinkle. Tinklo mazgai naudoja serverį kitų mazgų paieškai, kurie turi jiems reikiamą informaciją. Tokio tinklo trūkumas – centrinio serverio mechanizmas, kurio neveikimo atveju, tinklas nebefunkcionuotų.
- *Gryni P2P tinklai:* Dar žinomi kaip visiškai decentralizuoti tinklai. Visi tinklo mazgai atlieka serverio-kliento rolę. Tokiame tinkle neegzistuoja centrinis serveris arba maršrutizatorius.
- *Mišrūs P2P tinklai:* Dar žinomi kaip pusiau centralizuoti tinklai. Toks tinklas turi ir gryno ir hibridinio P2P tinklo savybių. Kai kurie tinklo mazgai įgauna kiek kitokį statusą: turi indeksuoti failus, saugomus kitų tinklo mazgų ir padėti mazgams užmegzti tarpusavio ryšius. Tokie mazgai dar vadinami *Super Mazgais*.



2.1 pav. P2P tinklų klasifikacija

Kita galima klasifikacija – pagal tinklo struktūrą:

- *Nestruktūrizuotas P2P tinklas:* nestruktūrizuotas P2P perdangos tinklas, yra toks tinklas, kuriame mazgas, norintis persiųsti žinutę kitam mazgui, remiasi gretimais tinklo mazgais. Tokiame tinkle mazgas nežino visos tinklo topologijos, todėl naudojami įvairūs mazgų paieškos algoritmai. Mazgai, norėdami surasti reikiamą informaciją, tiesiog „užtvindo“ visą tinklą užklausomis su ieškomų failų pavadinimais. Nesėkmės tikimybė tokiame tinkle yra daug didesnė net ir po to, kai tinklas užtvindomas failų užklausomis, kadangi tarp mazgo ir jo saugomo turinio nėra jokios koreliacijos. Užklausų judėjimas tinkle mažina darbo našumą. Nestruktūrizuotų tinklų pavyzdžiai: Napster, Gnutella, Kazaa ir t.t.
- *Struktūrizuotas P2P tinklas:* Struktūrizuoto P2P perdangos tinklo architektūra, yra tokia architektūra, kurioje mazgai bendradarbiauja tarpusavyje, siekiant kaupti ir išlaikyti tinklo topologijos informaciją, kuri leidžia bet kuriuo metu tinklo mazgui pasiekti bet kurį kitą tinklo mazgą, taigi struktūrizuotame P2P tinkle mazgai žino visą tinklo struktūrą. Tam dažniausiai naudojamos DHT lentelės. Perdangos tinklo topologijos yra griežtai laikomasi ir turinio failai saugomi tik jiems skirtoje vietoje. Šie tinklai saugo paskirtstyto maršrutizavimo lenteles, siekiant efektyviai atsakyti į užklausas. Struktūrizuotų tinklų pavyzdžiai: Chord, Pastry, Tapestry, Can ir t.t.

P2P tinklų privalumai

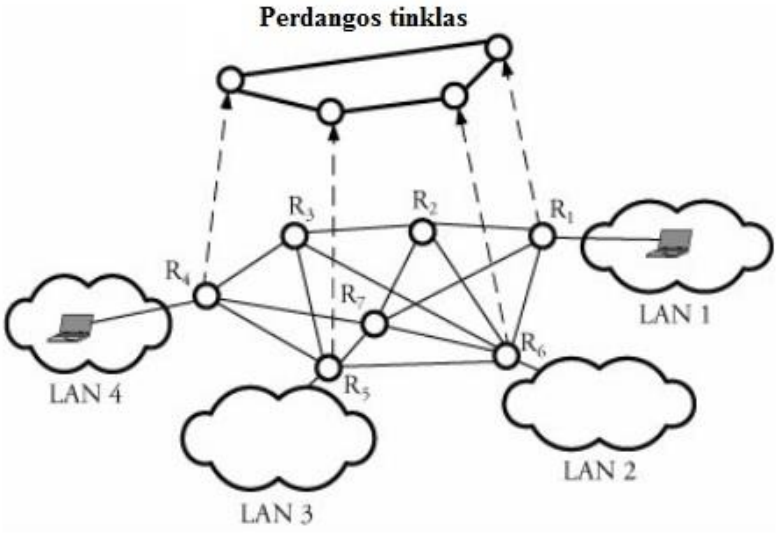
1. Pagrindinis P2P tinklo privalumas – efektyvus resursų panaudojimas, kadangi kiekvienas tinklo mazgas atskirai suteikia resursų (pralaidumas, duomenų saugojimas, skaičiavimo galia). Resursų švaistymas palyginus su kitais tinklais (kur vartotojų kompiuterių resursai visiškai neišnaudojami) yra mažesnis.
2. Tinklo mazgai dalindamiesi ir kaupdami informaciją didina tinklo mastelį.

3. Duomenų paskirstymo per atskirus tinklo mazgus mechanizmas padidina tinklo patikimumą. Taip pat neegzistuoja vienintelio pažeidimo taško.
4. Tinklo administravimo paprastumas (viena iš priežasčių, kodėl P2P technologija susilaukė tokio didžiulio populiarumo).
5. Vartotojams prisijungimas prie tinklo ir atsijungimas nuo jo yra paprastesnis palyginus su kitais tinklais.

Perdangos tinklai [5]

Perdangos tinklas (angl. *Overlay network*) yra kompiuterinis tinklas, orientuotas konkrečiai programai, sudarytas ant kito tinklo. Perdangos tinklas sukuria virtualią topologiją ant esamo tinklo topologijos. Tokio tipo tinklas kuriamas siekiant apsaugoti jau egzistuojančio tinklo struktūrą nuo naujų protokolų diegimo, kurių testavimo stadija reikalauja internetinio ryšio. Perdangos tinklai izoliuoja testavimo metu naudojamus paketus nuo pagrindinio tinklo infrastruktūros.

P2P perdangos tinklai iš prigimties yra informacijos platinimo sistemos, neturinčios jokios hierarchinės struktūros ir centralizuoto valdymo. Tinklo mazgai sudaro perdangos tinklą, kuris yra sukonstruotas ant IP tinklų. P2P perdangos tinklai siūlo: gerą maršrutizavimo architektūrą, efektyvią duomenų paiešką, aplinkinių mazgų parinkimą, perteklinį informacijos saugojimą, pastovumą, hierarchinį vardų priskyrimo mechanizmą, patikimumą ir autorizaciją, anonimiškumą, didelį tinklo praplečiamumą ir atsparumą gedimams.



2.2 pav. Perdangos tinklo schema

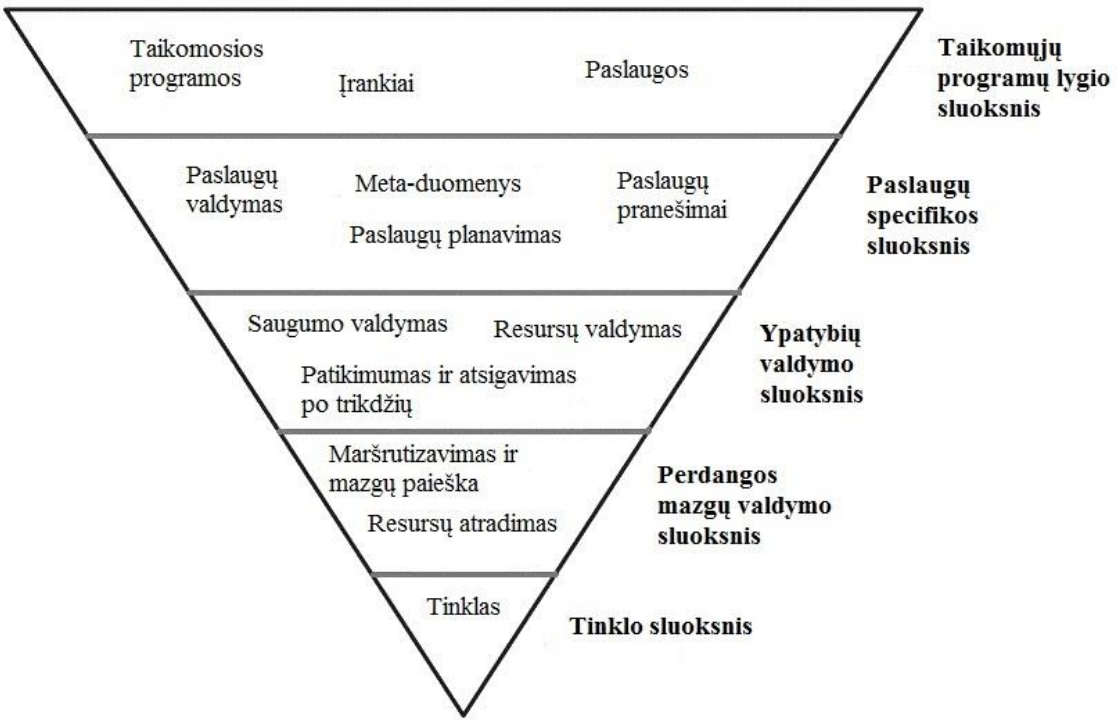
Perdangos tinklas, kuris yra sukonfigūruotas ir realizuotas ant plačiajuosčio tinklo topologijos, pavaizduotas 2.2 pav. Maršrutizatoriai R_1, R_6, R_5, R_4, dalyvauja perdangos tinklo kūrime, kur tarpusavio ryšiai yra realizuoti kaip perdangos loginiai ryšiai. Geras perdangos tinklo pavyzdys yra

P2P tinklas, kuris veikia virš IP tinklo topologijos. Perdangos tinklai negali kontroliuoti kaip paketai yra maršrutizuojami pagrindiniame tinkle, tarp siuntėjų/gavėjų perdangos tinklo mazgų. Tačiau šie tinklai gali kontroliuoti perdangos mazgų seką, panaudojus pranešimų perdavimo funkciją, prieš pasiekiant tikslą.

Perdangos tinklo gali prireikti dėl įvairių priežasčių. Perdangos tinkle galima siųsti pranešimus gavėjams, iš anksto nežinant jų IP adresų. Kai kuriais atvejais perdangos tinklas gali būti pasiūlytas kaip būdas pagerinti internetinio tinklo maršrutizavimą, siekiant užtikrint geresnę duomenų transliavimo kokybę. Kai kurių perdangos tinklų įgyvendinimui gali prireikti perkonfigūruoti IP tinklo maršrutizatorius. Tokiu atveju, perdangos tinklo mazgas gali būti įdiegtas galutinio vartotojo kompiuteryje, kuriame veikia perdangos protokolo programinė įranga, neprašant interneto tiekėjų pagalbos.

Perdangos tinklai – savaime prie situacijos prisitaikantys tinklai. Nustojus funkcionuoti perdangos tinklo mazgui, perdangos algoritmo pagalba, atkuriama tinklo struktūra, reikalinga jo pilnavertiškam funkcionavimui.

P2P perdangos tinklo modelį galima interpretuoti kaip tinklą, susidedantį iš plataus spektro ryšių struktūrų, kurios apibrėžia visiškai paskirstyto, kooperatinio tinklo konstrukciją, kurioje tinklo mazgai kuria savarankiškai veikiančią sistemą. Abstrakti P2P perdangos tinklo architektūra, iliustruojanti perdangos ryšių struktūros komponentus, pavaizduota 2.3 pav.

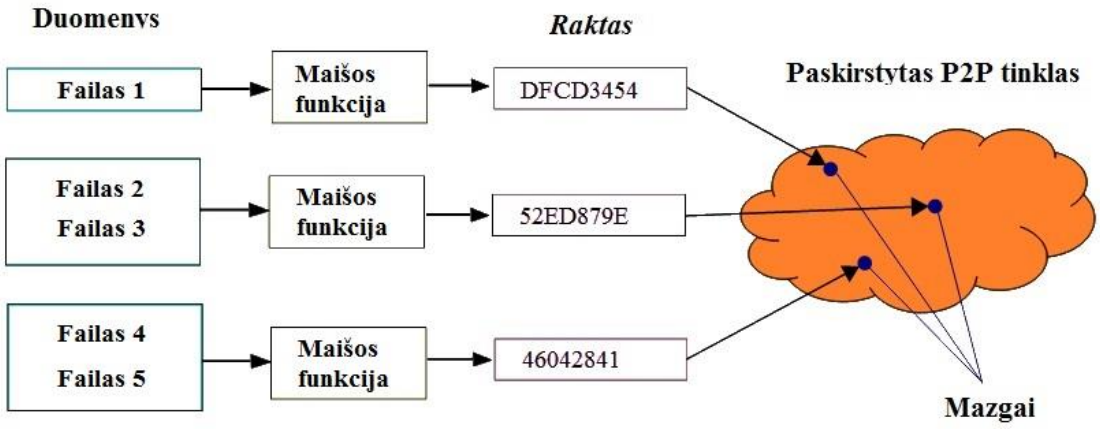


2.3 pav. Abstraktaus P2P perdangos tinklo architektūra [5].

Tinklo sluoksnis apibūdina vartotojų kompiuterių, prisijungusių prie interneto, tinklo charakteristikas. Mazgai pastoviai prisijungia/atsijungia prie P2P tinklo. *Perdangos mazgų valdymo sluoksnis* apima mazgų valdymą, į kurį įeina mazgų paieška ir maršrutizavimo algoritmai tinklo darbo optimizavimui. *Ypatybių valdymo sluoksnis* atsakingas už tinklo saugumo, patikimumo, sistemos atsistatymo po trikdžių ir bendrų resursų eksploatacinės parengties aspektus, siekiant išlaikyti P2P sistemos patikimumą. *Paslaugų specifikos sluoksnis* palaiko pamatinę P2P infrastruktūrą ir taikomųjų programų specifikos komponentus, planuojant lygiagrečias ir daug skaičiavimo reikalaujančias užduotis, turinio ir failų valdymą. *Taikomųjų programų lygio sluoksnis* aprūpintas įrankiais, programomis ir paslaugomis, turinčiais specifinį funkcionalumą ir įgyvendintais virš pamatinės P2P perdangos infrastruktūros.

Platinamos adresų lentelės (DHT) [6] [7]

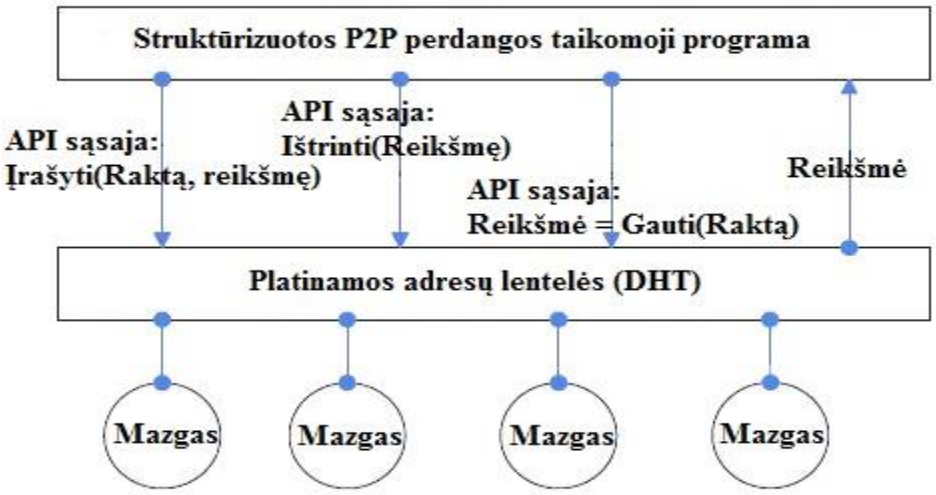
Platinamos adresų lentelės (angl. *Distributed hash tables* arba *DHT*) yra decentralizuotų platinimo sistemų klasė, kuri teikia mazgų paieškos paslaugas P2P tinkle, panašiai kaip maišos lentelės (angl. *hash tables*): rakto ir reikšmės (angl. *key* ir *value*) poros yra patalpinamos į platinamą adresų lentelę ir kiekvienas tinklo mazgas bet kuriuo metu pagal pateiktą raktą gauna informaciją (pav. 2.4). DHT lentelių saugojimas yra patikėtas tam tikro skaičiaus tinklo mazgų grupei. Lentelių saugojimo būdas realizuotas taip, kad bet koks pasikeitimas toje mazgų grupėje nedaro didelės įtakos viso P2P tinklo darbui. Tokia metodika leidžia DHT lentelėse išplėsti įrašus iki labai didelio mazgų skaičiaus, suvaldyti didelius tinklo mazgų prisijungimo prie tinklo ir atsijungimo nuo jo srautus, bei suvaldyti tinklo klaidas.



2.4 pav. *Raktų* priskyrimas duomenų objektams P2P perdangos tinkle [6].

DHT paremtose sistemose duomenų objektams priskiriami unikalūs raktai (angl. *keys*). Šiuos raktus P2P perdangos tinklo protokolas susieja su jų unikaliais perdangos tinklo mazgais. P2P

perdangos tinklas palaiko kintamos apimties {*raktas, reikšmė*} porų saugojimą ir gavimą perdangos tinklo viduje (pav. 2.5). *Raktui* pritaikius operaciją (*įrašyti*(*raktą,reikšmę*)), inicijuojama paieškos ir gavimo operacija *(reikšmė = gauti(raktą))*, kurios tikslas yra gauti ir išsaugoti duomenų objektą, atitinkantį *rakto* parametro reikšmę. Į šią operaciją įeina užklausų maršrutizavimas mazgams, atitinkantiems *raktą*.



2.5 pav. DHT technologija paremto P2P perdangos tinklo programinės įrangos sąsaja [5].

Kiekvienas tinklo mazgas saugo kaimyninių mazgų *ID* reikšmes ir IP adresus. Paieškos užklausos yra nukreipiamos per visą perdangą įvairiais keliais progresiniu būdu. Skirtingos DHT paremtos sistemos turi skirtingas duomenų objektų ir jų *raktų* organizavimo schemas ir maršrutizavimo strategijas. Teoriškai, DHT technologija paremtos sistemos turi garantuoti, kad bet kuris duomenų objektas gali būti aptiktas atlikus nedidelį skaičių O(*logN*) šuolių tarp perdangos mazgų. *N* yra tinklo mazgų skaičius. Kelias tarp mazgų pamatiniame tinkle gali visiškai skirtis nuo kelio DHT technologijos perdangos tinkle. Dėl tos priežasties paieškos vėlinimo laikas DHT technologija paremtame P2P perdangos tinkle gali būti ganėtinai didelis ir neigiamai paveiktų taikomųjų programų veikimo spartą.

DHT suformuoja infrastruktūrą, kuri panaudojama daug sudėtingesnių paslaugų realizavimui: failų skirstymo sistemoms, P2P failų dalinimosi ir turinio platinimo sistemoms, srities vardų paslaugoms (DNS), momentiniams pranešimams ir *multicast* bei *anycast* technologijai.

Atakos P2P tinkluose [8]

P2P tinklų veikimo principas lemia pagrindinį viso tinklo pažeidžiamumą – tarp visų tinklo mazgų tam tikra dalis mazgų gali būti kenkėjiški. Šie kenkėjiški tinklo mazgai gali įvykdyti daug skirtingų atakų prieš tikrus tinklo mazgus, kurios įtakos tinklo integralumą ir saugą.

Bendru atveju atakos suprantamos kaip:

Metodai, kuriais siekiama sunaikinti, pakeisti arba užblokuoti informaciją, esančią kompiuteriuose arba atskiruose tinkluose.

Galimi trys atakų P2P tinkluose tipai:

- *Atakos prieš ryšio kanalus:* Šių atakų tikslas susilpninti arba nutraukti ryšį tarp dviejų tinklo mazgų.
- *Atakos prieš mazgų anonimiškumą:* Šio tipo atakos tikslas – atskleisti mazgo, kuris dalinasi informacija P2P tinkle, tapatybę.
- *Atakos prieš atskirus failus:* Šių atakų metu mėginama tinklo mazgų saugomus failus užteršti arba juos pakoreguoti.

Egzistuoja keli pagrindiniai atakų įgyvendinimo P2P tinkluose metodai:

- *IP adreso padirbinėjimas (klastojimas):* (angl. *IP spoofing*) Šios atakos metu atakuotojas, pasinaudojęs svetimu IP arba MAC adresu, gauna neteisėtą prieigą prie kompiuterio ar tinklo, kol jie galvoja, kad bendrauja su patikimu mazgu.
- *Duomenų srauto analizė:* Atakuotojas pasyviai stebės ir analizuos duomenų srautą, sukurtą įvairių P2P tinklo mazgų. Tokia analizė suteiks atakuotojui statistinės informacijos apie kai kuriuos tinklo mazgus. Ši informacija leis jam įgyvendint įvairias skirtingas atakas, aptariamas toliau.

Atakų rūšys [8]

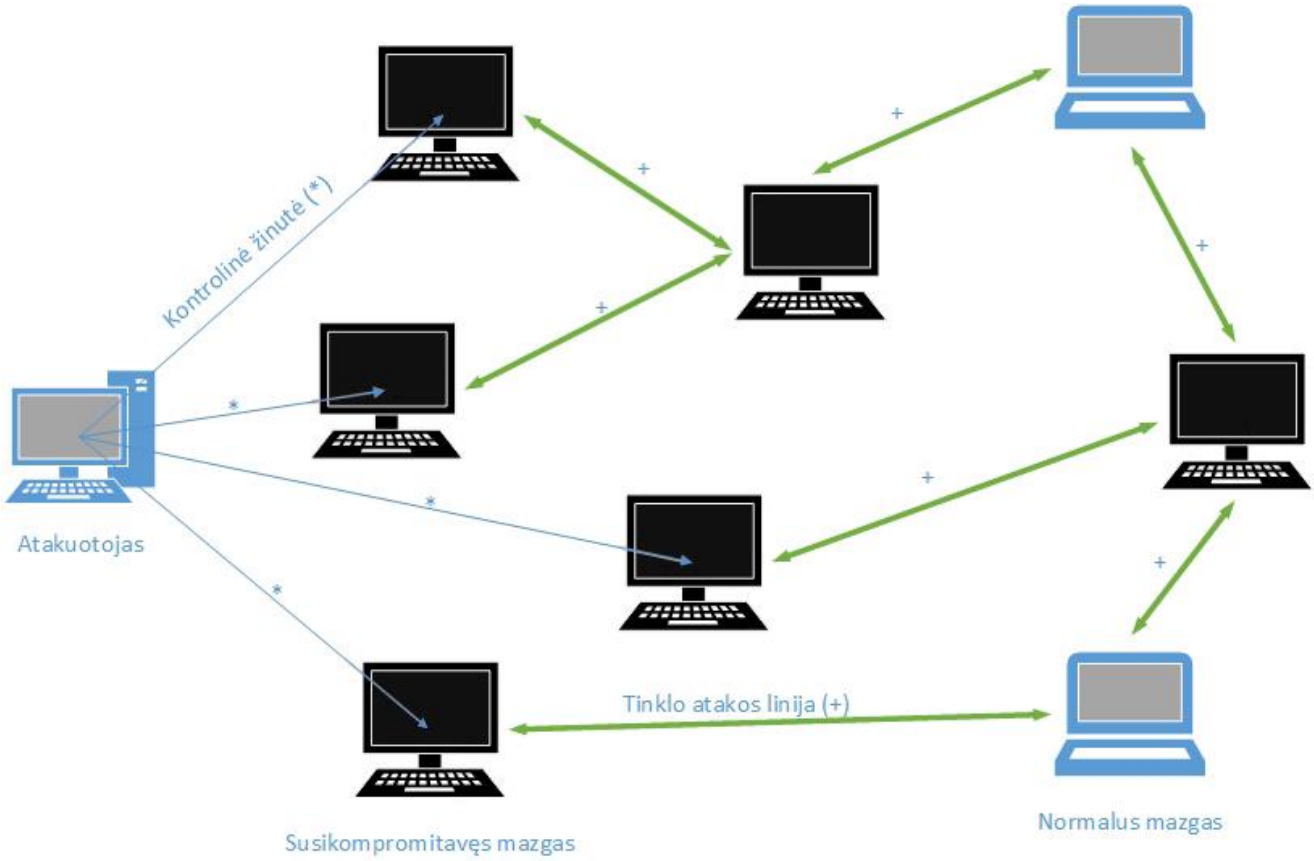
DOS ataka

DOS (angl. *Denial-of-Service*, liet. *Paslaugų užblokavimo*) ataka, kurios tikslas sustabdyti tinklo funkcionavimą. Piktavaliai mazgai, rengiantys šią ataką, bando užblokuoti kokią nors paslaugą, kurią teikia tinklas ar atskiras kompiuteris.

DOS atakos gali būti:

1. *Užtvindymo atakos:* (angl. *flooding*) Tai labiausiai paplitęs DOS atakos tipas. Kenkėjiški mazgai užtvindo tinklą su negaliojančiais netikrais paketais, paveikdami normalių paketų judėjimą tinkle.
2. *Procesorių apkrovos atakos:* Šios atakos privers aukos įrenginio procesorių daryti sudėtingus begalinius apskaičiavimus, taip jį visiškai apkraudami, todėl jis daug lėčiau atsakinės į užklausas arba iš vis nebeatsakinės.
3. *Gobšaus vartotojo atakos:* Atakos metu yra bandoma užkirsti norimam tinklo mazgui pasiekti kokią nors paslaugą, arba sužlugdyti konkrečią paslaugą, kurią naudoja atakuojamas mazgas.

Įtraukiant didelį kiekį mazgų atakos atlikimui, ši ataka tampa dar pavojingesnė. Tokiu atveju ji jau vadinasi DDOS (angl. *Distributed Denial-of-Service*, liet. *Paskirstytą paslaugų blokavimo*) ataka. Kompiuteriai (susikompromitavę tinklo mazgai), dalyvaujantys tokioje atakoje, dažniausiai būna užgrobti atakuotojo, kuris kažkokiu būdu įgyja nesankcionuotą prieigą prie jų. Atakuotojas slepiasi užgrobtuose kompiuteriuose, todėl būna labai sunku aptikti tikrąją atakos prigimtį.



2.6 pav. DDOS ataka vykdoma prieš ryšių kanalus.

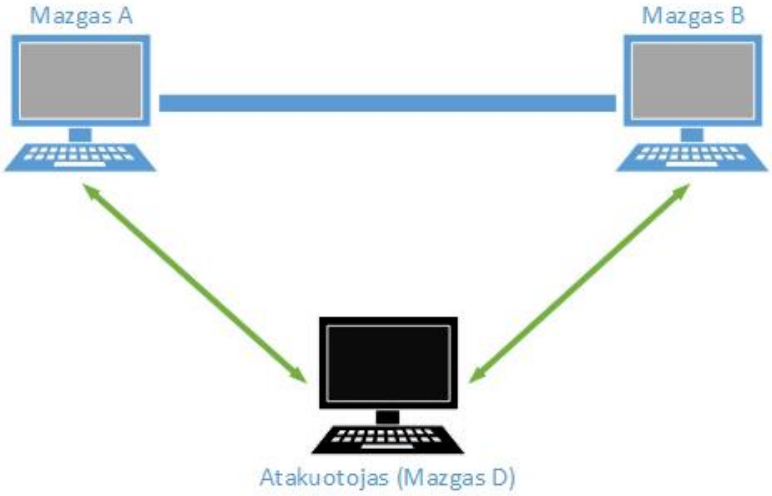
Ši ataka gali sutrikdyti normalų P2P tinklo darbą, ko pasekoje, gali būti paveiktas tinklo mazgų anonimiškumas. Iš kitos pusės, tas pats anonimiškumas padaro šią ataką dar efektyvesnę, kadangi yra labai sunku surasti ir pašalinti tinklo mazgą, kuris atlieka ataką.

***Žmogus viduryje ataka* [8]**

Angliškai ši ataka vadinasi: *Man-in-the-Middle Attack*. Jos metu piktavalis mazgas slepiasi ryšio kanale tarp dviejų tinklo mazgų ir visas duomenų srautas, kuriuo apsikeičia abu mazgai, pereina per piktavalį mazgą. Atakuojantis mazgas išlieka pasyvus visos atakos metu ir šnipinėdamas ryšį tarp mazgų, stengiasi surinkti kuo daugiau jam naudingos informacijos. Atakuotojas tampa aktyviu, kada sužino jam reikalinga informacija, ir jam prireikia truputį pakoreguoti arba įterpti į informacijos srautą suklastotus duomenis.

Galima sudėlioti tokį šios atakos scenarijų:

- Mazgas *A* išsiunčia užklausa dėl norimo failo, o mazgas *B* į ją atsako, išsiųsdamas patvirtinimo žinutę.
- Piktavalis mazgas *D* perima žinutę, siųstą mazgo *B* mazgui *A* ir pakeičia IP adreso ir prievado laukų reikšmes į tokias pat kaip savo.
- Mazgas *A* atsisiunčia failą ne iš mazgo *B*, o iš mazgo *D*, kuris bus netikras ir jame gali būti įdiegtas kenkėjiškas kodas.



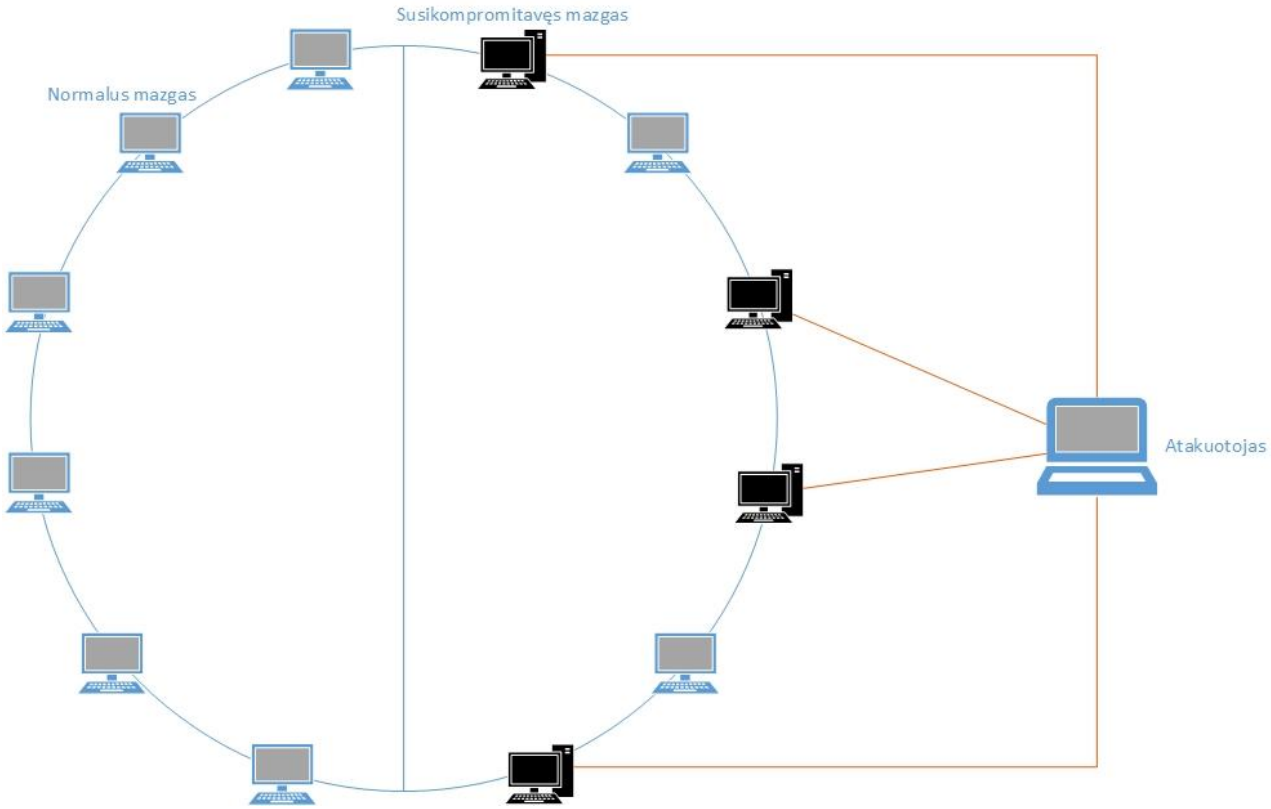
2.7 pav. Žmogus viduryje ataka.

Šios atakos tikslai priklauso nuo tinklo naudojamo protokolo. Bendru atveju, dažniausi atakos tikslai: tapatybės klastojimas ir padirbtos informacijos siuntinėjimas.

Atakuotojas pasinaudoja P2P tinklo maršrutizavimo algoritmo veikimo principu, kad pasislėpti tarp tinklo mazgų.

***Sybil ataka* [8]**

Atakos metu atakuotojas bando užgrobti dalį tinklo mazgų prisistatinėdamas skirtingomis tapatybėmis. Atakuotojas bandys pastatyti sau pavaldžius mazgus toje pačioje ID erdvėje. Tai padeda atakuotojui padaryti tinklą pažeidžiamesniu, kadangi jis gauna tam tikro tinklo segmento kontrolę. Taip pat jis gali kontroliuoti visas žinutes, kurios kerta jo valdomą tinklo segmentą.

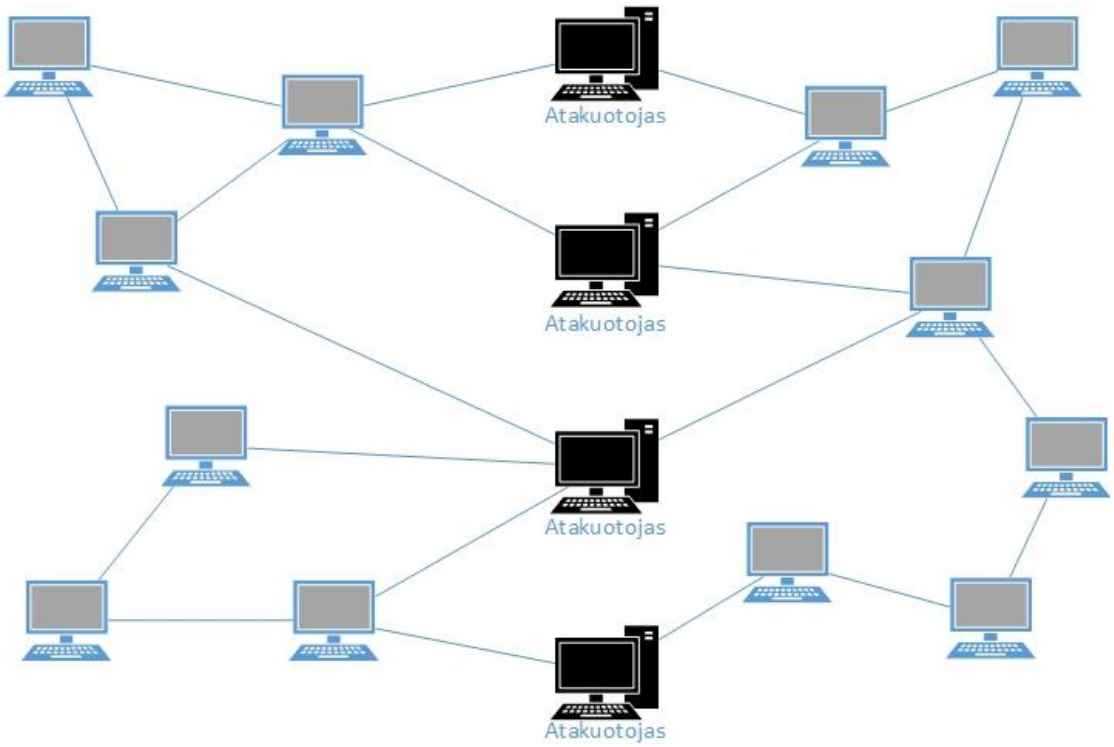


2.8 pav. Sybil atakos pavyzdys. Atakuotojas gauna dalies tinklo kontrolę.

Tokiu būdu atakuotojas gali pridaryti P2P tinklui nemažai žalos, kontroliuodamas ribotą kiekį tinklo mazgų. Šios atakos tikslai priklauso nuo kontroliuojamo tinklo mazgų skaičiaus. Gavus tinklo kontrolę, galima išnaudoti tinklo protokolą arba gauti prieigą prie tam tikrų failų, saugomų tinkle, ir juos užteršti. Atakuotojas gali praplėsti šią ataką iki *užtemimo* (angl. *eclipse*) atakos arba stipriai sulėtinti tinklą, nukreipiant visas užklausas neteisinga kryptimi.

***Užtemimo ataka* (angl. *eclipse)* [8]**

Galima būtų šią ataką perkvalifikuoti į didelio mąsto Žmogus viduryje ataką. Ataka kelia didelę grėsmę P2P tinklo funkcionavimui, kadangi strategiškai išdėstyti atakuotojo mazgai gali padalinti tinklą į du ar daugiau potinklius ir įvairūs duomenų srautai, kurie vaikšto tarp tinklo mazgų, praeis būtent per šiuos mazgus.



2.9 pav. Užtemimo ataka. Piktavaliai mazgai padalina tinklą į dvi dalis ir izoliuoja jas vieną nuo kitos.

Pasinaudojęs užtemimo ataka, piktavalis gali:

- paveikti P2P programų funkcionalumą neefektyviai nukreipiant tinklo žinutes,
- užblokuoti visas praeinančias žinutes, taip izoliuodamas atskirtas tinklo dalis,
- suklastoti žinutes, vaikščiojančias tarp dviejų mazgų,
- paveikti duomenis, kuriuos naudoja P2P programos, įterpiant užterštus failus arba prašant užterštų failų kitų mazgų vardu.

Susikirtimo ataka **[8]**

Geriausias būdas susekti mazgo tapatybę tinkle – stebėti jo elgseną ilgesnį laiką. Konkretus mazgas pademonstruos tipinius prisijungęs/atsijungęs periodus, laikui bėgant naudos tuos pačius resursus ir dažniausiai siųs užklausas tai pačiais adresais skirtingų sesijų metu. Taigi, stebint tam tikrą rinkinį aktyvių tinklo mazgų skirtingu metu, galima gauti įvairios naudingos informacijos: kada mazgas būna aktyvus ir kurie mazgai komunikuoja tarpusavyje ir kada. Surinkęs tam tikrą kiekį duomenų, atakuotojais gali pagal tai nuspręsti, kokią ataką vykdys.

Tapatybės ataka **[8]**

Atakos tikslas – pasisavinti siuntėjo, gavėjo arba abiejų tapatybes. Atakuotojas naudojasi įvairiais metodais, siekdamas pasisavinti kitų tinklo mazgų tapatybes:

- P2P tinkle paleisti Trojos arklį, kirminą ar kitą virusą ir po to ieškoti apsikrėtusių mazgų.
- Ilgą laiką pasyviai stebėti tinklą, apsimesti garbingu ir sąžiningu mazgu, ir įgavus kitų mazgų pasitikėjimą, pagrobti tapatybę/es.
- Sukurti konkrečiam atvejui (mazgui) pritaikytą virusą.

Be aukščiau pateiktų būdų, atakuotojas gali pasinaudoti ir kitomis atakomis.

2.2. Anonimiškos P2P sistemos [9]

Anonimiškumas tapo svarbia šiuolaikinių P2P sistemų charakteristika. Anonimiška P2P sistema ir tokia sistema, kurioje jos vartotojų tapatybė išlieka paslėpta panaudojant įvairius anonimiškumo užtikrinimo metodus ir slapyvardžius.

Didžioji dauguma anoniminių P2P sistemų yra „draugas-draugas“ tipo. Tai reiškia, kad tokių tinklų mazgai užmegs ryšį tik su patikimais, pažįstamais kaimyniniais mazgais. Bendravimas tarp skirtingų mazgų vyksta anonimiškai panaudojant įvairius maršrutizavimo metodus.

Anonimiškumas [10]

Kas yra anonimiškumas?

Anonimiškumas yra būsena, kada išlaikoma neatskleista tapatybė, panaudojus įvairias priemones siekiant ją atskleisti.

Anonimiškumo idėja – paslėpti faktą, kad buvo užmegztas ryšys tarp dviejų vartotojų, taip pat paslėpti tų vartotojų tapatybes sistemoje. Anonimiškumo sąlygą įvairiuose šaltiniuose aiškinama skirtingai, tačiau pagrindiniai principai išlieka nepakitę.

Komunikacijos tarp mazgų požiūriu, egzistuoja trys anonimiškumo tipai:

1) *Siuntėjo anonimiškumas:* kai slepiama vartotojo, kuris inicijuoja ryšio užmezgimą, tapatybė.
2) *Gavėjo anonimiškumas:* kai slepiama vartotojo tapatybė, kuris atsako į siuntėjo užklausas ir išsiunčia failus.
3) *Bendras anonimiškumas:* kai slepiama siuntėjo i gavėjo tapatybės vienas nuo kito ir nuo likusių tinklo mazgų.

Apibendrinus, anonimiškumo P2P tinkle tikslas – paslėpti konkretaus mazgo tapatybę nuo visų likusių tinklo mazgų.

Anonimiškumo aspektai pagal skirtingų vartotojų tipus ir jų veiklą sistemoje.

1) *Autoriaus anonimiškumas:* kai slepiama konkretų dokumentą sukūrusio autoriaus tapatybė.

2) *Leidėjo anonimiškumas:* kai slepiama vartotojo tapatybė, kuris tam tikrą failą ar dokumentą padarė viešai prieinamą P2P tinkle. Autorius ir leidėjas kartais būna tas pats vartotojas.
3) *Skaitytojo anonimiškumas:* slepiama tapatybė mazgo, kuris pasiuntė užklausą dėl konkretaus failo ar dokumento kitam tinklo mazgui.
4) *Serverio anonimiškumas:* slepiama konkretaus failo saugojimo vieta arba mazgo/serverio, kur patalpintas šis failas, tapatybė.
5) *Dokumento anonimiškumas:* kai slepiama dokumento turinio tapatybė nuo mazgo ar serverio kur tas dokumentas patalpintas.

Be aukščiau išvardintų sąvokų, egzistuoja dar viena – anonimiškumo laipsnis, kuris nusako anonimiškumo lygį, kurį gali suteikti įvairūs anoniminiai mechanizmai.

1. *Absoliutus privatumas:* suteikia vartotojui visišką privatumą ir atakuotojas neturi jokių šansų atskleisti jo tapatybę. Realiose sistemose praktiškai nepasiekiamas.
2. *Beveik neįtariamas:* atakuotojo požiūriu, tikimybė, kad aptiktas vartotojas galėjo atlikti tam tikrus veiksmus, yra tokia pati, kaip ir bet kurio kito tinklo mazgo.
3. *Tikėtinas nekaltumas:* atakuotojo požiūriu, tikimybė, kad aptiktas vartotojas galėjo atlikti tam tikrus veiksmus, yra mažesnė, negu, kad jis jų neatliko.
4. *Įmanomas nekaltumas:* tikimybė, kad aptiktas vartotojas atliko tam tikrus veiksmus yra didesnė, negu, kad jis jų neatliko.
5. *Demaskuotas:* atakuotojas gali identifikuoti žinutės siuntėją.
6. *Demaskuotas su įkalčiais***:** atakuotojas gali ne tik identifikuoti siuntėją, bet ir įrodyti jo tapatybę kitiems.

Toliau darbe nagrinėjamas anonimiškumo laipsnis bus išreikštas matematiškai ir jo reikšmė bus kintama intervale nuo 0 iki 1.

Anonimiškumo motyvacija

Yra daug priežasčių naudotis P2P anonimiškumo technologija, dauguma jų yra bendros visoms internetinio anonimiškumo formoms.

P2P tinklo vartotojai, siekia anonimiškumo, nenorėdami būti atpažinti kaip informacijos siuntėjai ar gavėjai. Labiausiai paplitusios priežastys:

- Medžiaga ar jos dalijimasis yra neteisėti.
- Medžiaga yra legali, tačiau visuotinai laikoma smerktina, gėdinga ar problematiška (pvz.: anonimiškumas yra pagrindinis faktorius kovos su priklausomybių ligomis organizacijoms).
- Bausmės baimė (pvz.: informatoriai, neoficialus informacijos nutekėjimas ir t.t.).
- Vietinė, organizacinė arba nacionalinė cenzūra.

- Asmens privatumo nuostatos (pvz.: užkirsti kelią stebėti asmenį ir rinkti duomenis apie jį).

Anonimiškų P2P tinklų veikimas [11] [12] [13]

Kai kurie tinklai paprastai vadinami „anonimiškais P2P tinklais" yra iš tikrųjų anonimiški ta prasme, kad tinklo mazgas neturi jokių identifikatorių. Likę tinklai yra pseudonimiški: vartotojai ar mazgai identifikuojami pagal tokius pseudonimus kaip kriptografiniai raktai. Pavyzdžiui kiekvienas *Mute* tinklo mazgas turi perdangos adresą, kuris yra kilęs iš jo viešojo rakto. Toks perdangos adresas funkcionuoja kaip mazgo pseudonimas, leidžiantis tą mazgą pasiekti įvairioms žinutėms. Freenet tinkle, pranešimai yra nukreipiami naudojant raktus, kurie identifikuoja tam tikras siunčiamų duomenų dalis, o ne mazgus; mazgai lieka anonimiški.

Terminas „anonimiškas" naudojamas apibūdinti anonimiškiems ir pseudonimiškiems tinklų tipams, nes yra sunku arba neįmanoma nustatyti, ar mazgas siunčia pranešimą ar tiesiog jį persiunčia kito tinklo mazgo vardu. Kiekvienas anoniminis anonimiško taškas į taškas tinklo mazgas veikia kaip universalus siuntėjas ir universalus gavėjas siekiant išlaikyti anonimiškumą. Jei mazgas buvo tik gavėjas ir nieko nesiuntė, kaimyniniai mazgai žinos, jog informacija, kurios jis prašė, buvo skirta tik jam pačiam, panaikindami bet kokias abejones, jog jis buvo tikrasis informacijos gavėjas. Todėl P2P tinkle mazgai norėdami išlikti anonimiškais, visą laiką turi persiųsti informaciją kitiems mazgams tinkle.

Iš pradžių anonimiškumą užtikrinantys tinklai buvo naudojami nedidelių ir draugiškų tų tinklų kūrėjų bendruomenių. Populiarėjant anonimiškiems P2P tinklams, augo ir jų vartotojų skaičius, todėl neišvengiamai atsirado piktybiškų vartotojų, kurie bandė įvykdyti įvairias atakas. Tai yra panašu į internetą, kur didelę tarptautinio tinklo plėtrą sekė didžiulė brukalų siuntinėjimo ir DDoS (angl. *Distributed Denial of Service*) atakų banga. Tokių atakų vykdymas anonimiškuose tinkluose reikalauja visiškai kitokių sprendimų. Pavyzdžiui, tokių atakų anoniminiame P2P tinkle kaltininko adreso įtraukimas į juodąjį sąrašą neduos jokių rezultatų, nes anoniminiai tinklais tokią informaciją slepia. Šie tinklai yra daug labiau pažeidžiami DDoS atakų taip pat ir dėl mažesnių tinklo pralaidumų.

Sąmokslas atakuoti anoniminį tinklą galėtų būti laikomas baudžiama nusikalstama veikla, tačiau dėl tinklo pobūdžio to neįmanoma padaryti nepakenkiat duomenų anonimiškumui tinkle.

Anonimiškumo P2P tinkle užtikrinimo mechanizmas

Pati anonimiškų tinklų pradžia buvo 1981m., kai Chaum pristatė *Mix-net* anonimiškų tinklų koncepciją [14], kur anonimiškumas pasiekiamas naudojant *derinius* (angl. *mixes*, panašiai kaip tarpiniai tunelio mazgai P2P tinkle). Derinys yra patobulintas, praplėstas įgaliojimas (*angl.* proxy),

kuris padeda paslėpti siuntėją nuo gavėjo, taip pat suteikia siuntėjo ir gavėjo nesusiejamumą prieš sekimą.

Tokia sistema persiunčia žinutes per mazgų grandinę, panaudojant sluoksninį šifravimą, todėl kiekvienas mazgas, siunčiant pranešimą, gali žinoti tik pirmtaką, iš kurio gavo žinutę, ir tęsėją, kuriam persiuntė žinutę. Pagrindinis principas, kuris naudojamas taikant šį metodą žinučių persiuntimui tinkle, kiekvienas anoniminės grandinės mazgas laukia kol pas jį susikaupia nemaža grupė persiunčiamų žinučių, tada jos sumaišomos ir persiunčiamos tolimesniam mazgui.

Tiksliau, kiekvienas derinys surenka grupę šifruotų žinučių iš įvairių siuntėjų, jas iššifruoja, sudėlioja partijomis, pertvarko eilės tvarką, pašalina iš jų (jeigu randa) siuntėjo vardą ir identifikavimo informaciją ir persiunčia sekančiam mazgui. Toks procesas atakuotojui pasunkina įeinančių į mazgą žinučių susiejimą su atitinkamomis išeinančiomis žinutėmis. Pavyzdžiui kiekvienas mazgas *P*, kuris nori užmegzti ryšį su mazgu *Q*, nusiųsdamas jam žinutę *m*, turi užšifruoti šią žinutę *Q* mazgo viešuoju raktu, kad gautų *m'*. Tada mazgas *P* užšifruos porą (*m', q*), panaudodamas derinio viešąjį raktą. Derinys, gavęs žinutę, ją iššifruos panaudojęs savo privatų raktą ir persiųs ją gavėjui *Q*.

Pagrindinis tokios sistemos trūkumas: kiekvienas mazgas grandinėje turi sulaukti, kol gaus reikiamą kiekį žinučių, kad galėtų vykdyti sumaišymo procedūras, ir tik tada persiųsti kitam mazgui. Tokio tinklo veikimas būtų labai lėtas.

Opennet [15] ir darknet [16] tinklų tipai

Kaip ir įprastuose P2P tinkluose, anoniminiuose P2P tinkluose gali būti įgyvendinti opennet ir darknet (dar vadinamas draugas – draugui) tinklų tipai. Tai paaiškina kaip tinklo mazgas užmezga ryšį tik su sau lygiaverčiu mazgu:

- *Opennet* tipo tinkle, mazgai yra randami automatiškai. Nereikia jokio konfigūravimo, tačiau trūksta kontrolės, kurie mazgai tampa lygiaverčiai.
- *Darknet* tipo tinklo vartotojai rankiniu būdu nustato ryšius tarp mazgų, kurie priklauso žmonėms, kuriuos jie pažįsta. *Darknet* tinklo kūrimui paprastai reikia daugiau pastangų, tačiau mazgas užmezga ryšius tik su kitais patikimais ir žinomais mazgais ar taškais.

Kai kurie tinklai, kaip kad *Freenet*, palaiko abu tinklų tipus tuo pat metu (mazgas gali turėti dalį rankiniu būdu pridėtų *darknet* lygiaverčių mazgų ir dalį automatiškai parinktų opennet taškų).

Draugas – draugui (arba *F2F*) tinkle, vartotojai gali sukurti tik tiesioginį ryšį su žmonėmis, kuriuos žino. Daugelis *F2F* tinklų palaiko netiesioginius anoniminius arba pseudoniminius ryšius tarp vartotojų, kurie vienas kito nežino arba nepasitiki vienas kitu. Pavyzdžiui, draugas – draugui perdangos mazgas gali automatiškai perduoti failą (arba failo prašymą) anonimiškai tarp dviejų

„draugų", nesakydamas nė vienam iš jų kito vartotojo vardo arba IP adreso. Šie „draugai" gali persiųsti tą patį failą (ar to paties failo prašymą) savo draugams ir taip toliau. *F2F* tinkle vartotojai nežino kas dar dalyvauja už jų draugų rato, taigi F2F tinklai gali plėstis nesukompromituodami savo vartotojų anonimiškumo.

Failų apsikeitimo įrankių palyginimas

Iš *1.1 lentelės* galima susidaryti vaizdą apie šiuo metu rinkoje egzistuojančius anonimiškumą garantuojančius failų apsikeitimo įrankius. Visos programos sukurtos naudojant įvairias programavimo kalbas, taip pat yra suderinamos su visomis populiariausiomis platformomis.

2.1 lentelė. Anonimišką failų apsikeitimą garantuojantys įrankiai ir tinklai.

Pavadinimas	Tinklas	P2P anonimiškumas	Kaina	Platforma	Licenzija	Programavimo kalba
ANts P2P	ANts	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Java
Freenet's FProxy	Freenet	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Java
Frost	Freenet	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Java
GNUnet	GNUnet	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	C
I2Phex	I2P (protocol gnutella)	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Java
I2PSnark	I2P (protocol BitTorrent)	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Java
iMule	I2P (protocol kad)	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	C++
Calypso	MUTE	Yra	Aukojimas	Linux, Windows	GPL	C++ and QT
MUTE	MUTE	Yra	Aukojimas	Linux, Mac OS X, Windows	GPL	C++
Nodezilla	Private	Yra	Nemokamas	Linux, Windows	partially GPL	Java
Retroshare	Retroshare	Yra	Nemokamas	Windows, Linux, MacOS	GPL	C++
RShare	RShare	Yra	Nemokamas	Windows	GPL	C#
Robert	BitTorrent	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Python
Share	Share	Yra	Nemokamas	Windows	Proprietary	Delphi
StealthNet	RShare	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	C#
Thaw	Freenet	Yra	Nemokamas	Linux, Mac OS X, Windows	GPL	Java
Winny	Winny	Yra	Nemokamas	Windows	Proprietary	C++
Datawire	P2P Private Networking	Yra	Nemokamas	Windows, Linux, Mac OS X	Proprietary	Java

2.3. Anoniminių P2P tinklų pavyzdžiai

2.3.1. MUTE P2P tinklo anonimiškumo užtikrinimo metodas [17].

Pagrindinis būdas, kurio dėka MUTE apsaugo vartotojų privatumą yra tiesioginio sujungimo vengimas tarp informacijos siuntėjų ir gavėjų. Standartiniame P2P tinkle vartotojas padaro paiešką, kuri yra siunčiama jo kaimynams, kurie persiunčia savo kaimynams ir taip toliau. Naudojant tinklą paieškos užklausos persiuntimui, ši konkreti užklausa nukeliauja į daugelį tinklo mazgų neužmegzdama tiesioginio kontakto. Kai ateina laikas siųsti failą, tiesioginis ryšys yra užmezgamas.

MUTE nukreipia visas žinutes, įskaitant paieškos užklausas, paieškos rezultatus, ir failus per kaimyninius mazgus. Taigi, nors žinome kaimyninių tinklo taškų interneto adresus, mes nežinome mazgo, iš kurio siunčiame informaciją, adreso.

MUTE tinklo topologija yra identiška standartinio tinklo topologijai. Jei atliksime „muzikos_failas“ failo paiešką tinkle, mūsų rezultatai atrodys taip:

2.2 lentelė. MUTE failo paieškos rezultatų pavyzdys.

Mano adresas: **7213D...2DCA5**	Mano Failas: **Muzikos_failas_1.mp3**
Mano adresas: **7213D...2DCA5**	Mano Failas: **Muzikos_failas_2.mp3**
Mano adresas: **7213D...2DCA5**	Mano Failas: **Muzikos_failas_3.mp3**

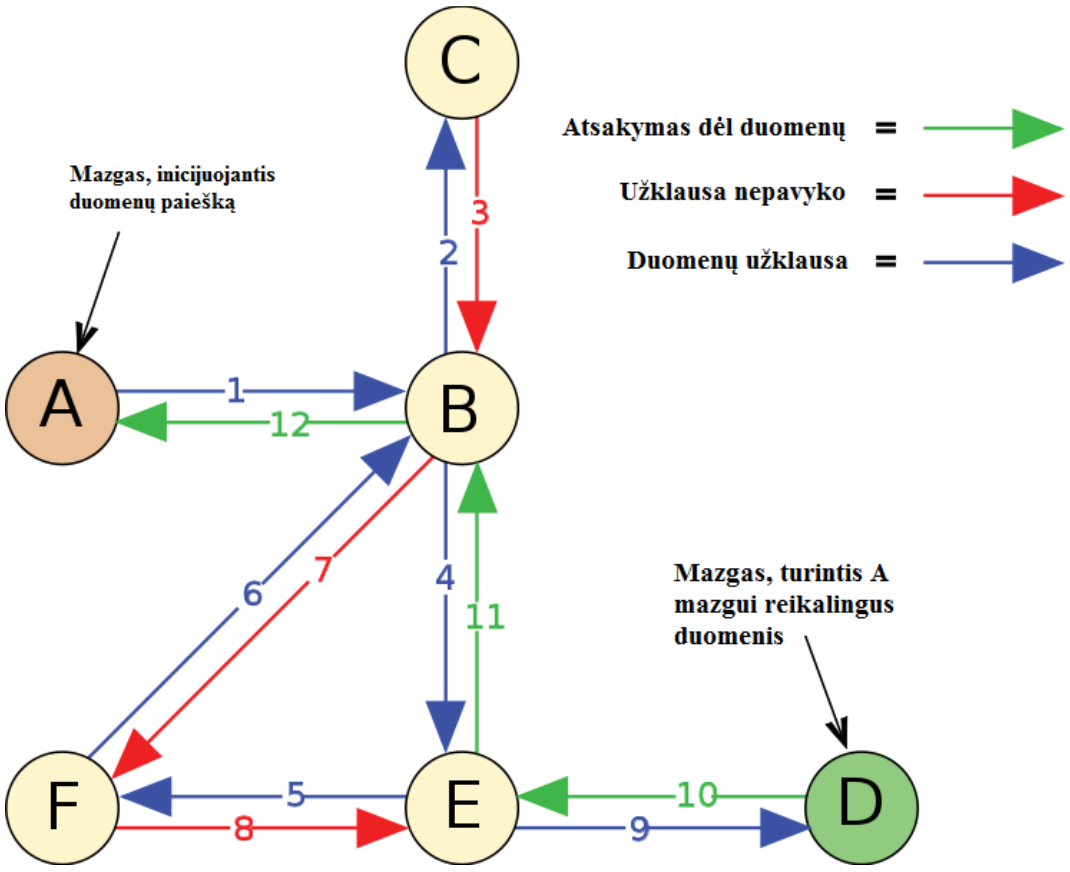
„*Mano adresas*“ skiltyje rodomas adresas yra sutrumpintas kad tilptų į lentelę, pilnas adresas: 7213D29781593840CF00CDD1E9A7A425AE16DCA5. Tai yra MUTE virtualus adresas. Kiekvienas MUTE tinklo mazgas turi virtualų adresą, kuris kiekvieną kartą atsitiktinai sugeneruojamas mazgui iš naujo prisijungus į tinklą. Kaimynai tinkle (mazgai, kurie iš tikrųjų žino mūsų internetinį adresą) nežino koks yra mūsų virtualus adresas, todėl niekas negali susieti fizinio internetinio adreso su virtualiu MUTE adresu, ir todėl niekas neatskleis tinklo vartotojo asmenybės.

2.3.2. Freenet P2P tinklo anonimiškumo užtikrinimo metodas [18].

Freenet anoniminiame tinkle informacija „plaukioja” iš vienos vietos į kitą, kur jinai yra aktualiausia tuo metu. Ši architektūra decentralizuota ir pagrįsta nestruktūrizuota netiesioginio susijungimo architektūra.

Freenet architektūra visiškai užtikrina anonimiškumą. Informacijos gavimas, kol ji nėra paplitusi po tinklą, yra lėtas, tačiau populiari informacija yra greitai pasiekiama. Tinkle nėra saugoma nenaudojama informacija. Kuo mazgas ilgiau gyvuoja tinkle, tuo daugiau jis žino apie tinklą ir gali sėkmingiau vykdyti informacijos saugojimą ir paėmimą. Failams tinkle yra suteikiami raktiniai vardai, kurių prasmė nieko nepasako apie failo kilmę. Kiekvienas mazgas tinkle yra lygiavertis ir turi nuosavą lentelę, kurioje saugo informaciją apie kitus mazgus – tai yra, kaip juos pasiekti ir kokius dokumentus jie turi. Taip pat kiekvienas mazgas leidžia savo disko vieta dalintis su kitais mazgais. Užklausos siunčiamos iš vieno mazgo į kitą, nes nėra centrinio mazgo, ir taip grandine vyksta paieška. Kiekvienas mazgas pats nusprendžia, kuris kitas mazgas galėtų turėti prašomą informaciją.

Kiekviena užklausa turi gyvavimo laiko skaičių (*GL*, kuris yra mažinamas, perduodant užklausą į tolimesnį mazgą. Taip pat kiekviena užklausa turi identifikatorių, kuris yra saugomas identifikatorių lentelėje. Kad nesusidarytų amžinų užklausos ciklų, žinutės yra neapdorojamos, jei jų identifikatorių jau turi lentelė arba *GL* skaičius lygus nuliui.



2.10 pav. Freenet tinklo veikimo schema. [18]

2.3.3. MorphMix P2P tinklas [19]

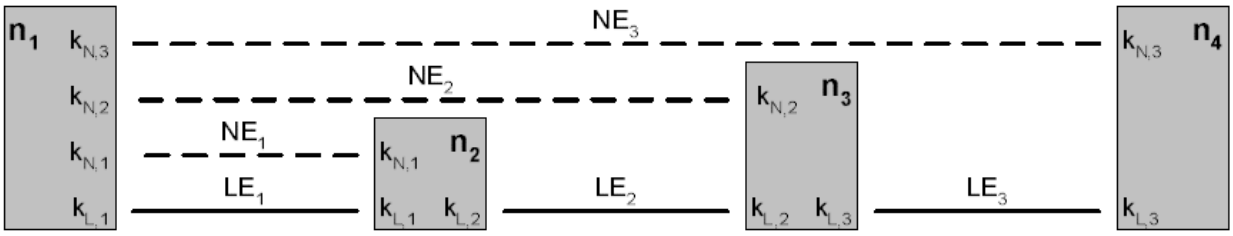
MorphMix yra *mix* technologija paremtas P2P tinklas, kuris turi neribotą kiekį tinklo mazgų. Tinklo mazgas *i* atpažįstamas pagal jo IP adresą ip_i. Taip pat, kiekvienas tinklo mazgas turi

kriptografinių raktų porą: viešą raktą PuK_i ir privatų raktą PrK_i. Ši raktų pora sugeneruojama, kada mazgas pirmą kartą prisijungia prie MorphMix tinklo.

Jei vartotojas nori naršyti internete anonimiškai, jam reikia sukurti anoniminį tunelį, kuris prasideda jam priklausančiam tinklo mazge ir eina per kitus P2P tinklo mazgus. Toks mazgas vadinamas *iniciavimo mazgu* arba tiesiog *iniciatoriumi*. Paskutinis anoniminio tunelio mazgas vadinamas *galutiniu mazgu*, o tinklo mazgai, kurie tunelyje yra tarp iniciatoriaus ir galutinio mazgo vadinami *perdavimo mazgais*. Anoniminio P2P tinklo mazgai skirstomi į dvi grupes: *neutralūs mazgai*, kurie neturi ketinimų pakenkti kitų mazgų anonimiškumui ir *atakuojantys mazgai*, kurie bando atskleisti iniciavimo mazgų tapatybę. Atakuojantys mazgai bendradarbiauja tarpusavyje, padidindami vykdomų atakų tinkle sėkmės tikimybę.

Visos žinutės, kuriomis apsikeičia du anoniminio P2P tinklo mazgai (žymimi n_i), yra vienodo ilgio; $\{m\}_k$ žymima žinutė *m* užšifruota šifro raktu *k*. Kai mazgas n_1 anoniminiu tuneliu siunčia žinutę m, ji yra kelis kartus užšifruojama simetriniais šifro raktais panaudojant *įterptinį šifravimą* (NE) ir gaunama: $\{\{\{m\}k_{N,3}\}k_{N,2}\}k_{N,1}$. Šio šifro bloko pradžioje pridedama antraštė, kurioje įrašomas identifikatorius, pagal kurį žinutę yra persiunčiama tarp anoniminio tinklo gretimų mazgų.

Prieš pat tinklo mazgui n_1 išsiunčiant žinutę mazgui n_2, žinutės antraštė užšifruojama simetriniu šifro raktu $k_{L,1}$ panaudojus *sujungimo šifravimą* (LE). Mazgas n_2, gavęs žinutę, iššifruoja antraštę panaudojęs raktą $k_{L,1}$ ir iššifruoja vieną žinutės turinio šifro lygį raktu $k_{N,1}$. Toliau n_2 pagal žinutės antaštėje esantį identifikatorių nustato sekantį anoniminio tunelio mazgą, kuriam reikia persiųsti žinutę m, pakoreaguoja žinutės atraštę sujungimui su sekančiu mazgu ir užšifruoja antraštę raktu $k_{L,2}$ bei persiunčia žinutę mazgui n_3. Šis procesas kartojasi, kol pasiekiamas galutinis anoniminio tunelio mazgas, kuris persiunčia informaciją n_1 siųstos informacijos gavėjui. Visas žinutės perdavimo-šifravimo procesas pavaizduojamas 2.11 pav.



2.11 pav. Žinutės šifravimo lygiai. [19]

Persiuntimo mazgo parinkimas

MorphMix anoniminiame tinkle iniciatorius parenka tik pirmą persiuntimo mazgą, tada kiekvienas kitas anoniminio tunelio mazgas parenka sekantį mazgą. Toks procesas turi vieną didelį privalumą – kiekvienas tinklo mazgas turi žinoti tik kelis kitus tinklo mazgus. Jie gali komunikuoti tarpusavyje ir apsikeisti informacija, kad žinotų, kuris mazgas turi atliekamų resursų priimti naują

anoniminį tunelį. Jeigu iniciatorius turėtų parinkti visus tunelio mazgus pats, jis nežinotų dabartinės tinklo mazgų būklės (išskyrus mazgą n_1), pvz. nežinotų ar mazgai turi atliekamų resursų anoniminiam tuneliui užmegzti. Kokybiškam tokios sistemos funkcionavimui būtų reikalingas efektyvus mazgų paieškos protokolas, į kurį iniciatorius galėtų kreiptis dėl laisvų anoniminio tinklo mazgų paieškos. Egzistuoja kintami P2P tinklo mazgų paieškos protokolai pvz. Chord [20] tačiau jie generuoja labai didelį informacijos srautą, kuris didina viso tinklo apkrovą, kadangi mazgų, prisijungusių prie P2P tinklo kaita yra labai didžiulė, reikia pastoviai daryti visų mazgų būsenų indeksaciją, kad bet kuriuo metu būtų greitai gaunama naujausia informacija apie norimą tinklo mazgą. Leidžiant kiekvienam mazgui pasirinkti sekantį anoniminio tunelio mazgą, MorphMix tinklui suteikiamas didesnis dinamiškumas, mazgai turi stebėti tik savo aplinką, o ne visą tinklą. Nepriklausomai nuo sistemos dydžio, tinklo mazgai bet kuriuo metu komunikuoja tik su nedideliu kiekiu kitų tinklo mazgų.

Tačiau toks tunelio formavimas turi saugumo trūkumą – jei į anoniminį tunelį patenka atakuojantis mazgas, jis gali pats susimuliuoti likusią anoniminio tunelio mazgų grandinę arba nukreipti tunelį į kitą atakuojantį mazgą.

Mazgų aplinka ir persiuntimo mazgų radimas

Bet kuriuo laiko momentu, bet kuris anoniminio P2P tinklo mazgas žino tam tikrą informaciją apie kai kuriuos kitus tinklo mazgus: jų IP adresus ir viešuosius raktus. Du mazgai yra susijungę tik tuo atveju, jeigu tuo metu tarp jų yra nustatytas sujungimo šifravimas (LE). Du tarpusavyje susijungę tinklo mazgai apsikeičia kontroline informacija, kuri pasako ar mazgas gali tapti anoniminio tunelio persiuntimo mazgu. Mazgai taip pat gali sužinoti užmegzto ryšio kokybę, nusiųsdami *ping* žinutę, nes nėra tikslo į anoniminį tunelį įtraukti mazgą, kuris negali užtikrinti patikimo informacijos perdavimo anoniminiu tuneliu. Taigi, MorphMix tinkle bet kuris tinklo mazgas bet kuriuo momentu yra užmezgęs ryšį su tam tikru skaičiumi kitų tinklo mazgų ir bent keli iš tų mazgų bet kuriuo laiko momentu gali sutikti tapti anoniminio tunelio persiuntimo mazgu.

Yra skirtingų būdų mazgui gauti informaciją apie kitą tinklo mazgą: kad mazgas užmegztų ryšį, jis turi žinoti bent vieną tuo metu aktyvų tinklo mazgą. Iš pradžių tikrinami anksčiau aktyviais buvę tinklo mazgai, vėliau kreipiamasi į tinklo informacinius serverius ir prašoma, kad jie nurodytų bent keletą aktyvių mazgų. Šie serveriai žino tam tikrą kiekį aktyvių tinklo mazgų, tačiau jiems visiškai nesvarbu, kurią dalį visų tinklo mazgų jie sudaro. Kiekvienas tinklo mazgas laikas nuo laiko praneša serveriui apie kitus aktyvius tinklo mazgus aplink jį, už tai jis iš serverio gauna informacija apie keletą kitų aktyvių tinklo mazgų. Serveriuose pastoviai trinama mazgų informacija, apie kuriuos kurį laiką jis negauna informacijos iš kitų tinklo mazgų. Gavęs užklausą, serveris praneša apie kelis atsitiktinai parinktus aktyvius tinklo mazgus iš tuo metu turimos aktyvių mazgų lentelės.

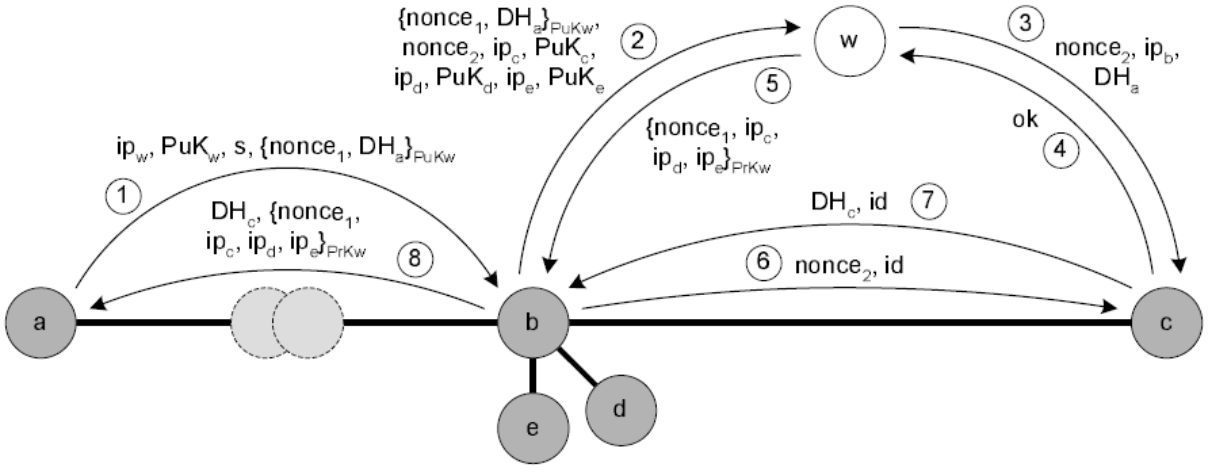
Anoniminiam tinklui labai svarbu, kad būtų kreipiamasi į skirtingus šaltinius, dėl informacijos apie aktyvius tinklo mazgus. Jei ką tik prisijungęs naujas tinklo mazgas kreiptųsi tik į vieną šaltinį, o tas šaltinis priklausytų atakuojančių mazgų grupei, naujas tinko mazgas žinotų ir siūlytų kitiems mazgas tik atakuojančius mazgus, taip sukeldamas grėsmę kitų mazgų anonimiškumui.

Įterptinio ir sujungimo šifravimo nustatymas

Tinklo mazgui *a* norint nustatyti sujungimo šifravimą su kitu tinklo mazgu *b*, iš pradžių reikia sukurti TCP ryšį su tuo mazgu. Mazgas *a* tada parenka atsitiktinę bitų eilutę, kuri naudojama kaip simetrinis raktas sujungimo šifravimui. Šis raktas užšifruojamas *b* mazgo viešuoju raktu ir nusiunčiamas *b* mazgui.

Įterptinio šifravimo nustatymas vyksta tarp iniciatoriaus ir sekančio mazgo anoniminio tunelio viduje. Šifravimo nustatymo tikslas yra sukurti simetrinį šifro raktą, kuris būtų žinomas tik anoniminio tunelio siuntėjau ir gavėjui. Kadangi iniciatorius nežino kitų anoniminio tunelio mazgų ir jų viešųjų šifro raktų (išskyrus pirmąjį tunelio mazgą), naudojamas Diffie-Hellman (DH) [21] raktų apsikeitimo metodas. Jei iniciatorius tiesiog nusiųstu savo DH rakto dalį sekančiam mazgui *b*, kad jis persiųstų jį mazgui *c*, mazgas *b* nesunkiai gali apsimesti likusiais anoniminio tunelio mazgais ir iniciatorius to nė nepastebėtų. Tam, kad tai neįvyktų, naudojama „*liudininkų*“ sąvoka – iniciatorius kiekvienam „šuoliui“ tarp mazgų anoniminiame tunelyje atsitiktinai parenka mazgus-liudininkus (iš savo žinomų aktyvių mazgų sąrašo). Mazgų-liudininkų užduotis – dalyvauti sekančio anoniminio tunelio mazgo parinkimo procese (2.12 pav.).

Mazgas *a* yra iniciatorius. Tarkime anoniminis tunelis yra sukurtas iki mazgo *b* (per nulį ar daugiau tarpinių mazgų, o mazgas *b* turi užmezgęs ryšį (ne anoniminį tunelį) su mazgais *c*, *d*, *e*, kurie gali pratęsti anoniminį tunelį.



2.12 pav. Įterptinio šifravimo kūrimas. [19][21]

Norint įgyvendinti įterptinį šifravimą tarp dviejų mazgų, reikia įvykdyti šiuos punktus:

1) Mazgas *a* atsitiktinai parenka liudininką *w* iš žinomų mazgų rinkinio. Mazgas *a* sugeneruoja savo DH rakto dalį (GH_a) ir $nonce_1$, siekiant apsisaugoti nuo pakartojimo atakų, kurie yra užšifruojami mazgo *w* viešuoju šifro raktu PuK_w. Gaunamas šifruotas blokas $\{nonce_1, DH_a\}_{PuKw}$. Mazgas *a* taip pat nurodo *s* – tai yra mazgų skaičius, kurį mazgas *b* turi nurodyti 2-oje žinutėje. Tarkime *s* = 3. Mazgas *a* išsiunčia žinutę mazgui *b*, kurioje yra mazgo *z* IP adresas ip_w, PuK_w, *s*, ir užšifruoti nonce ir DH parametrai. 2 – tra žinutė praneša mazgui *b*, kad reikia pridėti prie anoniminio tunelio sekantį mazgą, panaudojant liudininką *c*.
2) Mazgas *b* sukuria sujungimo šifravimą su mazgu *w*, panaudojęs ip_w, ir PuK_w. Taip pat jis sugeneruoja $nonce_2$, kuris naudojamas 6-tai žinutei pripažinti. Mazgas *b* sugeneruoja žinutę, kurioje yra užšifruoti nonce ir DH parametrai iš mazgo *a*, $nonce_2$, IP adresai trijų potencialių tunelio mazgų (ip_c, ip_d, ip_e) ir jų viešieji raktai (PuK_c, PuK_d, PuK_e). Ši žinutė nusiunčiama mazgui *w*.
3) Mazgas *w* gauna žinutę ir atsitiktinai parenka sekantį tunelio mazgo iš siūlomų. Nagrinėjamame pavyzdyje (2.12 pav) mazgas *w* parenka mazgą *c* kaip sekantį tunelio mazgą ir sukuria sujungimo šifravimą su mazgu *c* panaudodamas PuK_c. Mazgas *w* iššifruoja $nonce_1$ ir DH_a, panaudojęs savo privatų raktą PrK_w, sugeneruoja žinutę, kurioje patalpina $nonce_2$, ip_b, DH_a ir nusiunčia ją mazgui *c*.
4) Mazgas *c* gauna žinutę ir pasitikrina ar jis iš tikrųjų pasiruošęs tęsti anoniminį tunelį. Jei viskas gerai, mazgui *w* nusiunčiama patvirtinimo žinutė.
5) Mazgas *w* gauna patvirtinimo žinutę ir sugeneruoja „kvitą“ mazgui *a*, kuriame yra IP adresai mazgų, kuriuos pasiūlė *b* mazgas, pasirašytas mazgo *w* panaudojus raktą PrK_w. Pirmas IP adresas kvite yra tas, kurį parinko mazgas *w*, kaip sekantį tunelio mazgą. Kvite taip pat patalpintas $nonce_1$ parametras ir viskas siunčiamas mazgui *b*.
6) Mazgas *b* gauna žinutę iš mazgo *w* ir sužino, kad pasirinktas mazgas *c*. Mzagas *b* sugeneruoja naują žinutę, į kurią įdeda $nonce_2$ ir identifikatorių id, kad identifikuoti duomenis, priklausančius anoniminio tunelio daliai tarp mazgų *b* ir *c* ir nusiunčia ją mazgui *c*. Mazgo *w* paslaugų daugiau nebereikia, todėl ryšys su juo nutraukiamas.
7) Mazgas *c* gauna žinutę ir nusiunčia savo dalį DH apsikeitimo rakto (DH_c) atgal mazgui žinutėje, pažymėtoje id identifikatoriumi.

8) Mazgas *b* sugeneruoja žinutę, į kurią įdedama DH_c ir „kvitas“, gautas iš mazgo *w*, kuri nusiunčiama mazgui *a*.

Jei kuris nors etapas nepavyksta, mazgas *a* gauna klaidos žinutę ir tada jis turi nuspręsti, ar sunaikinti tunelį, ar bandyti iš naujo. Reikia pažymėti, kad tokia pat procedūra, naudojama pridėti pirmą tunelio mazgą po iniciatoriaus mazgo. Aišku mazgas *a* galėtų iškart pats nustatyti įterptinį šifravimą, tačiau taip išsiduotų sekančiam tunelio mazgui, kad jis yra iniciatorius.

Jeigu mazgas *b* norėtų susimuliuoti anoniminio tinklo šuolį į mazgą *c*, jis galėtų mazgui *w* 2-oje žinutėje nusiųsti netikrus viešuosius raktus, kurių privačius raktus jis žino, perimti 3-ią žinutę ir elgtis lyg būtų mazgas *c*. Kad tai padarytų, jam reikia aktyvios ryšio tarp mazgų *w* ir *c* kontrolės, norint perimti ir įterpti duomenų paketus, tačiau jis negali nuspėti, kuris mazgas bus pasirinktas kaip liudininkas. To pasekoje, jis negali iš anksto pasiruošti paketų perėmimui, o atlikti tokią ataką šalia mazgo *w* iš anksto nepasiruošus yra labai sunku. Realesnis variantas, kad mazgas *b* perims paketus šalia mazgo *c*, tuo labiau, kad mazgas *b* ir parenka mazgų sąrašą, kurį siunčia 2-troje žinutėje. Siekiant kuo labiau sumažinti tokios atakos tikimybę, yra reikalaujama, kad mazgų, kuriuos siūlo *b* mazgas, IP adresai ir pačio *b* mazgo IP adresas neturėtų to paties prefikso.

Jei *b* ir *w* mazgai priklauso tai pačiai atakuojančių mazgų grupei, mazgui *b* yra lengvi susimuliuoti šuolį iš tarp mazgų *b* ir *c*, nes mazgas *w* gali atskleisti DH_a. Tačiau mazgas *a* visus liudininkus parenka visiškai atsitiktinai, tikimybė, kad visi mazgai-liudininkai priklausytų tai pačiai grupei kaip ir mazgas *b*, yra labai maža, turint omeny, kad sąlyginai maža visų tinklo mazgų dalis yra piktavaliai atakuojantys mazgai. Mazgui *b* tampa dar sunkiau susimuliuoti tunelio šuolį, kai ateina etapas, kada liudininkas baigia savo darbą.

Nagrinėjant variantą, kad mazgas *b* priklauso didėliai atakuojančių mazgų grupei, jis tiesiog gali liudininkui 2-oje žinutėje pasiūlyti mazgus, iš tos „piktavalių“ grupės ir taip būtų garantuota, kad kitas tunelio mazgas bus iš tos mazgų grupės. Tačiau yra tikrinama, kad mazgai neturėtų tuo paties IP adreso prefikso, atakuojantys mazgai turėtų būti iš skirtingų potinklių, todėl tokio pobūdžio ataka yra komplikuojama. Tačiau jeigu vis dėlto atakuotojui pavyktų sukaupti mazgus, esančius skirtingose Interneto vietose, tada tokią ataką turėtų būti ganėtinai lengva įgyvendinti.

Galima padaryti išvadą, kad pati realiausia ataka – kada grupė atakuojančių mazgų bando kontroliuoti kuo didesnį skaičių anoniminio tunelio perdavimo mazgų, siūlydami kuo daugiau atakuojančių mazgų liudininkams pasirinkti. Visos kitos atakos reikalauja aktyvios kelių tinklo sąsajų kontrolės vienu metu, todėl jas daug sunkiau įgyvendinti, arba reikia labai didelių resursų.

Kadangi mazgai-liudininkai žino savo kaimyninius mazgus, iniciatorius turėtų parinkti liudininkus iš to mazgų rinkinio, kurį žino tačiau niekada dar nebuvo užmezgęs tiesioginio ryšio.

2.3.4. Tarzan P2P tinklas [22]

Tarzan yra decentralizuotas anoniminis P2P tinklas, lengvai plečiamas ir patogiai valdomas. Skirtingai nei MorphMix tinkle, Tarzan anoniminiame P2P tinkle anoniminio tunelio iniciatorius pats parenka visus tunelio mazgus. Tarzan tinklas užtikrina tam tikrą anonimiškumo lygį tiek paprastiems klientas, tiek serveriams. Abiem atvejais yra naudojamas tinklo adresų vertėjas (NAT).

Tarzan tinklo veikimas remiasi tais pačiais principais kaip ir MorphMix: kiekvieno šuolio tarp anoniminio tunelio metu, prie siunčiamos žinutės yra pridedamas arba atimamas šifravimo sluoksnis (priklauso nuo siuntimo krypties), žinutės antraštė saugo informaciją apie anoniminio tunelio kelią.

Kuriant naują anoniminį tunelį Tarzan tinkle, iniciatoriaus mazgas atsitiktinai parenka mazgų grandinę remdamasis savo vietine topologija. Iniciatorius atsakingas už anoniminio tunelio sukūrimą, prijungiant po vieną mazgą vienu metu, o ne visus iškart. Šis procesas apima simetrinių šifro raktų generavimą ir pristatymą kiekvienam tunelio mazgui atskirai, užšifruojant juos viešu to mazgo šifro raktu. Kiekvienas Tarzan tinklo mazgas, naujai prisijungęs į tinklą, susigeneruoja savo privatų šifro raktą.

Mazgas n_0 norėdamas sukurti anoniminį tunelį, pasiunčia užklausą sekančiam tunelio mazgui n_i per mazgą n_1. Užklausa persiunčiama paprastame duomenų pakete. Šią užklausą n_i, gauna ne tiesiogiai iš iniciatoriaus, o per kitą tinklo mazgą n_{i-1}, todėl jis negali tiksliai žinoti iš kurio tinklo mazgo atėjo užklausa. Mazgas n_{i-1} negali atskirti sėkmingų tunelio sudarymo užklausos atsakymų iš paprasto duomenų srauto.

Mazgas n_0 sudaro užklausą užmegzti anoniminį tunelį, panaudodamas mazgo n_i viešąjį raktą. Į šią užklausą taip pat įeina pradinės išankstinės sesijos raktas. Mazgas n_i, gavęs užklausą iš mazgo n_{i-1}, panaudodamas savo privatų šifro raktą, ją išsišifruoja. Sesijos rakte yra užšifruojama: persiuntimo vientisumo raktas, tolesnis anoniminio tunelio atgalinis raktas iššifruoti paketams, gautiems iš mazgo n_{i+1}, anoniminio tunelio mazgų adresai (n_{i-1}, n_{i+1}) ir persiuntimo identifikatoriai, kurie naudojami duomenų paketų žymėjimui, vaikščiojančių abejomis anoniminio tunelio kryptimis.

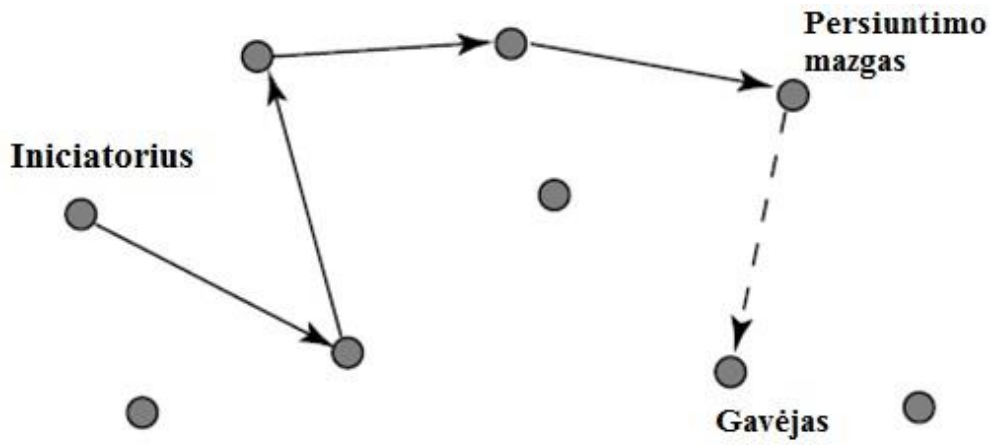
Mazgas h_i, sėkmingai išsaugojęs užklausos būseną, praneša tunelio pradiniam mazgui, kad yra pasiruošęs anoniminio tunelio korektiškumo patikrai.

2.3.5. AP3 P2P tinklas [23]

AP3 anoniminis P2P tinklas vartotojams suteikia kooperatyvią, paskirstytą anoniminio bendravimo paslaugą. AP3 tinklas yra visiškai decentralizuotas ir pats save palaikantis. AP3 tinklas savo vartotojams teikia šias anonimiškos komunikacijos paslaugas: anonimišką žinučių perdavimą, anoniminio ryšio kanalus (tunelius) ir saugius pseudonimus, kurie suteikia galimybę anoniminiame tinkle įgauti tam tikrą reputaciją, neatskleidžiant savo tikrosios tapatybės.

Anoniminio žinučių perdavimo AP3 tinkle strategija yra panaši į tą, kuri naudojama Tarzan anoniminiame P2P tinkle. AP3 tinkle mazgas nežino, iš kurio ar kitas tinklo mazgas, atsiuntęs užklausos žinutę, yra tos žinutės autorius ar tik persiuntėjas. Taigi žinutės gavėjas sužino tik tą mazgą, kuris persiuntė žinutę.

Kai tinklo mazgas ketina anonimiškai persiųsti žinutę, iš pradžių jis turi sukurti anoniminės užklausos objektą, sudarytą iš pačios žinutės ir gavėjo adreso. Žinutėje negali būti jokios informacijos, galinčios atskleisti jos autoriaus tapatybę. Ši žinutė tada persiunčiama į atsitiktinai parinktą tinklo mazgą. Gavęs užklausą, AP3 tinklo mazgas, pasitelkęs „monetos metimo“ principą, nusprendžia, ar pagal gautą užklausą nusiųsti žinutę galutiniam gavėjui, ar persiųsti ją kitam atsitiktinai parinktam tinklo mazgui. Sprendimo tikimybė dėl žinutės persiuntimo yra p_f. Toks mechanizmas reikalingas, kad būtų sukurtas atsitiktinis maršrutas per įvairiuose tinklo vietose išsidėsčiusius skirtingus mazgus. Šis mechanizmas apsaugo siuntėjo tapatybę nuo žinutės gavėjo ir nuo kitų kenksmingų tinklo mazgų. Anoniminis žinutės perdavimas pademonstruotas 2.13 pav.



2.13 pav. Anoniminis užklausos maršrutizavimas. [23]

Norint išlaikyti tinkamą anonimiškumo lygį, tikimybė p_f turi būti ne mažesnė negu 0,5 ir nepasiekti 1, nes persiuntimo kanalas taptų begalinio ilgio.

Anoniminis žinučių perdavimas suteikia galimybę iniciatoriui nusiųsti pranešimą kitam tinklo mazgui, neatskleidžiant savo tapatybės, tačiau toks maršrutizavimas yra neveiksmingas, jeigu yra reikalingas atgalinis ryšys. Kadangi žinutės gavėjas nežino siuntėjo adreso. Tam, kad vartotojai galėtų pasinaudoti tokiu funkcionalumu, AP3 tinkle kuriami anoniminiai tuneliai, kurių dėka, iniciatorius gali įrašyti į siunčiamą žinutę savo atgalinį adresą ir likti anonimiškam.

Prieš sukuriant anoniminį tunelį tinkle, iš pradžių parenkamas atsitiktinis skaičius i_d, kuris tampa anoniminio kanalo adresu. Žinutės, siunčiamos šiuo tunelio adresu yra anonimiškai persiunčiamos gavėjui, o tinklo mazgai, kurie pasiunčia žinutę į anoniminį kanalą, nežino, kas yra tikrasis jos

gavėjas. Jeigu tinklo mazgas nori anonimiškai nusiųsti užklausą ir gauti atsakymą, pirmiausia jis sukuria anoniminį kanalą, sugeneruoja jo i_d (arba adresą), kurį įterpia į siunčiamą užklausą.

Kad sukurti pilną anoniminį tunelį nuo iniciatoriaus iki gavėjo (nuo pradžios iki pabaigos), iniciatorius parenka atsitiktinį adresą *L*, ir pradeda tunelio konstravimą, siunčiant P2P tinklu anonimines žinutes, tokiu būdu, kuris aprašytas prieš tai esančioje pastraipoje. Kiekvienas mazgas, dalyvavęs žinutės persiuntime, įsimena kiekvieną mazgą, iš kurio gaudavo žinutę, ir įrašo jo adresą į *persiuntimo lentelę*. Žinutė galiausiai pasiekia mazgą *s*, esantį arčiausiai arčiausia *L*, kuris, davęs sutikimą persiųsti žinutes, siunčiamas mazgui *L*, pabaigia tunelio konstravimą. Naudojant tokį tunelio konstravimo mechanizmą yra išlaikomas anonimiškumas, kadangi joks tunelio mazgas nežino ar prieš tai esantis mazgas yra žinutės autorius ar tik persiuntėjas.

Tuneliui jau egzistuojant, gavėjas nurodo persiuntimo lentelių galiojimo laiką. Pasibaigus šiam galiojimo laikui, tunelis sunaikinamas ir iniciatorius tokiu atveju turi sukurti naują tunelį.

Nustatant tunelio galiojimo laiką, gavėjui reikia atsižvelgti į tunelio mazgų kaitą, nes tunelio mazgui atsijungus nuo P2P tinklo, tunelis tampa nebefunkcionuojantis. Dėl šios priežasties, tunelio iniciatorius periodiškai turi atnaujinti anoniminį tunelį. Atnaujinimo periodas yra vidutinis tinklo mazgo gyvenimo laikas.

2.4. Darbo užduoties formulavimas

Išanalizuoti anoniminių tinklų ir sistemų MIX-NET, OPENNET, DARKNET, FREENET, MUTE, MORPHMIX, TARZAN, AP3 sudarymo principai ir nustatyta, kad publikacijose pateikiami galutiniai analizės rezultatai be detalaus matematinio modelio. Kuriant naujus anoniminius tinklus, bei rengiant mokomąją medžiagą pasigendama gilesnės tokių tinklų veikimo analizės. Remiantis atlikta anoniminių P2P tinklų analize suformuluotas darbo tikslas.

Šio darbo tikslas - *Išanalizuoti anoniminių P2P tinklų sudarymo principus ir įvertinti jų anonimiškumo užtikrinimo metodus.*

Tikslui pasiekti sprendžiami šie uždaviniai:

7. Išnagrinėti P2P tinklų sudarymo principus
8. Išanalizuoti P2P tinklų veikimo principus
9. Išanalizuoti esamas P2P tinklų anonimiškumo įvertinimo metodus
10. Sudaryti P2P tinklų anonimiškumo įvertinimo matematinį modelį
11. Naudojantis matematiniu modeliu atlikti P2P tinklų anonimiškumo įvertinimą
12. Suformuluoti rekomendacijas reikiamo lygio anonimiškumo lygio užtikrinimui

3. ANONIMIŠKUMO ĮVERTINIMO METODAS

3.1. Anonimiškumo matas

Pagrindinis anonimiško ryšio P2P perdangos tinkle principas – paslėpti ryšio kanalo informacijos siuntėją ir gavėją pasitelkus į pagalbą kitus tinklo mazgus. Jei visų tinklo mazgų elgsena vienoda (ar bent labai panaši), piktavalis turi spėti, kas buvo tikrasis siuntėjas. Jei visų tinklo mazgų skaičius yra *N*, tai tikimybė piktavaliui atspėti tikrąjį siuntėją yra $p_i = 1 / N$. Jeigu laikyti kiekvieną P2P tinklo mazgą kaip informacijos siuntėją, tai visos sistemos anonimiškumo laipsnis gali būti įvertinamas kaip entropija *H*.

Entropija yra informacijos teorijoje naudojamas dydis, kuris apibūdina vidutinį informacijos kiekį, kurį teikia vienas pranešimas (kai perdavimo kanale nėra trukdžių) arba kuris parodo, kiek informacijos gaunam atlikdami bandymą.

Entropijos savybės:

1. Entropija yra lygi nuliui tik tada, kai visos tikimybės p_i išskyrus vieną yra lygios nuliui.

2. Duotam skaičiui rezultatų *n,* entropijos vertė yra maksimali ir lygi ln(*n*), kai visos tikimybės p_i yra vienodos (lygios 1/n).

3. Informacinė entropija yra adityvi: dviejų nepriklausomų bandymų bendra entropija yra lygi jų atskirų entropijų sumai.

Anoniminį P2P tinklą, turintį N mazgų, pažymėjus *X, H(X)* bus šio tinklo entropijos reikšmė.

$$H(X) = -\sum_{i=1}^{N}(p_i log_2 p_i) = -N \cdot \frac{1}{N} log_2\left(\frac{1}{N}\right) = log_2 N \qquad (1)$$

Formulėje (1) p_i yra tikimybė identifikuoti *i*-tąjį tinklo mazgą iš visų tinklo mazgų *N* kaip tikrąjį siuntėją. Tinkle, kuriame yra *N* mazgų, ši tikimybė bus lygi: $p_i = 1 / N$

Toliau formulėse naudojami parametrai:

N – visų (piktavališkų ir nepiktavališkų) mazgų skaičius tinkle. $N = T + M$

M – piktavališkų mazgų skaičius

T – tikrųjų (nepiktavališkų) mazgų skaičius tinkle.

Santykį: N/M pažymėsime R

$$R = M / N \qquad (2)$$

Informacijos siuntėjas maršruto sudarymo pirmoje paieškoje tuo pačiu yra ir paieškos iniciatorius. Tolimesniuose paieškos žingsniuose iniciatorius yra kiekvienas mazgas, ieškantis tolimesnio maršruto.

Reikia parašyti formulę, nusakančią kas yra anonimiškimo laipsnis. P2P sistemos anonimiškumo įvertinimui naudingą skaičiuoti ne pačią entropiją, bet normuotą dydį, reiškiantį anonimiškumo laipsnį:

$D(X) = H(X) / {}_{max}H(X)$ (3)

Formulėje (3) ${}_{max}H(X) = log_2 N$.

3.2. Struktūrizuotos ir nestruktūrizuotos sistemos entropija

Struktūrizuotame anoniminiame P2P tinkle maršrutus sudarančių mazgų paieškai naudojamas DHT mechanizmas, tačiau jis nenusako ar lentelėje įrašytas mazgas tuo metu yra aktyvus. Tokiu atveju mazgas ima kitą adresą iš lentelės ir tikrina ar kitas mazgas aktyvus. Toks paieškos procesas nutekina į P2P tinklą perteklinę informaciją, pagal kurią atakuotojas gali identifikuoti patį paieškos procesą.

Konstruojant anoniminį tunelį, tikimybė kad paieškos procesas nebus identifikuotas pirmojo bandymo metu:

$$P_1 = \frac{N - M}{N} = \frac{T}{N} \qquad (4)$$

Tikimybė, kad proceso identifikacija bus nesėkminga po *k* iniciatoriaus paieškos bandymų:

$$P_k = \left(\frac{N - M}{N}\right)^k = \left(\frac{T}{N}\right)^k \qquad (5)$$

Tikimybė, kad po *k* bandymų bus identifikuotas paieškos procesas:

$P_a = 1 - P_k$. Jeigu paieškos procesas bus identifikuotas, piktavalis gali identifikuoti iniciatoriaus (siuntėjo) mazgą, tačiau jis negali identifikuoti tunelio, kuriam iniciatorius priklauso. Šiam tikslui pasiekti, atakuotojas turi kontroliuoti paskutinį anoniminio tunelio mazgą. Tegul *S* reiškia įvykį, kad tikrasis siuntėjas bus identifikuotas, tai tokio įvykio tikimybė:

$$P(S) = (1 - P_k)R \qquad (6)$$

Tikimybė, kad po įvykio *S* tikrasis siuntėjas nebus identifikuotas: $P(S_{ne}) = 1 - P(S)$.

Kai P2P sistemos visų mazgų aptikimo tikimybė vienoda, tada sistemos entropija:

$E = \log_2(N - M)$, (7)

ir tokios sistemos anonimiškumo laipsnis:

$D(X) = E/log_2 N$. (8)

Naudojantis entropijos aditivumo savybe P2P sistemos entropija yra dviejų entropijų suma:

$$A = A_1 + A_2 = P(S) \cdot H(X/S) + \big(1 - P(S)\big) \cdot log_2(N - M) \qquad (9)$$

Čia: A_1 – piktavalių mazgų skaičiaus sąlygojama entropija, A_2 – tikrųjų mazgų sąlygojama entropija. Kai visos tikimybės p_i bus lygios P2P tinklo entropija:

$$A = A_1 + A_2 = P(S) \cdot log_2 M + \big(1 - P(S)\big) \cdot log_2(N - M) \qquad (10)$$

Struktūrizuotos P2P sistemos anonimiškumo laipsnis D(X)

$$D(X) = \frac{P(s) log_2 M + \big(1 - P(S)\big) log_2(N - M)}{log_2 N} \qquad (11)$$

$$D(X) = \frac{P(s)log_2M - P(s)log_2(N-M) + log_2(N-M)}{log_2N} \quad (12)$$

$$D(X) = \frac{P(s)(log_2M - log_2(N-M)) + log_2(N-M)}{log_2N} \quad (13)$$

$$D(X) = \frac{\left(P(s)log_2\left(\frac{M}{N-M}\right)\right) + log_2(N-M)}{log_2N} \quad (14)$$

Entropija nestruktūrizuotoje P2P sistemoje apskaičiuojama: $A_2 = log_2(N - M)$. Iš to seka, kad nestruktūrizuotos P2P sistemos anonimiškumo laipsnis bus:

$$D(X) = \frac{A_2}{log_2N} \quad (15)$$

3.3. Tunelio anonimiškumo įvertinimas

Norint nustatyti tikrąjį siuntėją anoniminiame ryšio tunelyje, atakuotojas turi kontroliuoti pirmąjį ir kai kuriuos kitus tunelio mazgus (įskaitant ir paskutinį), kurie leis nustatyti iš kokio kaimyninio mazgo gauta žinutė ir kuriam kaimynui ji persiųsta. Anoniminiame ryšio tunelyje kiekvienas persiuntimo mazgas žino tik mazgą esantį prieš jį ir už jo, taigi atakuotojui reikia kontroliuoti dagiau negu pusę tunelio mazgų, siekiant jį visiškai atkurti ir identifikuoti iniciatorių. Tarkime tunelis sudarytas iš V_1, V_2, V_3, V_4 ir V_5 mazgų. Kad atstatyti visą tunelį (įvardinti jame dalyvavusius mazgus) reikia žinoti bent V_1, V_3 ir V_5. Tikimybė, kad bus nustatytas visas L ilgio tunelis:

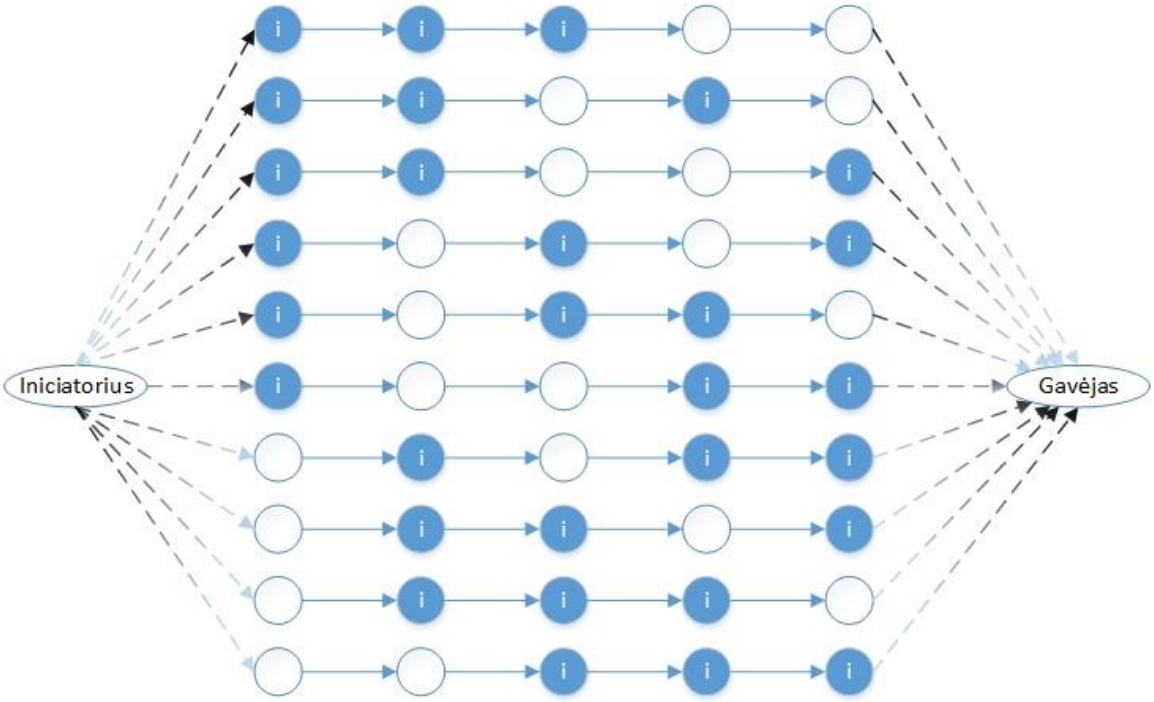
$P(S) = R^{\frac{L}{2}+1}$, kai *L* yra lyginis. (16)

$P(S) = R^{\frac{L+1}{2}}$, kai *L* yra nelyginis. (17)

R = M/N – santykis tarp tikrųjų ir piktavalių mazgų tinkle.

Tikimybė, kad nebus identifikuotas siuntėjas: *P(Ne) = 1 – P(S).*

Nors atakuotojas ir nekontroliuoja mazgų (ar reikiamo jų kiekio), kurie galėtų sukompromituoti anoniminį tunelį, jis gali bandyti atspėti siuntėją, pagal mazgus kuriuos kontroliuoja (ar kontroliavo). Tarkime, tunelio ilgis *L=5*, o atakuotojo kontroliuojamų mazgų jame *i=3*, tai yra 10 skirtingų variantų (apskaičiuojama panaudojus *derinius* C_5^3), kaip mazgai galėjo skirtingai būti išsidėstę tunelyje. Jeigu atakuotojo kontroliuojamus *i* mazgus parinksime skirtinga tvarka, vistiek gausime ta patį *i=3* mazgų rinkinį tunelyje.



3.1 pav. Atakuotojo valdomų mazgų išsidėstymo būdai tunelyje, kurio ilgis *L=5*, (C_5^3).

Šiuo atveju gausime derinių skaičių

$$C_L^i = \frac{L!}{i!\,(L-i)!} = \frac{5!}{3!\,(2)!} = \frac{5 \cdot 4}{2} = 10$$

Tegul E_i yra įvykis, kad i mazgų ($I <= L$) yra kontroliuojami priešininko, tai tikimybė, kad bus nustatytas visas tunelis:

$$P(E_i) = C_L^i \cdot {P_1}^{(L-i)} \cdot R^i, \qquad (18)$$

Formulėje (18) P_1 lygus:

$$P_1 = \frac{N-M}{N} = \frac{T}{N} \qquad (19)$$

Įvertinus visus galimus derinius L ilgio tunelyje entropija (anonimiškumas) išreiškiama kaip atskirų derinių entropijų suma:

$$H(X) = A_1 + P(S_{Ne}) \sum_{i=0}^{L} P(E_i) log_2(N-i) \qquad (20)$$

Čia: A_1 – piktavalių mazgų skaičiaus sąlygojama entropija,

$$A_1 = P(S) \cdot log_2 M \qquad (21)$$

Panagrinėkime atskirus atvejus:

L = 1. Įvertinus (16) arba (17) formules

$$P(S) = \left(\frac{M}{N}\right)^1$$

$$P(E_0) = 1\left(1-\frac{M}{N}\right)^1\left(\frac{M}{N}\right)^0$$

$$P(E_1) = 1\left(1-\frac{M}{N}\right)^0\left(\frac{M}{N}\right)^1$$

$$H(X) = \frac{M}{N}H(X/S) + (1-\frac{M}{N})\left\{\left(1-\frac{M}{N}\right)^1 Log_2 N + \left(1-\frac{M}{N}\right)^0\frac{M}{N}Log_2(N-1)\right\}$$

Jei *L* = *2*, tai

$$P(S) = \left(\frac{M}{N}\right)^2$$

$$P(E_0) = 1\left(1-\frac{M}{N}\right)^2\left(\frac{M}{N}\right)^0$$

$$P(E_1) = 2\left(1-\frac{M}{N}\right)^1\left(\frac{M}{N}\right)^1$$

$$P(E_1) = 1\left(1-\frac{M}{N}\right)^0\left(\frac{M}{N}\right)^2$$

$$H(X) = \left(\frac{M}{N}\right)^2 H(X/S) + \left(1-\left(\frac{M}{N}\right)^2\right)\left\{\left(1-\frac{M}{N}\right)^2\left(\frac{M}{N}\right)^0 Log_2 N + 2\left(1-\frac{M}{N}\right)^1\left(\frac{M}{N}\right)^1 Log_2(N-1) + \left(1-\frac{M}{N}\right)^0\left(\frac{M}{N}\right)^2 Log_2(N-2)\right\}$$

Jei *L* = *3*, tai

$$P(S) = \left(\frac{M}{N}\right)^2$$

$$P(E_0) = 1\left(1-\frac{M}{N}\right)^3\left(\frac{M}{N}\right)^0$$

$$P(E_1) = 3\left(1-\frac{M}{N}\right)^2\left(\frac{M}{N}\right)^1$$

$$P(E_2) = 3\left(1-\frac{M}{N}\right)^1\left(\frac{M}{N}\right)^2$$

$$P(E_3) = 1\left(1-\frac{M}{N}\right)^0\left(\frac{M}{N}\right)^3$$

$$H(X) = \left(\frac{M}{N}\right)^2 H(X/S) + \left(1 - \left(\frac{M}{N}\right)^2\right)\left\{\left(1 - \frac{M}{N}\right)^3 \left(\frac{M}{N}\right)^0 Log_2 N + 3\left(1 - \frac{M}{N}\right)^2 \left(\frac{M}{N}\right)^1 Log_2(N-1) + \left(1 - \frac{M}{N}\right)^1 \left(\frac{M}{N}\right)^2 Log_2(N-2) + \left(1 - \frac{M}{N}\right)^0 \left(\frac{M}{N}\right)^3 Log_2(N-3)\right\}$$

Jei *L* = *4*, tai

$$P(S) = \left(\frac{M}{N}\right)^3$$

$$P(E_0) = 1\left(1 - \frac{M}{N}\right)^4 \left(\frac{M}{N}\right)^0$$

$$P(E_1) = 4\left(1 - \frac{M}{N}\right)^3 \left(\frac{M}{N}\right)^1$$

$$P(E_2) = 6\left(1 - \frac{M}{N}\right)^2 \left(\frac{M}{N}\right)^2$$

$$P(E_3) = 4\left(1 - \frac{M}{N}\right)^1 \left(\frac{M}{N}\right)^3$$

$$P(E_4) = 1\left(1 - \frac{M}{N}\right)^0 \left(\frac{M}{N}\right)^4$$

$$H(X) = \left(\frac{M}{N}\right)^3 H(X/S) + \left(1 - \left(\frac{M}{N}\right)^3\right)\left\{\left(1 - \frac{M}{N}\right)^4 \left(\frac{M}{N}\right)^0 Log_2 N + 4\left(1 - \frac{M}{N}\right)^3 \left(\frac{M}{N}\right)^1 Log_2(N-1) + 6\left(1 - \frac{M}{N}\right)^2 \left(\frac{M}{N}\right)^2 Log_2(N-2) + 4\left(1 - \frac{M}{N}\right)^1 \left(\frac{M}{N}\right)^3 Log_2(N-3) + \left(1 - \frac{M}{N}\right)^0 \left(\frac{M}{N}\right)^4 Log_2(N-4)\right\}$$

Jei *L* = *5*, tai

$$P(S) = \left(\frac{M}{N}\right)^3$$

$$P(E_0) = 1\left(1 - \frac{M}{N}\right)^5 \left(\frac{M}{N}\right)^0$$

$$P(E_1) = 5\left(1 - \frac{M}{N}\right)^4 \left(\frac{M}{N}\right)^1$$

$$P(E_2) = 10\left(1 - \frac{M}{N}\right)^3 \left(\frac{M}{N}\right)^2$$

$$P(E_3) = 10\left(1 - \frac{M}{N}\right)^2 \left(\frac{M}{N}\right)^3$$

$$P(E_4) = 5\left(1 - \frac{M}{N}\right)^1 \left(\frac{M}{N}\right)^4$$

$$P(E_5) = 1\left(1-\frac{M}{N}\right)^0\left(\frac{M}{N}\right)^5$$

$$H(X) = \left(\frac{M}{N}\right)^3 H(X/S) + \left(1-\left(\frac{M}{N}\right)^3\right)\left\{\left(1-\frac{M}{N}\right)^5\left(\frac{M}{N}\right)^0 Log_2 N + 5\left(1-\frac{M}{N}\right)^4\left(\frac{M}{N}\right)^1 Log_2(N-1) + 10\left(1-\frac{M}{N}\right)^3\left(\frac{M}{N}\right)^2 Log_2(N-2) + 10\left(1-\frac{M}{N}\right)^2\left(\frac{M}{N}\right)^3 Log_2(N-3) + \left(1-\frac{M}{N}\right)^1\left(\frac{M}{N}\right)^4 Log_2(N-4) + \left(1-\frac{M}{N}\right)^0\left(\frac{M}{N}\right)^5 Log_2(N-5)\right\}$$

Analogiškai galima gauti P2P sistemos tunelio, kurio ilgis $L > 5$, entropijų išraiškas, pagal kurias sekančioje dalyje pateiktos charakteristikos.

3.4. Atsitiktinio ilgio tunelio anonimiškumo įvertinimas

Prie kiekvieno skirtingo ilgio tunelio, gaunama skirtinga $H(X|L=x)$ reikšmė. Tegul su tikimybe $P(L=1)$ bus vieno mazgo ilgio tunelis, $P(L=2)$ – dviejų mazgų, $P(L=3)$ – trijų mazgų ir t.t. Tada bendra sistemos entropija, kai L yra atsitiktinis, apskaičiuojama remiantis tunelio ilgio tikimybiniu pasiskirstymu. Pavyzdžiui, tunelio ilgis gali būti atsitiktinis intervale $L \in [1;5]$, tai kiekvienas skirtingas tunelio ilgis bus su tikimybe $P(L = x)$, kur x kinta intervale nuo 1 iki 5. Tai remiantis entropijos adityvumo principu bendras P2P sistemos anonimiškumas, kai L yra atsitiktinis, apskaičiuojamas sudedant kiekvieno tunelio ilgio entropijas, prieš tai kiekvieną jų padauginus iš atitinkamos tikimybės:

$$\bar{H}(L) = P(L=1)\cdot H(X|L=1) + P(L=2)\cdot H(X|L=2) + \cdots + P(L=5)\cdot H(X|L=5) \quad (22)$$

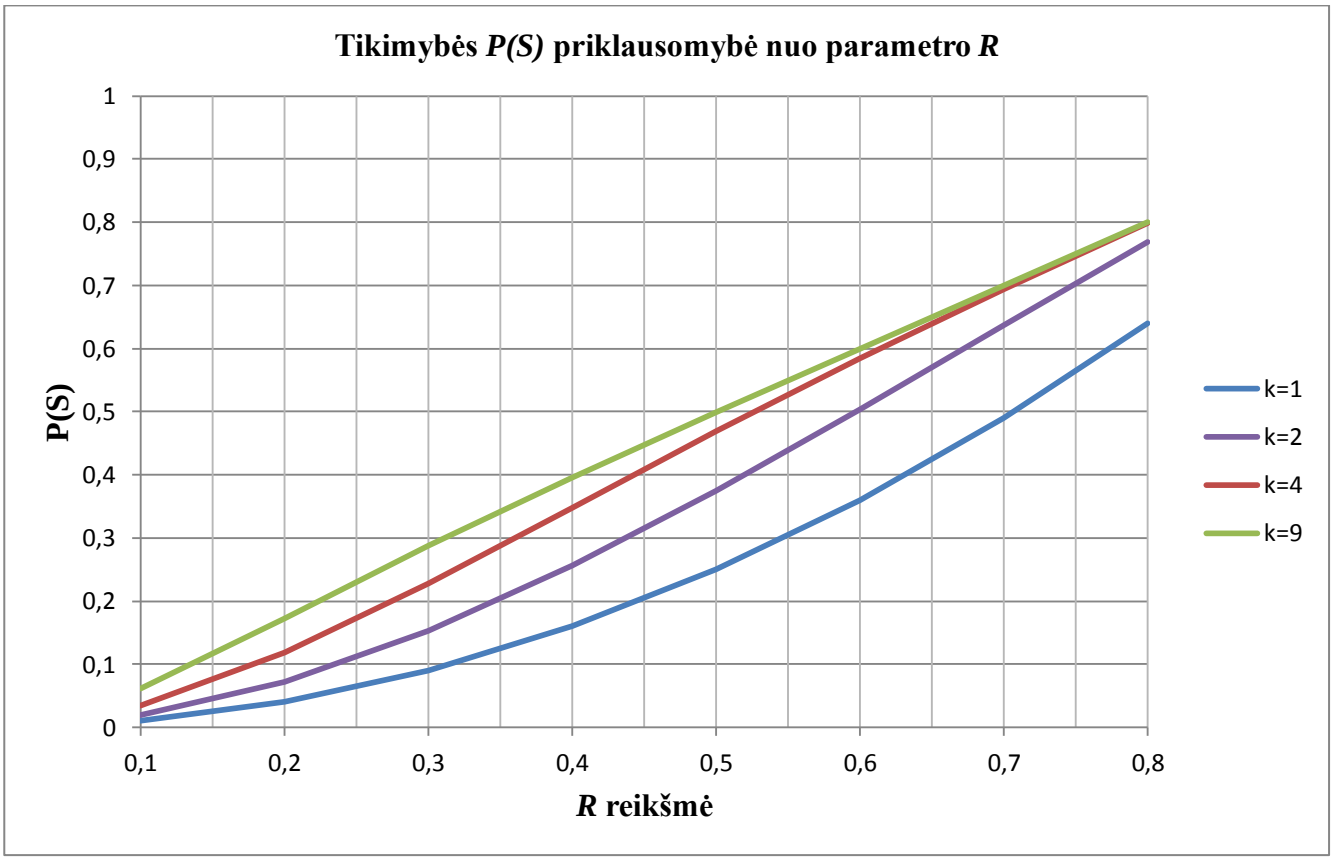
Entropijų $H(X)$ reikšmės skaičiuojamos pagal aukščiau pateiktas išraiškas. Atsitiktinio ilgio tunelio anonimiškumas apskaičiuojamas pagal formulę:

$$\bar{D}(L) = P(L=1)\cdot D(X|L=1) + P(L=2)\cdot D(X|L=2) + \cdots + P(L=5)\cdot D(X|L=5) \quad (23)$$

4. MODELIAVIMO REZULTATAI

4.1. Anonimiškumo charakteristika: tikimybės *P(S)* priklausomybė nuo parametrų *R* ir *k*.

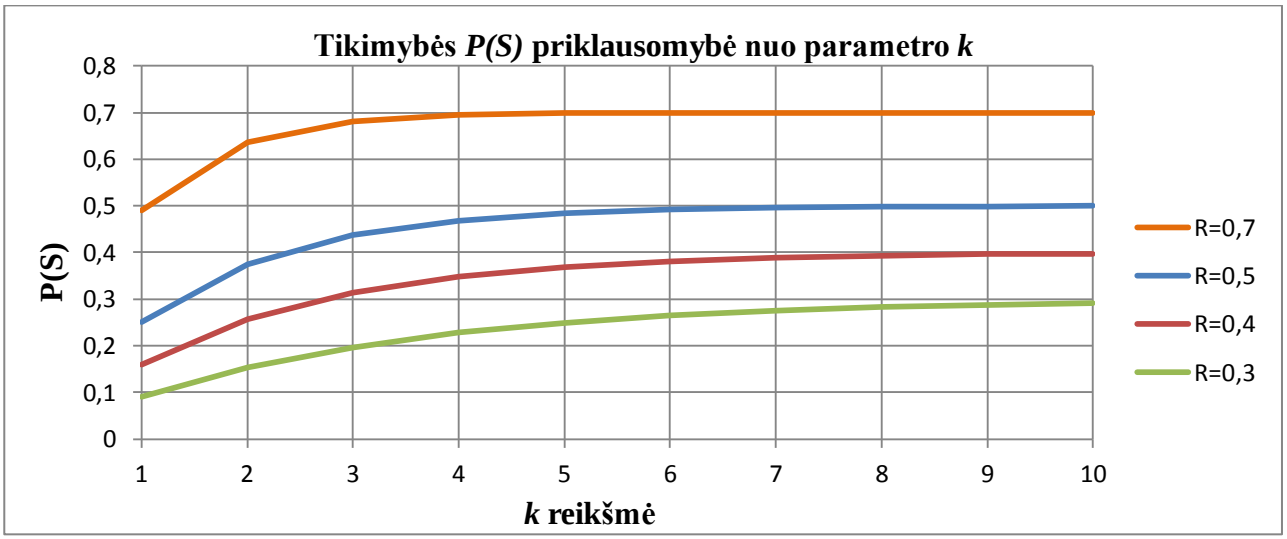
Remiantis (6) formule, tiriamos dvi tikimybės *P(S)* priklausomybės: nuo piktavalių dalies visame tinkle *R*, ir nuo mazgo paieškos bandymų skaičiaus *k*. Reikšmė *S* nusako įvykį, kada tikrasis siuntėjas bus identifikuotas. Tikimybės *P(S)* priklausomybė nuo parametro *R* pateikta pav. 4.1.



4.1 pav. Tikimybė, kad siuntėjas bus identifikuotas, prie skirtingų *R* reikšmių.

Didėjant piktavalių mazgų santykiniam skaičiui anoniminiame tinkle, tikimybė identifikuoti mazgą didėja. Kai visi mazgai piktavaliai *(R=1)*, tai *P(S)=1*. Kai R artėja prie 0, *P(S) = 0*. Grafike atvaizduoti keturi atvejai, kada kito mazgo paieškos bandymų kartai (*k* = 1, 2, 4, 9). Kiekvienu atveju, *P(S)* dydis prie skirtingos *R* reikšmės yra skirtingas, kuo didesnis paieškos bandymų skaičius, tuo tikimybė *P(S)* yra didesnė. Grafike matoma, kad *P(S)* dydis beveik tiesiogiai proporcingas *R* (0 < R < 1). *P(S)* reikšmė grafike, kada *R* = 0,5, priklausomai nuo *k*, neviršis 0,5: kai *k* yra 1, *P(S)* = 0,25; kai *k* = 2, *P(S)* = 0,375; kai *k* = 4, *P(S)* = 0,469; kai *k* = 9, *P(S)* = 0,499. Didėjant paieškos bandymų skaičiui, tikimybė identifikuoti mazgą artėja prie *R* reikšmės.

Kitame grafike nagrinėjama *P(S)* priklausomybė nuo parametro *k* (pav. 4.2).

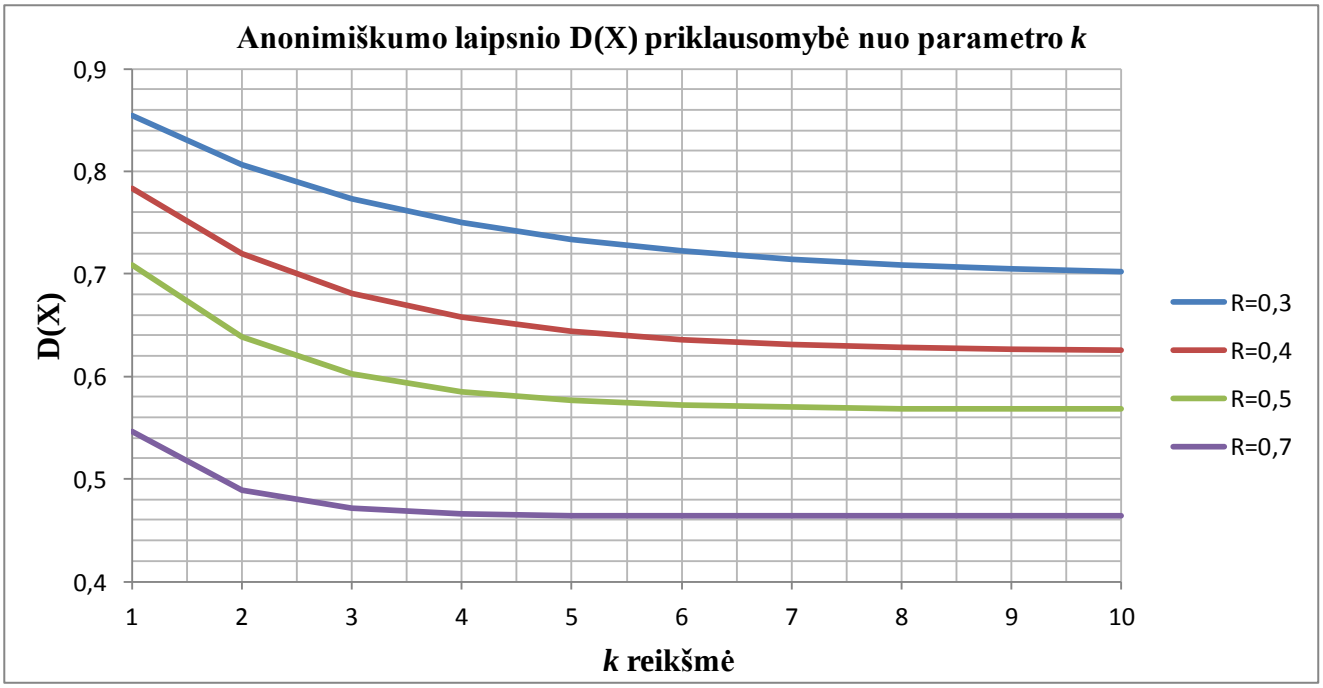


4.2 pav. Tikimybė, kad siuntėjas bus identifikuotas, prie skirtingų *k* reikšmių.

Tikimybė *P(S)*, didėjant tinklo mazgo paieškos bandymų skaičiui *k*, didėja iki tol, kol *k* reikšmė susilygina su viso anoniminio P2P tinklo mazgų ir piktavalių mazgų santykiu *R* ir nepriklauso nuo bandymų skaičiaus *k*.

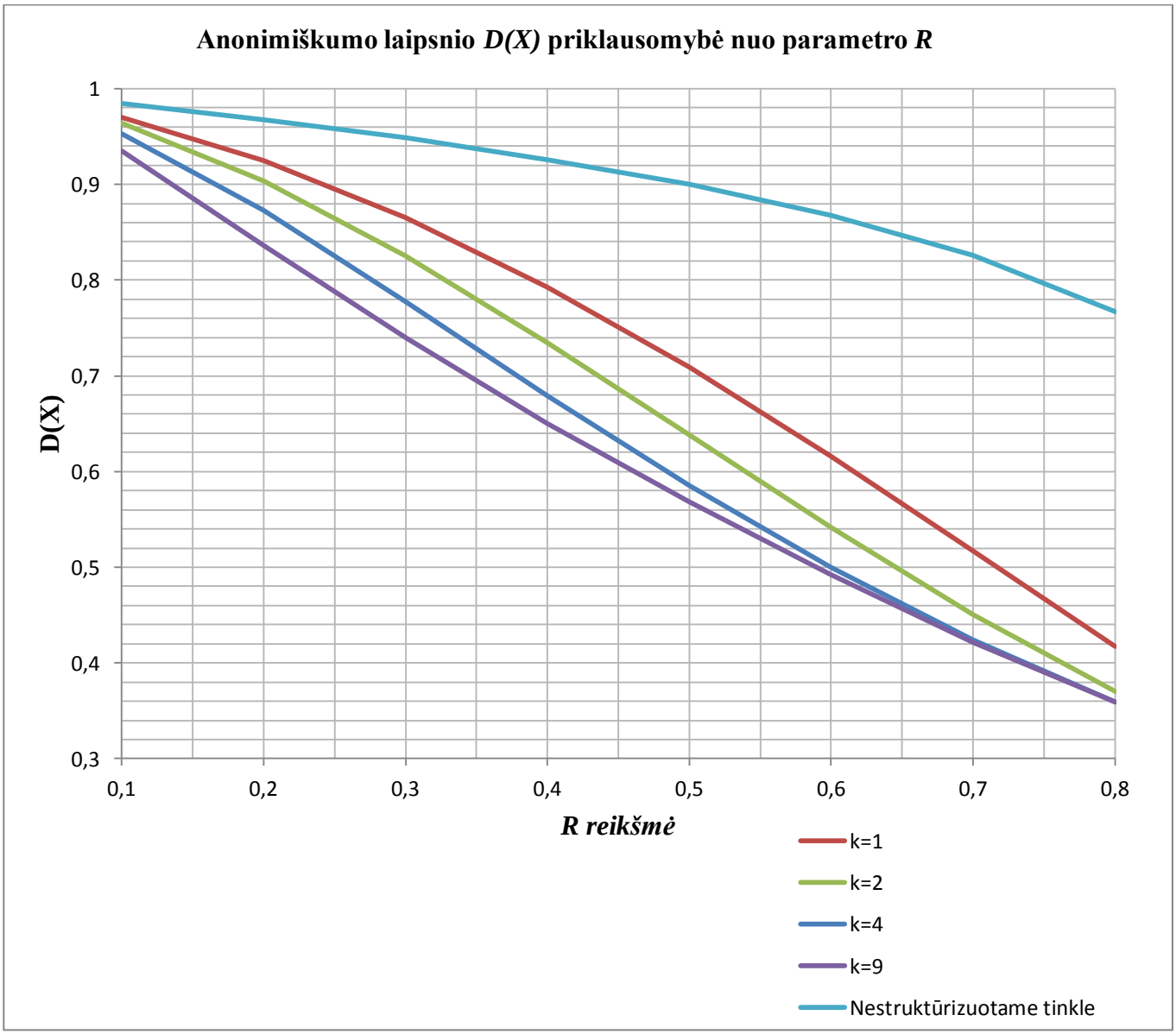
4.2. Anonimiškumo charakteristika: *D(X)* priklausomybė nuo parametrų *R* ir *k*.

Struktūrizuoto P2P tinklo anonimiškumo laipsnis skaičiuojamas pagal (14) formulę. Į ją įeina tikimybė *P(S)*, kuri priklauso nuo *k*, todėl tiriame *D(X)* priklausomybę nuo *k* (4.3 pav.).



4.3 pav. Anonimiškumo laipsnis prie skirtingų *k* reikšmių.

Didėjant paieškos bandymų skaičiui *k*, anonimiškumo laipsnis mažėja, kol pasiekia *(1 - R)* reikšmę. Didžiausias anonimiškumo laipsnio dydis yra, kai *k* = 1. Grafike nagrinėjami keturi atskiri atvejai, kai *R* = 0,3; 0,4; 0,5; 0,7. Didėjant *R* reikšmei, *D(X)* reikšmės kitimo pobūdis išlieka toks pat, tik grafikas truputį pasistumia į apačią. Kai *R* = *1* (visi tinkle mazgai piktavaliai), tai anonimiškumo laipsnis lygus nuliui (4.3 pav.).



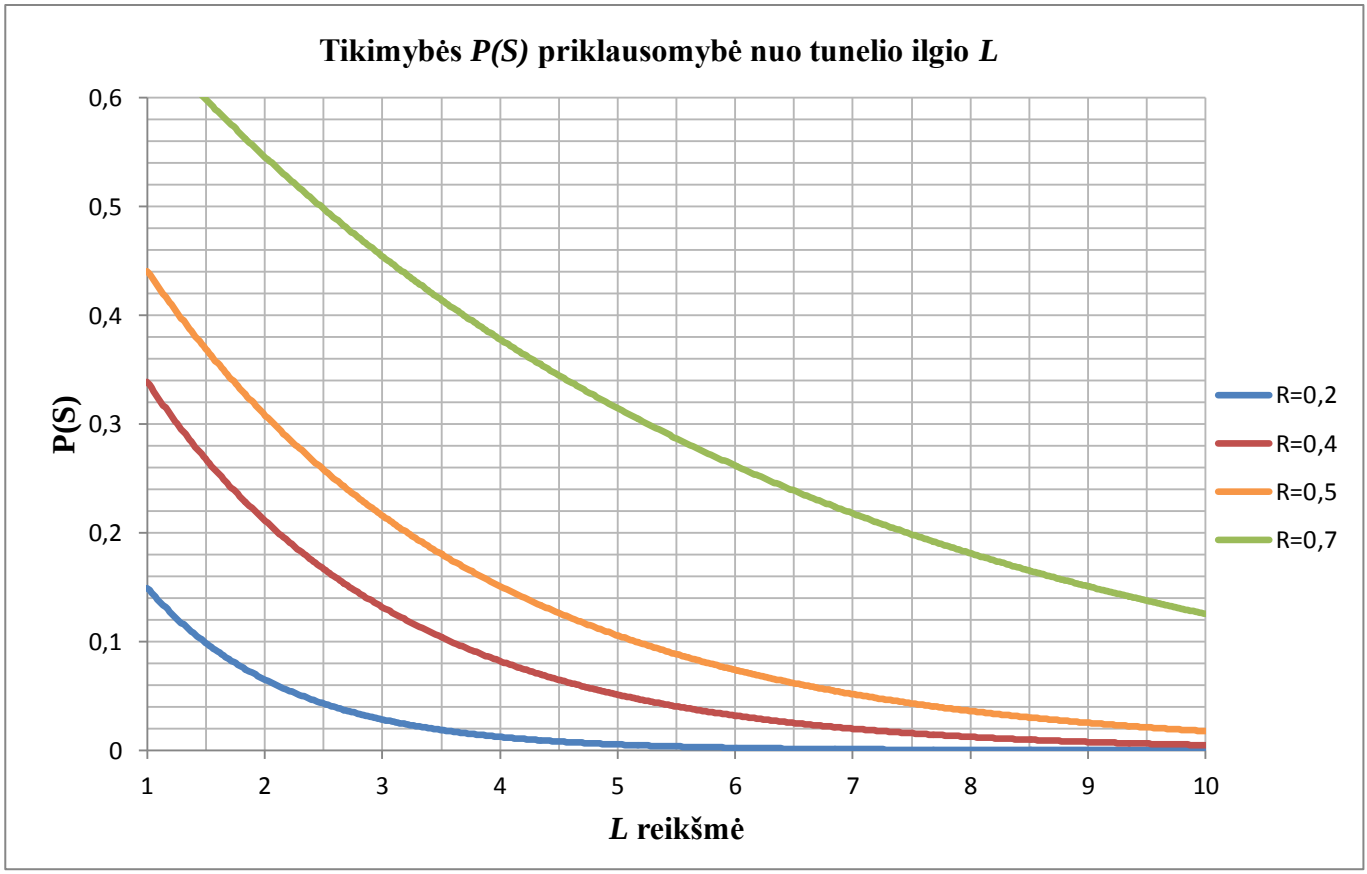
4.4 pav. Anonimiškumo laipsnis prie skirtingų *R* reikšmių.

Didėjant piktavalių mazgų skaičiui tinkle, tinklo anonimiškumo laipsnis mažėja (4.4. pav.). Nagrinėjami penki atvejai: keturiais atvejais struktūrizuotas anoniminis tinklas, ir vienu – nestruktūrizuotas. Kreivės, kai *k* = 1, 2, 4 ir 9 kinta panašiai – tinklo anonimiškumo laipsnis proporcingas piktavalių mazgų skaičiui tinkle, tačiau nestruktūrizuoto tinklo kreivei, dydis *R* daro daug mažesnę įtaką – anonimiškumo laipsnis mažėja daug lėčiau, negu struktūrizuoto tinklo atveju. Taip yra todėl, kad iniciatorius nežino viso tinklo topologijos ir mazgų paieškai naudoja žinomus

arba kitų mazgų rekomenduojamus aktyvius mazgus, todėl skaičiuojant jo anonimiškumo laipsnį, dydis *k* nereikalingas (formulė (15)) ir įtakos rezultatams neturi.

4.3. Anonimiškumo charakteristika: tikimybės *P(S)* priklausomybė nuo parametrų *L* ir *R*.

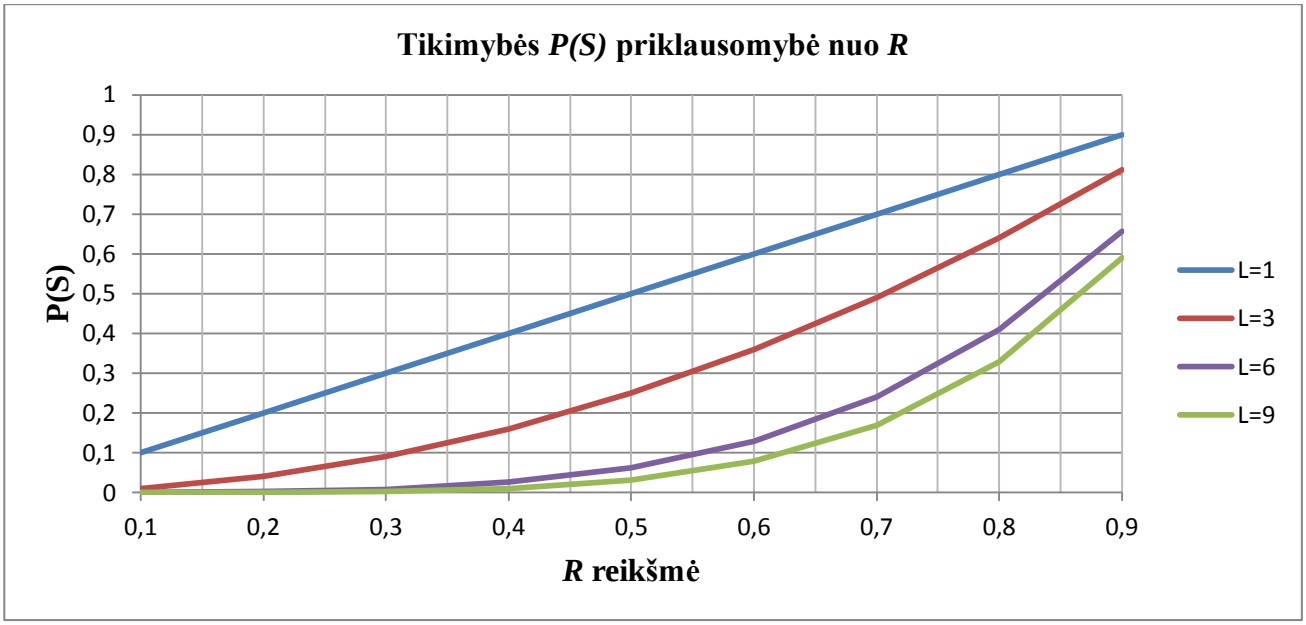
Dydis *S* nusako įvykį, kad tikrasis siuntėjas bus identifikuotas (formulės (16) ir (17)). Tikimybės *P(S)* priklausomybė nuo tunelio ilgio *L* atsispindi grafike (4.5 pav.).



4.5 pav. Tikimybė, kad siuntėjas bus identifikuotas, prie skirtingų *L* reikšmių.

Didėjant anoniminio tunelio ilgiui, tikimybė identifikuoti tikrąjį siuntėją mažėja, tačiau niekada nepasieks reikšmės, kad siuntėjas yra visiškai anonimiškas *(P(S) = 0))*. Kai piktavalių mazgų santykinis dydis yra labai mažas, o tunelio ilgis labai didelis, tai *P(S)* artėja prie 0. Nagrinėjami anoniminiai tinklai, kuriuose yra skirtingi piktavalių ir visų tinklo mazgų santykiai (*R* = 0,2; 0,4; 0,5 ir 0,7). Kuo daugiau piktavalių mazgų tinkle, tuo tikimybė identifikuoti tikrąjį siuntėją didesnė, prie tų pačių tunelio ilgių *L*.

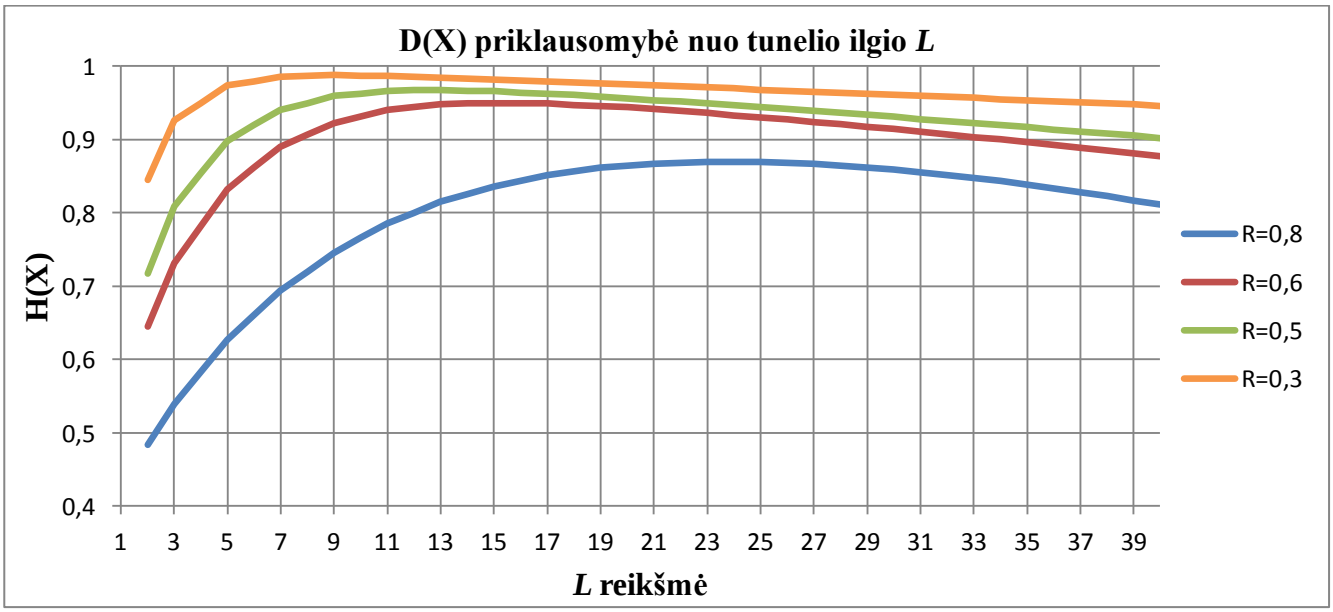
Kitame grafike matome *P(S)* priklausomybę nuo parametro *R*. Didėjant piktavalių tinklo mazgų skaičiui, tikimybė *P(S)* didėja (pav. 4.6). Nagrinėjami keturi atvejai (*L* = 1, 3, 6, 9). Kai *L = 1*, *R* priklausomybė nuo *P(S)* yra tiesinė. Kuo tunelio ilgio *L* reikšmė didesnė – tuo tikimybė identifikuoti siuntėją lėčiau kinta, didėjant piktavalių mazgų skaičiui.



4.6 pav. Siuntėjo identifikavimo tikimybės priklausomybė nuo *R* prie skirting tunelio ilgių.

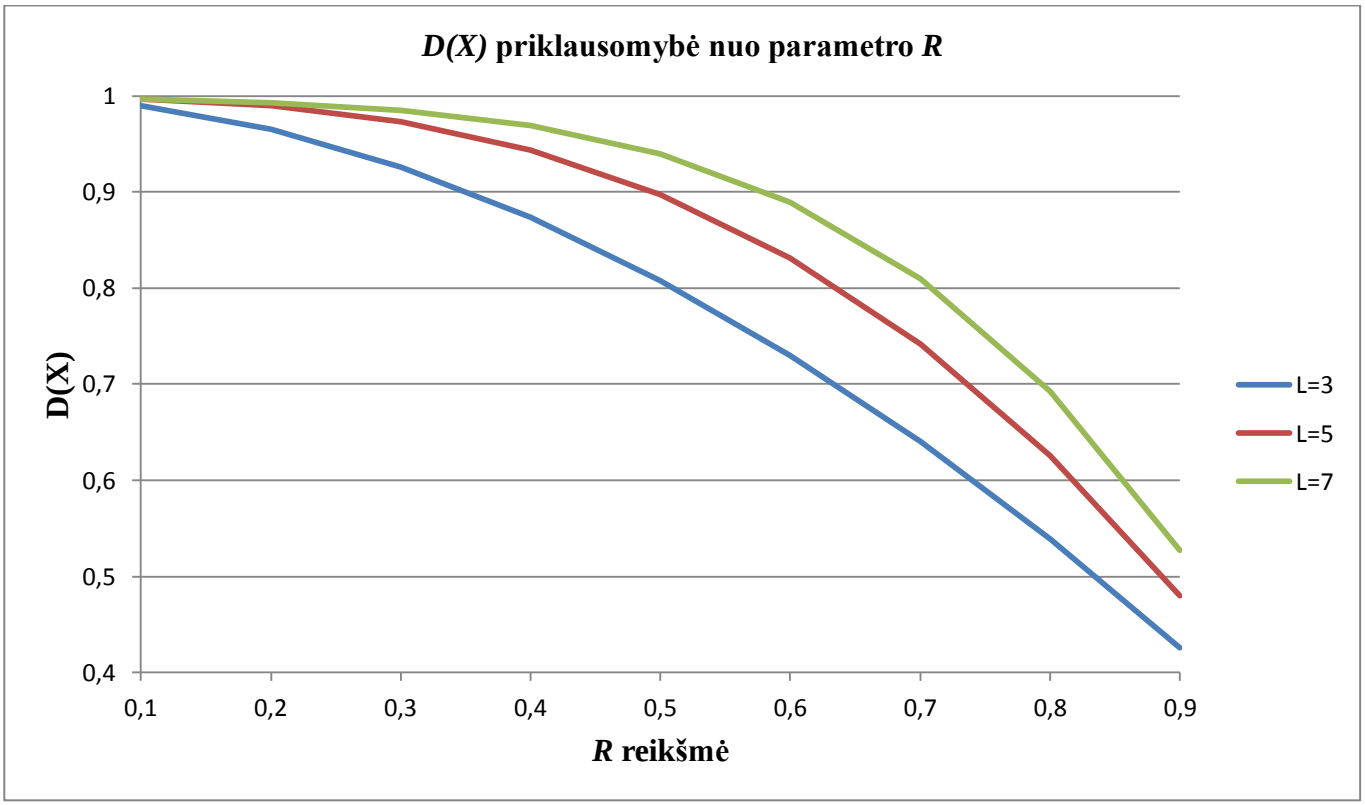
4.4. Anonimiškumo charakteristika: *D(X)* priklausomybė nuo parametrų *L* ir *R*.

Anonimiškumo laipsnis (formulė (20)) labai priklauso nuo tunelio ilgio *L* ir piktavalių mazgų santykinio skaičiaus (4.7 pav.). Kai piktavalių mazgų santykinis skaičius artėja prie 0, tai anonimiškumo laipsnis artėja prie 1, nepriklausomai nuo tunelio ilgio. Tuo atveju, kai piktavalių mazgų santykinis skaičius artėja prie 1, anonimiškumo laipsnis artėja prie 0. Didėjant tunelio ilgiui prie $L > 10$, anonimiškumo laipsnis mažėja, nes piktavaliai mazgai nuspėja visą maršrutą žinant mažiau maršruto tikrųjų mazgų.



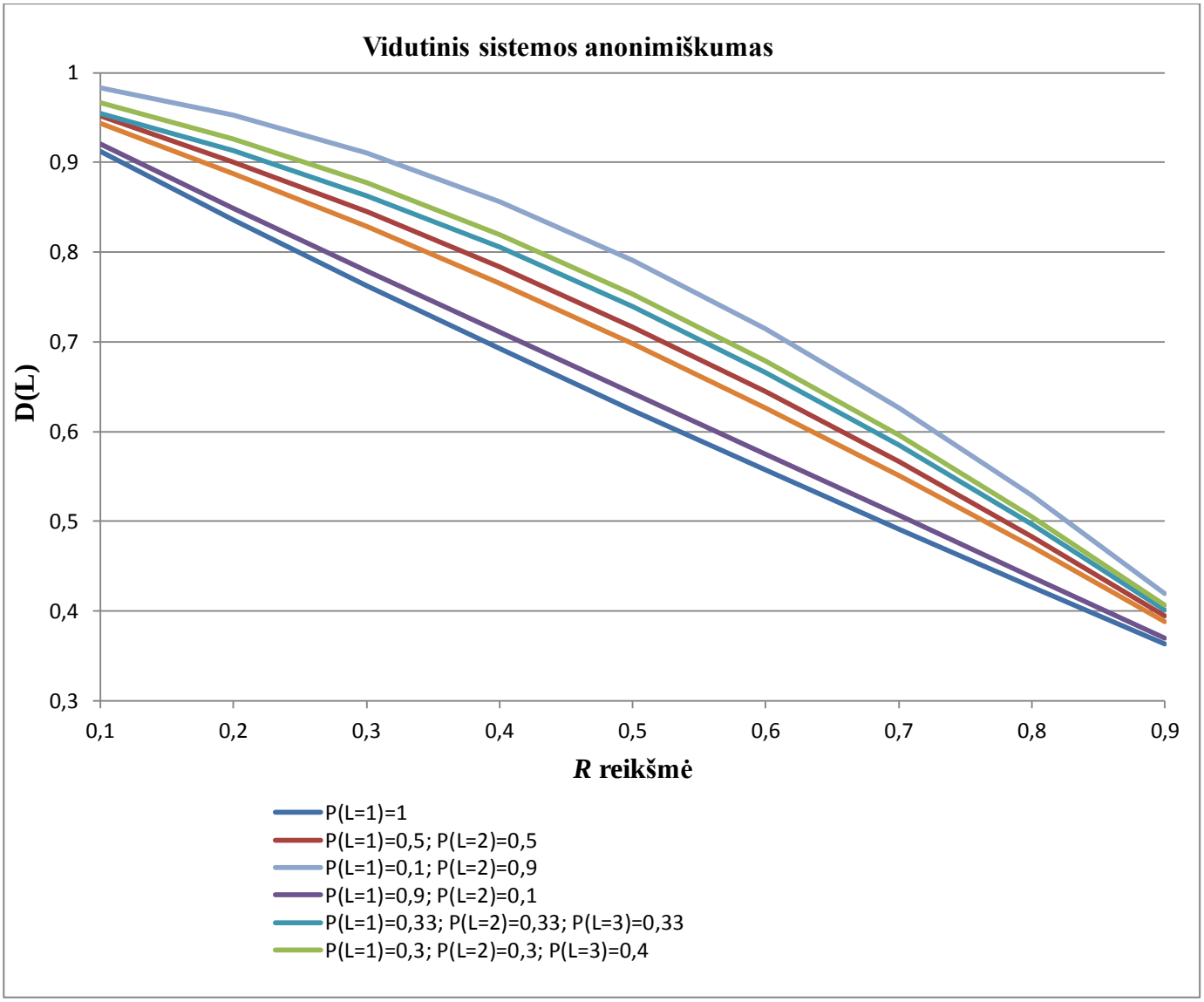
4.7 pav. Anonimiškumo laipsnis prie skirtingų *L* reikšmių.

Tunelio ilgiui didėjant, *H(X)* grafike staigiai kyla, tačiau tunelio ilgiui pasiekus reikšmę 10, anonimiškumo laipsnis daugumoje kreivių nustoja kilti, ir po truputį pradeda mažėti. Tunelio ilgiui pasiekus tam tikrą ribą (priklauso nuo įvairių aplinkybių ir anoniminio tinklo charakteristikų) piktavalis turi daugiau šansų prisijungti ir atspėti siuntėją. Toks fenomenas reiškia, kad dažnos silpnos atakos anoniminiame tinkle gali pakenkti tinklo anonimiškumui.



4.8 pav. Anonimiškumo laipsnis prie skirtingų *R* reikšmių.

Didėjantis piktavalių mazgų skaičius anoniminiame tinkle daro panašią įtaką įvairių tunelio ilgių *L* sistemų anonimiškumo laipsniui (4.8 pav). Grafike matosi, kad anonimiškumo laipsnis tiesiogiai proporcingas dydžiui *R*. Palyginus tris kreives su skirtingais tunelio ilgiais (L = 3, 5, 7) matome, kad kuo didesnis tunelio ilgis, tuo anonimiškumo laipsnis mažėja lėčiau. Piktavalių mazgų ir tinklo mazgų santykiui *R* pasiekus 1, tinklo anonimiškumo laipsnis priartės prie 0.



4.9 pav. Vidutinis sistemos anonimiškumas.

Grafike matomi atsitiktinio tunelio ilgio anonimiškumo skaičiavimo metodo rezultatai (4.9 pav.) Metodo principas (formulės (22) ir (23)) – kiekvienas tunelio ilgis gali būti nustatomas su tam tikra tikimybe. Įvertinus šias tikimybes, apskaičiuojamas vidutinis sistemos anonimiškumo laipsnis pasinaudojus anonimiškumo laipsnio normavimą pagal *L*. Grafike pateiktos septynios skirtingos kreivės, su skirtingais tunelių pasiskirstymais pagal ilgį. Visi dėsningumai yra tokie patys, kaip anksčiau pateiktuose grafikuose, tačiau tokiu būdu galima tiksliau apskaičiuoti *visos* P2P sistemos anonimiškumo laipsnį. *P(L = x)* nusako tikimybę, kad *x* ilgio anoniminis tunelis bus suformuotas tarp tinklo mazgų. Anonimiškumo laipsniai prie visų *P(L = 1)*, *P(L = 2)* ir *P(L = 3)* reikšmių buvo panaudoti tie patys, todėl grafike puikiai matosi *D(X)* normavimo pagal atsitiktinį tunelio ilgį rezultatai: kreivių reikšmės skirtingos, prie vienodų tunelių anonimiškumo laipsnių.

5. DARBO REZULTATAI

1. Atlikta anoniminių P2P tinklų analizė parodė, kad siuntėjo ir gavėjo anonimiškumui užtikrinti gali būti naudojami tiek šifravimo mechanizmai, tiek ir maršrutų slėpimo mechanizmai. Didžioji dalis anoniminių tinklų naudoja maršrutų parinkimo informacijos šifravimą. Tam tikra dalis P2P tinklų naudoja paslėptus tinklo mazgus, per kuriuos sudaromi informacijos perdavimo tarp siuntėjo ir gavėjo maršrutai.
2. Sukurta ir pritaikyta P2P tinklų anonimiškumo įvertinimo metodika, įgalinanti paskaičiuoti anonimiškumo laipsnį tiek struktūrizuotos, tiek ir nestruktūrizuotos P2P sistemos atveju. Struktūrizuotos P2P sistemos anonimiškumo laipsnis visada yra mažesnis už nestruktūrizuotos P2P anonimiškumo laipsnį, nes informaciją apie galimus maršrutus turi nemažai tinklo mazgų.
3. Ištirta maršruto tunelio ilgio įtaka anonimiškumo laipsniui. Nustatyta, kad tinkamas tunelio ilgis yra tampriai susijęs su piktavališkų mazgų santykiniu skaičiumi visų tinklo mazgų atžvilgiu. Kuo daugiau P2P sistemoje piktavalių mazgų, tuo ilgesnį tunelį reikia naudoti.
4. Esant atsitiktinai kintančiam piktavalių vartotojų skaičiui, tunelio ilgis taip pat turi kisti atsitiktiniu dėsniu. Pasiūlytas modelis, įgalinantis įvertinti P2P sistemos entropiją, kai tunelio ilgis yra atsitiktinis. Atsitiktinis tunelio ilgis turi pranašumą prieš fiksuotą tunelio ilgį, nes piktavalis mazgas turi papildomai atspėti ir tunelio ilgį. Šis faktas didina sistemos neapibrėžtumą ir tuo pačiu ir entropiją.
5. Anonimiškumo įvertinimo metodas ir jo panaudojimo metu gauti rezultatai yra svarbūs kuriant anonimines P2P sistemas, pagrįstas atsitiktinių maršrutų sudarymo principais. Gautos anonimiškumo laipsnio skaitinės reikšmės įgalina parinkti tunelių ilgis atsižvelgiant į piktavalių mazgų santykinį skaičių P2P sistemoje.

6. LITERATŪRA

[1] Rimantas Plėštys, Dangis Rimkus, „Informacijos perdavimo tinklo vartotojų anonimiškumo įvertinimo metodika“ *Technologijos mokslo darbai Vakarų Lietuvoje,* pp. 245-248, 2010.

[2] „Peer-to-peer - Wikipedia,“ [Tinkle]. Available: http://en.wikipedia.org/wiki/Peer-to-peer. [Kreiptasi 03 gegužės 2013].

[3] S. A. Theotokis, D. Spinellis, „A Survey of Peer-to-Peer Content Distribution Technologies,“ *ACMCS,* pp. 335–371, 2004.

[4] Chithra Selvaraj, Sheila Anand, „A survey on Security Issues of Reputation Management Systems for Peer-to-Peer Networks,“ *SciVerse ScienceDirect,* pp. 145-160, 2012.

[5] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim, „A Survey and Comparison of Peer-to-Peer Overlay Network Schemes,“ *IEEE Communications Survey and Tutorial,* 2004.

[6] „Distributed hash table - Wikipedia,“ [Tinkle]. Available: http://en.wikipedia.org/wiki/Distributed_hash_table. [Kreiptasi 08 gegužės 2013].

[7] Claudia Roncancio, Maria del Pilar Villamil, Cyril Labbe, Patricia Serrano-Alvarado, „Data sharing in DHT based P2P systems,“ *Transactions on Large-Scale Data- and Knowledge-Centered Systems,* pp. 327-352, 2009.

[8] Baptiste Pretre, Roger Wattenhofer, Stefan Schmid, „Attacks on Peer-to-Peer Networks,“ [Tinkle]. Available: http://disco.ethz.ch/theses/ss05/freenet.pdf. [Kreiptasi 17 balandžio 2013].

[9] Ehsan Saboori, Shahriar Mohammadi, „Anonymous Communication in Peer-to-Peer Networks for Providing more Privacy and Security,“ *International Journal of Modeling and Optimization,* t. 2, 2012.

[10] A. Pfitzmann, M. Hansen, „Anonymity, Unobervability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology,“ 29 gegužės 2006. [Tinkle]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml. [Kreiptasi 8 balandžio 2013].

[11] T.Chothia, K. Chatzikokolakis, „A Survey on Anonymous Peer-to-Peer File-Sharing,“ *NCUS,* 2005.

[12] Sharon Shitrit, Eyal Felstaine, Niv Gilboa, Ofer Hermoni, „Anonymity Scheme for Interactive P2P Services,“ *Eighth IEEE International Symposium on Cluster Computing and the Grid,* pp. 33-40, 2008.

[13] Chigusa Kawashima, I. G. B. Baskara Nugraha, Hiroyoshi Morita, „Realizing and Evaluating Mutual Anonymity in P2P Networks,“ *ISITA,* pp. 66-71, 2010.

[14] D. L. Chaum, „Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms,“ *Technical Note Programming Techniques and Data Structures,* 1988.

[15] „OpenNet Initiative,“ [Tinkle]. Available: http://p2pfoundation.net/OpenNet_Initiative. [Kreiptasi 11 sausio 2013].

[16] „The Darknet and the Future of Content Distribution,“ [Tinkle]. Available: http://www.bearcave.com/misl/misl_tech/msdrm/darknet.htm. [Kreiptasi 4 gruodžio 2012].

[17] „Simple, Anonymous File Sharing,“ [Tinkle]. Available: http://mute-net.sourceforge.net/index.php. [Kreiptasi 21 vasario 2013].

[18] „The Freenet Project,“ [Tinkle]. Available: http://freenetproject.org. [Kreiptasi 10 sausio 2012].

[19] M. Rennhard, B. Plattner, „Introducing morphmix: Peer-to-peer based anonymous internet usage with collusion detection,“ *Proc. Workshop on Privacy in the Electronic Society,* p. 91–102, 2002.

[20] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, „Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications,“ įtraukta *In Proceedings of the Workshop on Design Issues in Anonymity and Unobservability,* 2000.

[21] W. Diffie, M. E. Hellman, „New directions in cryptography,“ *IEEE Transactions on Information Theory,* pp. 644-654, 1976.

[22] M. J. Freedman, R. Morris, „Tarzan: A Peer-to-Peer Anonymizing Network Layer,“ *Proc. the 9th ACM conference on computer and communications security,* p. 193–206, 2002.

[23] A. Mislove, G. Oberoi, A. Post, C. Reis, P. Druschel, D. S. Wallach, „Ap3: cooperative, decentralized anonymous communication,“ *Proc. 11th ACM SIGOPS European workshop,* 2004.

7. PRIEDAI

7.1. Freenet tinklo samprata ir bandymas

Freenet anonimiško duomenų apsikeitimo programinę įrangą galima prilyginti labai didelės talpos laikmenai. Kiekvienam viešai platinamam failui priskiriamas raktas, kurio reikia, norint tą failą gauti (parsisiųsti). Vartotojas, pateikęs failą Freenet sistemai, gauna tą raktinį failą atitinkančią informaciją. Saugojimo vieta yra paskirstyta tarp visų Freenet tinklo mazgų.

Freenet yra anonimiškas, decentralizuotas taškas į tašką (arba P2P) tinklas. Šio tinklo mazgai (arba klientai) žino tik savo kaimyninius mazgus, tačiau neturi jokios informacijos apie bendrą tinklo struktūrą.

Freenet sukurtas laikantis principo, jog kiekvieną sykį jungiantis prie tinklo, klientas jungiasi prie tų mazgų, kuriuos jau žino. Šie mazgai jungiasi prie mazgų kuriuos jie žino (ir taip toliau). Tokiu būdu tinklo mazgas gali pasiekti bet kurį kitą Freenet tinklo mazgą, kad ir kaip jis būtų nutolęs.

Duomenų saugojimas

Visi Freenet tinklo mazgai aukoja dalį savo kietųjų diskų (ar kitokių laikmenų) vietos informacijos laikymui. Visa informacija yra užšifruojama ir saugoma Freenet klientinės programos įdiegimo aplanke.

Skirtingai nuo kitų P2P programų, vartotojas neturi beveik jokios duomenų, saugojamų jo kompiuteryje, kontrolės. Vietoj to, failai yra saugomi arba trinami pagal jų populiarumą. Toks duomenų saugojimo ir reguliavimo būdas apsaugo Freenet tinklą nuo cenzūros. Vienintelis būdas panaikinti tam tikrus duomenis iš Freenet tinklo yra visiškai jų neieškoti ir tikėtis, jog kiti vartotojai taip pat neieškos.

Freenet maršrutizavimas

Iš pradžių, kiekvienas naujas Freenet mazgas neturi jokios informacijos apie kitų jam žinomų mazgų veikimą ir charakteristikas. Tai reiškia, jog užklausimų maršrutizavimas yra visiškai atsitiktinis. Kadangi skirtingi mazgai turi skirtingą maršrutizavimo atsitiktinumą (skirtingai nukreipia užklausas), jie greičiausiai tarpusavyje nesutars, kur siųsti užklausą pagal gautą raktą. Todėl duomenys naujame Freenet tinkle keliauja visiškai atsitiktiniais maršrutais.

Tam pačiam mazgui įkėlus vis daugiau skirtingų dokumentų, jie bus pradėti grupuoti su duomenimis, kurių raktai yra panašūs, nes visiems duomenims yra taikomos vienodos maršrutizavimo taisyklės. Negana to, kai duomenų elementų ir užklausų keliai iš įvairių mazgų „susikerta“, jie taip pat pradeda dalintis grupavimo informacija.

Viso to rezultatas – tinklas pats susireguliuos į paskirstytą, sugrupuotą struktūrą, kurioje tinklo mazgai laiko duomenų elementus, kurie pagal duomenų raktus yra arti vienas kito. Greičiausiai tokių duomenų grupių (angl. *clusters*) visame tinkle bus daugybė, bet kuris duoto dokumento yra padaroma daugybė kopijų priklausomai kaip dažnai jis yra naudojamas.

Duomenų pasiekimas

Norint pasiekti tam tikrus duomenis Freenet tinkle, gali būti naudojamas Fproxy. Reikia žinoti duomenų, kuriuos reikia pasiekti, raktą.

Freenet tinkle yra keturi raktų tipai:

- CHK – Turinio maišos raktai (angl. *Content Hash Keys*)
- SSK – Pasirašyti poerdvio raktai (angl. *Signed Subspace Keys*)
- USK – Atnaujinami poerdvio raktai(angl. *Updateable Subspace Keys*)
- KSK – Raktinio žodžio pasirašyti raktai (angl. *Keyword Sgned Keys*)

CHK raktai yra labiausiai fundamentalūs. Visi failai, kurie užima daugiau negu vieną kilobaitą (1 KB) galiausiai yra padalinami į vieną ar daugiau 32-jų kilobaitų (32 KB) dydžio CHK raktus. CHK rakto failo vardas yra nustatomas tik pagal jo turinį. SSK yra kitas bazinis raktų tipas. Šie raktai sujungia viešąjį raktą su vartotojui skirtu perskaityti failo vardu. KSK yra kitas SSK rakto variantas kuriame viskas nustatoma panaudojant žmogui lengvai perskaitomą failo vardą. USK raktai yra atnaujinamų raktų forma, kuri yra ypatingai naudinga adreso skaidymo raktui (angl. *Address Resolution Keys* arba ARK).

Adreso skaidymo raktas (ARK) yra USK, kuris yra įterpiamas mazgo, pasikeitus jo IP adresui. Jame saugoma nuoroda į mazgą – jo kriptografiniai duomenys, o ypač, jo IP adresas. ARK yra geriausias sprendimas, siekiant padėti žmonėms prisijungti prie Freenet tinklo, jeigu jie turi bėdų, kurias sukėlė ugniasienės, maršrutizatoriai ar pasikeitęs IP adresas.

ARK raktas yra įgyvendinimo elementas (angl. *implementation detail*) ir vartotojui naudojantis Freenet tinklu, nieko nereikia apie jį žinoti.

Turinio maišos raktai

Turinio maišos raktai (CHK) yra failai turintys statinį turinį (pvz. *mp3* failai arba *PDF* dokumentai). Šie raktai yra failų turinio maišos rezultatas. Maiša yra atkuriamas metodas, kada tam tikra duomenų dalis paverčiamas į santykiniai nedidelį informacijos vienetą, kuris naudojamas kaip tų duomenų „piršto antspaudas“. Turiniui pasikeitus, kad ir nežymiai, maišos failas pasikeičia radikaliai. Tai leidžia pastebėti bet kokius mėginimus keisti ar gadinti duomenis. CHK suteikia failui unikalų identifikatorių, todėl neįmanoma, jog tu skirtingi failai galėtų turėti tą patį CHK. Turinio maišos raktai sudaryti iš trijų dalių:

1. Duomenų maišos failo
2. Dešifravimo rakto, kuris atrakina raktą

3. Kriptografinių nustatymų

Tipinio CHK rakto struktūra:

```
CHK@Failo maišos funkcija,Iššifravimo raktas,Kriptografiniai nustatymai
```

CHK rakto pavyzdys:

```
CHK@SVbD9~HM5nzf3AX4yFCBc-A4dhNUF5DPJZLL5NX5Brs,bA7qLNJR7IXRKn6uS5PAySjIM6azPFvK~18kSi6bbNQ,AAEA--8
```

Šifro raktas laikomas užšifruotas duomenų failo viduje, todėl neįmanoma iššifruoti failo be CHK rakto.

Pasirašyti poerdvio raktai

Pasirašyti poerdvio raktai (SSK) yra naudojami svetainių, kurios laikui bėgant gali keistis. Pavyzdžiui svetainės, talpinančios informaciją, kurią dažnai reikia atnaujinti, redaguoti, pridėti ar ištrinti. SSK raktai naudojami tam, kad kas nors kitą nepradėtų platinti pakeistos svetainės kito vartotojo vardu.

SSK raktai veikia panaudodami viešojo rakto kriptografiją, suteikdami galimybę vartotojui užregistruoti savo svetainę. Tik asmuo turintis slaptąjį raktą gali į Freenet sistemą įkelti naujas to asmens svetainės versijas. SSK raktai yra sudaryti iš šių penkių dalių.

- **viešojo rakto maišos funkcija** – ši dalis yra viskas ko reikia norint identifikuoti failą (bet jo neiššifruoti), todėl tinklo mazgai tik šią dalį saugo. Pilnas viešasis raktas saugomas nešifruotas su užšifruotais duomenimis**.**
- **dokumentų dešifravimo raktas –** jis žinomas tinklo klientams, o ne mazgams, saugantiems duomenis, todėl mazgai negali iššifruoti duomenų be pilno adreso.
- **kripto nustatymai** – panaudoti kriptografiniai šifravimo algoritmai**.**
- **pasirinktas vartotojo vardas** – žodis ar sakinys, kuriuos parenka svetainės autorius.
- **versija** – dabartinė svetainės versija.

Versijos numeris atnaujinamas kiekvieną kartą, kada nauja svetainės versija sukuriama ir įkeliama į Freenet tinklą. Šis metodas naudojamas, nes šiuo metu nėra įmanoma atnaujinti jau į Freenet tinklą įterptų duomenų.

Tipinio SSK rakto struktūra:

```
SSK@viešojo failo maišos funkcija,šifro raktas,kriptografiniai nustatymai/vartotojo pasirinktas svetainės pavadinimas-versija
```

SSK rakto pavyzdys:

```
SSK@GB3wuHmt[..]o-eHK35w,c63EzO7u[..]3YDduXDs,AQABAAE/mysite-4
```

Pasirašytų poerdvio raktų veikimas

- Autorius sugeneruoja kriptografinių raktų porą: privatų raktą, skirtą failų pasirašymui ir viešąjį raktą, kuris reikalingas parašo tikrinimui.
- Autorius taip pat sugeneruoja vieną simetrinį raktą (kuris naudojamas užšifravimui ir dešifravimui).
- Kada failas įkeliamas į Freenet tinklą, jis yra užšifruojamas simetriniu raktu ir pasirašomas privačiu raktu. Parašas saugomas kartu su failu. Tinklo mazgai netalpina simetrinių raktų, tik SSK rakto viešąją dalį, skirta duomenims indeksuoti. Tokiu būdu tinklo mazgas gali neigti jog žino jame saugomos informacijos turinį.

SSK yra sudarytas iš viešojo ir simetrinio raktų maišos funkcijų. Viešojo rakto maišos funkcija yra skirta duomenims indeksuoti paieškos tikslais. Taip pat viešas raktas saugomas kartu su duomenimis, kurie migruoja po Freenet tinklą. Todėl tinklo mazgai gali patikrinti parašą, kai pas juos atklysta SSK failas, taip pat Freenet tinklo klientai gaudami norimą failą gali patikrinti jo parašą. Simetrinio rakto dėka, tinklo vartotojai gautą failą iššifruoja.

SSK raktus naudojantys puslapiai daugeliu atveju buvo pakeisti puslapiais, naudojančiais atnaujinamus poerdvio raktus (USK), kurie yra panašūs į SSK raktus, tačiau skirtingai nuo jų, visada vartotojui bando pateikti naujausią galimą svetainės versiją.

Atnaujinami poerdvio raktai

Atnaujinami poerdvio raktai (USK) yra naudingi susiejant SSK raktą su naujausia svetainės versija. Galima teigti, kad USK raktai yra patogi priemonė, į kurią „įvelkami“ SSK raktai, kuri paslepia naujesnių svetainių ieškojimo procesą.

Tipinio USK rakto struktūra:

USK	@	viešojo rakto maišos funkcija	,	šifro raktas	,	kriptografiniai nustatymai	/	vartotojo pasirenkamas svetainės vardas	/	numeris	/	

USK rakto struktūra yra beveik identiška SSK rakto struktūrai, išskyrus versija/numeriu. Egzistuoja dviejų tipų USK raktų adresai:

- USK raktas, turintis teigiamą numerį gale.
- USK raktas, turintis neigiamą numerį gale.

USK raktas su teigiamu numeriu veikia taip: Freenet mazgas vartotojo kompiuteryje laiko jam žinomų USK raktų versijų sąrašą, nesaugodamas duomenų. Šis sąrašas sudarytas iš prieš tai buvusių prisijungimų prie tinklo mazgo, taip pat iš foninių užklausų iš praeitų vizitų i tokio tipo adresus. Kada vartotojas aplanko USK adresą (žemiau esantis pavyzdys), svetainių sąrašas tikrinamas ir ieškoma vartotojo svetainės su numeriu į ar didesniu. Jeigu tokios svetainės egzistuoja, vartotojui grąžinama naujausia svetainės versija. Tada foniniame režime ieškoma naujesnių svetainės versijų

kurios iki šiol nebuvo žinomos. Jei tokių versijų randama, jos įdedamos į vartotojo USK registrą kitam kartui, kai bus aplankomas tas pats adresas.

Pavyzdys su numeriu 5:

```
USK@GB3wuHmt[..]o-eHK35w,c63EzO7u[..]3YDduXDs,AQABAAE/mysite/5/
```

Kada vartotojas aplanko nuorodą į svetainę su neigiamu numeriu gale, Freenet ieško tos versijos, kurios vartotojas pareikalavo (pvz.: -7) plius dar keturių versijų (pvz.: 7, 8, 9, 10) mazge kuriame yra jo kompiuteris ir kituose mazguose. Jeigu randama tik septinta versija, vartotoju ji bus ir pateikta. Jeigu buvo rasta ne tik septinta versija, ieškoma dar vieno versijų rinkinio (pvz. 12, 13, 14, 15, 16). Procesas kartojamas tol kol randama tik viena versija, o kitos keturios – ne. Tada vartotojui gražinamas aukščiausios rastos versijos puslapis.

Pavyzdys su numeriu -7:

```
USK@GB3wuHmt[..]o-eHK35w,c63EzO7u[..]3YDduXDs,AQABAAE/mysite/-7/
```

Raktažodžio parašo raktai (angl. *Keyword Signed Keys*)

Raktažodžio parašo raktai (KSK) leidžia vartotojui išsaugoti vardinius Freenet puslapius. Jie nėra atsparūs brukalų siuntimui ar puslapio vardo užgrobimui. Keli skirtingi vartotojai gali į Freenet tinklą įkelti skirtingus duomenų failus su vienodais adresais. Tačiau tuo atveju tinkle veikia pasikartojimų aptikimo sistema, kuri bando užkirsti kelią kartą jau įterpto failo perrašymui.

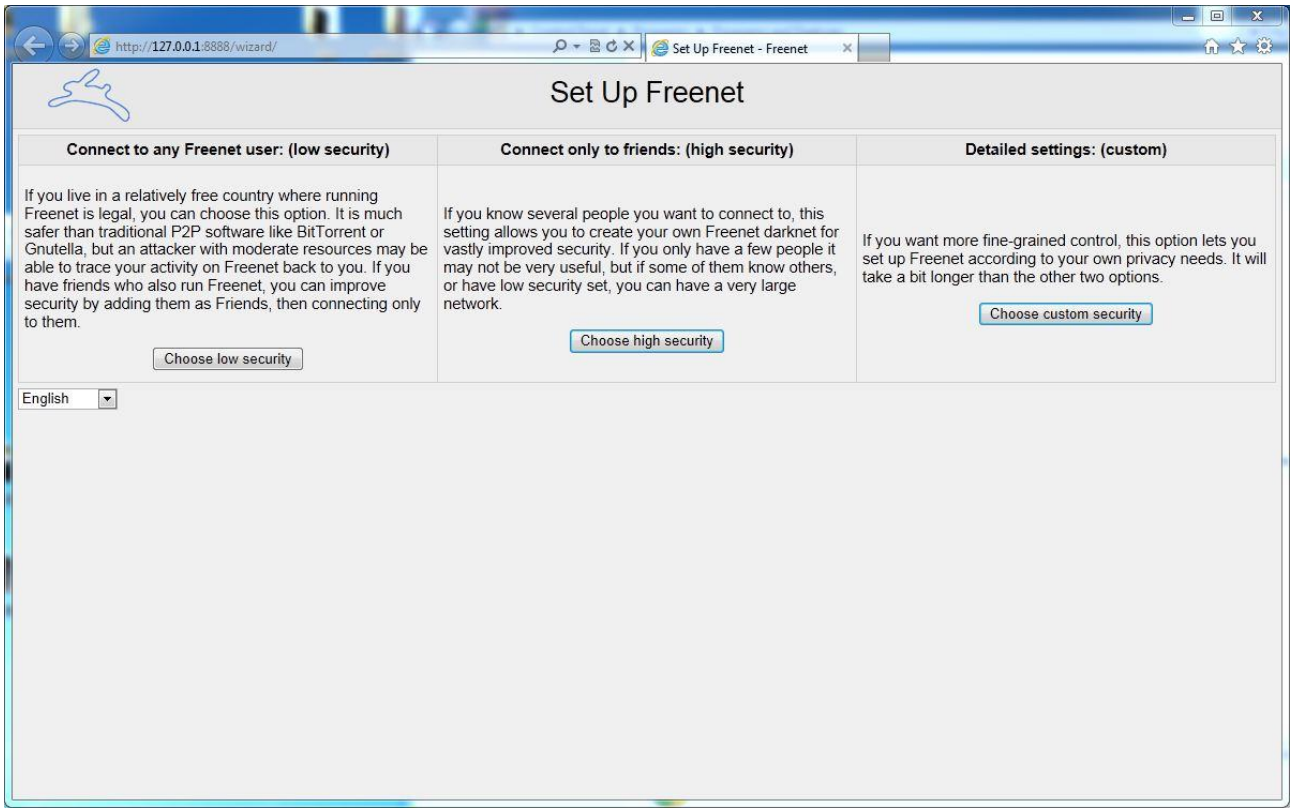
Pagrindinis KSK raktų trūkumas yra tas, jog bet kas gali įdėti failą, kurio pavadinimas sutampa su jau įdėto failo pavadinimu, ir nukreipt tinklo srautą nuo seniau įdėto failo prie savojo. KSK privalumas – žmogui suprantami ir nesunkiai atsimenami nuorodų pavadinimai.

KSK rakto adresas gal saugoti nukreipimą į CHK adresą arba į patį save.

Konteineriai

Konteineris, bendruoju Freenet požiūriu, yra failas, kuriame yra keletas kitų failų. Konteinerių privalumas – vartotojui pakrovus vieną puslapį, jis mato visus failus tame puslapyje. Jis turi pasirinkimą, jam užkraunami visi failai arba neužkraunamas nė vienas, tai leidžia sumažinti puslapiui reikalingų raktų skaičių.

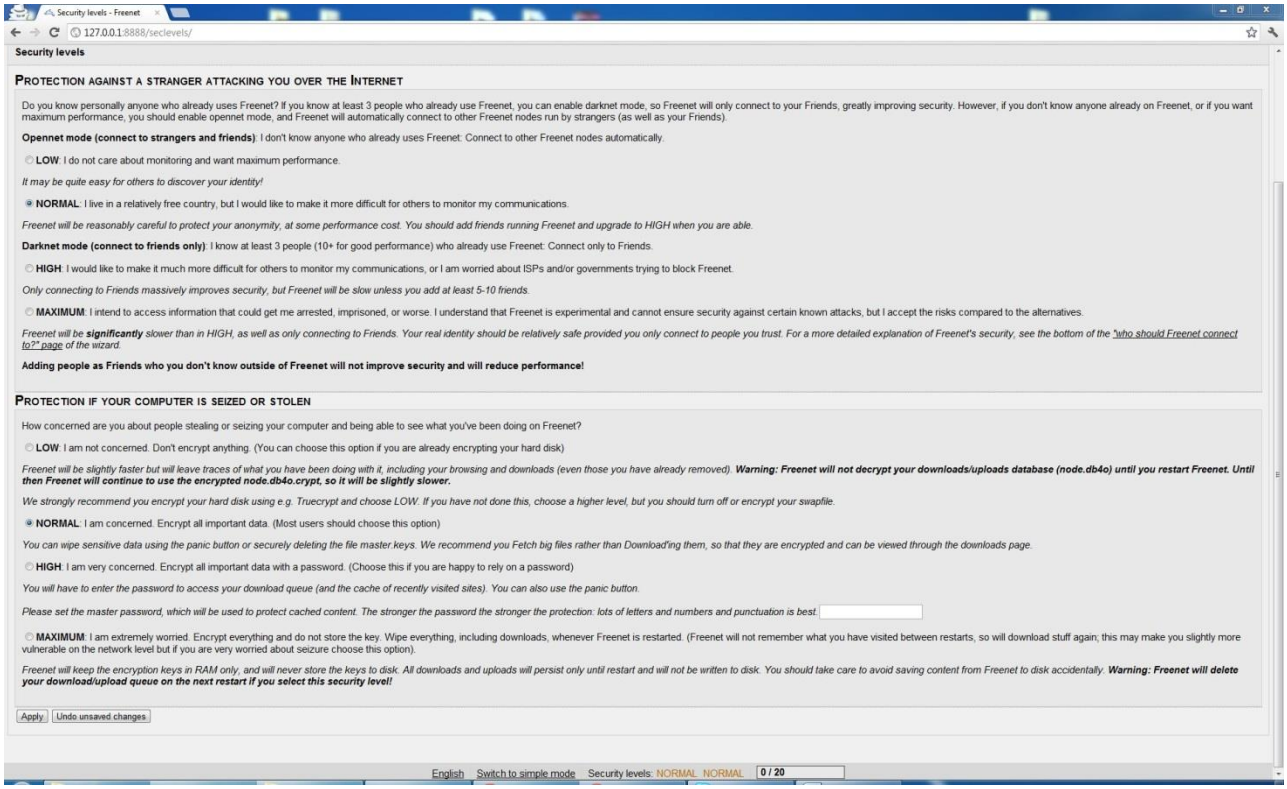
Freenet klientinės programos patikrinimas



7.1 pav. Freenet pradinis paleidimo vedlys.

Pirmą kartą paleidus Freenet programos klientą, vartotojui reikia pasirinkti saugumo lygio nustatymus (7.1 ir 7.2 pav.). Freenet sudaro trys saugumo lygiai:

- Žemas saugumo lygis – įsilaužėlis nesunkiai gali nustatyti norimo asmens tapatybę. Šis saugumo nustatymas mažiausiai įtakoja programos našumą.
- Vidutinis saugumo lygis – skirtas tiems vartotojams, kurie gyvena laisvoje ir demokratiškoje šalyje, kur įstatymai ir valdžia neriboja P2P programų veikimo. Toks saugumo lygis yra daug saugesnis negu tokių P2P programų kaip BitTorrent ar Gnutella, tačiau įsilaužėlis su vidutiniu lygio resursais ir žiniomis gali atsekti norimo vartotojo veiklą Freenet tinkle.
- Aukštas saugumo lygis – šis nustatymas naudingas tuo atveju, jei yra kitų draugų ar pažįstamų, kurie taip pat yra Freenet tinkle. Subūrus tam tikrą vartotojų skaičių į grupę, sukuriamas savas Freenet darknet tinklas.
- Maksimalus saugumo lygis – rekomenduojama naudoti tik tuo atveju, kai dirbama su informacija, dėl kurios vartotojas gali būti patrauktas baudžiamojon atsakomybėn. Kuo aukštesnis saugumo lygis, tuo mažesnis programos veikimo našumas.



7.2 pav. Freenet saugumo nustatymai.

Freenet programos kliente yra duomenų apsaugos nustatymai, jie reikalingi tam atvejui, jeigu kliento kompiuteris būtų pavogtas, arba pavogti patys duomenys. Freenet turi tris duomenų saugumo lygius:

- Žemas duomenų saugumo lygis – kompiuteryje saugomi duomenys visiškai nešifruojami. Programos veikimo sparta visiškai neįtakojama.
- Normalus duomenų saugumo lygis – visa svarbi informacija šifruojama.
- Aukštas duomenų saugumo lygis – šifruojama visa informacija, vartotojas pasirenka savo slaptažodį.
- Maksimalus duomenų saugumo lygis – viskas šifruojama, šifro raktas nelaikomas. Visa informacija ištrinama išjungus Freenet klientinę programą.

Freenet klientinės programos bandymas

Siekiant susipažinti su P2P programomis, užtikrinančiomis vartotojų anonimiškumą (arba bent jau tam tikrą anonimiškumo lygį) ir įvertinti jų naudojamo metodo efektyvumą buvo parsisiųstos ir praktiškai išbandytos dvi klientinės programos. Pirmoji bandoma programa Freenet. Aukščiau buvo apžvelgti jos naudojami metodai ir veikimas iš teorinės pusės. Dabar apžvelgsime programos instaliavimą, konfigūravimą ir naudojimą iš praktinės pusės.

1. Freenet klientinė programa buvo parsisiųsta iš *https://Freenetproject.org/* puslapio. Tai atvirojo kodo projektas, todėl programa yra visiškai nemokama. Norint pasileisti programinę įrangą, į kompiuterį privalo būti įdiegta naujausia JAVA vykdymo aplinkos veria, kadangi programa parašyta JAVA programavimo kalba. Freenet klientinės programos grafinė sąsaja pasileidžia interneto naršyklėje. Joje visą programos naudotojai ir atlieka visus veiksmus. Programos kūrėjai rekomenduoja naudoti Mozilla Firefox, Opera arba Google Chrome naršykles, kadangi Freenet klientinės programa išnaudoja šių naršyklių anoniminio naršymo funkciją.
2. Paleidus programą, reikia nustatyti pradinius saugumo nustatymus, būtinus programai veikti. Vėliau vartotojas gali juos keisti pagal savo poreikius. Yra keli nustatymų tipai: pradedantiesiems vartotojams ir pažengusiems vartotojams. Pagrindinis skirtumas tarp jų yra nustatymų skaičius (skiriasi apie keturiskart). Tik paleidus programą, ji bando jungtis prie Freenet tinklo. Vartotojams, kurie tinkle naudoja maršrutizatorių, norint, kad programa dirbtų sklandžiai reikia atsidaryti keletą portų (UDP Darknet port 22377 ir UDP Opennet port 25761). Atidarius portus daug greičiau surandami tinklo mazgai prie kurių galima jungtis. Kad programa pradėtų sėkmingai veikti, reikia, kad būtų prisijungta bent prie penkių išorinių mazgų.
3. Prieš pradedant naudotis programa, reikia nurodyti savo interneto pralaidumą, vietą, kuria leisime naudotis Freenet tinklui mūsų kompiuteryje, kadangi visa paviešinta informacija „klajoja“ po visą tinklą, pagal vartotojų poreikį ir duomenis, kuriuos patys norime padaryti prieinamus kitiems. Dar galima nurodyti katalogą, kuriame norime saugoti parsiųstus duomenis.
4. Atlikus programos nustatymą, galima pradėti ja naudotis. Freenet klientinė programa turi keletą paieškos vykdymo variantų: ieškoma puslapių (puslapis tai vieno vartotojo padaryti viešai prieinami duomenys, kiekvienas vartotojas turi po puslapį) pagal kategorijas arba ieškoma duomenų pagal paieškos frazę. Palyginus su paprastomis P2P duomenų apsikeitimo programomis, Freenet paieška labai lėta ir ganėtinai nefunkcionali. Pagal kategorijas ieškoti puslapių yra ganėtinai sudėtinga dėl puslapių kiekio ir lėto tų puslapių atidarinėjimo. Darant norimos informacijos paiešką pagal frazę, ji vyksta labai lėtai (greitis priklauso nuo randamos informacijos kiekio ir saugumo bei anonimiškumo nustatymų). Paieškos rezultatas pateikiamas ne failais, kurių pavadinimas sutapo su paieškos fraze, o puslapiais, kuriuose rasta failų, kurių pavadinimas sutapo su paieškos fraze. Tai yra ganėtinai didelis minusas, nes dar reikia atskirai naršyti po rastus puslapius, norint parsisiųsti norimą informaciją. Puslapių užkrovimo laikas labai priklauso nuo jų populiarumo ir nuo to, kaip jie toli nutolę Freenet tinkle. Labai dažnai jų net neužkrauna.

5. Reikiamos informacijos paieška, pati sudėtingiausia naudojimosi Freenet klientine programa dalis. Radus norimą informaciją, belieka ją atsisiųsti. Siuntimo greitis labiausiai priklauso nuo informacijos populiarumo ir paplitimo. Jei paplitimas mažas, siuntimas gali užtrukti. Siuntimo greičiui įtakos taip pat turi klientinės programos nustatymai.

Praktiškai išbandžius Freenet programą ir tinklą, galima teigti, kad tai yra labai funkcionali ir didelį tinklo vartotojų skaičių turinti programa. Ši programa plečiama nemokamais parsisiunčiamais įskiepiais, kurie vartotojams suteikia galimybę ne tik anonimiškai dalintis informacija, bet ir siųsti žinutes vieni kitiems arba elektroninį paštą ar rašyti blogą. Aišku Freenet neprilygsta savo funkcionalumu, vartotojų skaičiumi ar valdymo patogumu paprastoms P2P duomenų keitimosi programoms, tačiau augant vartotojų skaičiui, ji turėtų tobulėti.

7.2. Mute tinklo samprata ir bandymas

Šifravimas Mute tinkle

Prisijungimai prie kaimyninių mazgų Mute tinkle yra realizuoti naudojant saugių srautų technologijas:

- RSA vieši/privatūs raktai (jų dydį parenka vartotojas programos paleidimo metu) naudojami slaptųjų šifro raktų apsikeitimui.
- AES 128 bitų slaptieji raktai naudojami CFB režimu su nuliniais iniciacijos vektoriais, kurie skirti užšifruoti duomenų srautą.
- Atskiri AES raktai naudojami kiekvienai srauto krypčiai.
- AES raktai atnaujinami naujais kiekvieną kartą, kiekvienam naujam duomenų srautui.

Mute tinkle maršrutai nėra šifruojami nuo pradžios iki galo (tikslo). Dėl žmogus viduryje atakų grėsmės šifravimo raktų apsikeitimo metu, visiškai saugus nuo pradžios iki galo duomenų šifravimas yra neįmanomas jokiame tinkle.

Maršrutizavimo lentelės

Mute maršrutizavimo algoritmas tam, kad susekti kurie kaimyniniai prisijungimai yra susieti su tam tikru siuntėjo adresu, naudoja maršrutizavimo lenteles. Pavyzdžiui Mute vartotojas gauna keletą žinučių iš kito Mute vartotojo (pvz.: Alisos). Šios žinutės pasiekia vartotoją per tris iš galimų penkių kaimyninių mazgų. Kada vartotojas gaus Alisai adresuotus pranešimus, jis turėtų juos persiųsti per tuos kaimyninius mazgus, kurie jau kažkada yra siuntę vartotojo P2P klientui informaciją susijusią su Alisa. Atsižvelgiant į tai, kad vartotojas gali gauti žinutes iš Alisos per daugiau negu vieną kaimyninį

mazgą, įvairūs algoritmai gali būti panaudoti siekiant nuspręsti per kurį mazgą siųsti konkrečią žinutę.

Mute naudoja tikimybinį algoritmą maršruto parinkimui. Kiekvieną neseniai matytą siuntėjo adresą Mute deda į pastarųjų kaimyninių sujungimų rodyklių eilę, iš kurių vartotojas gavo žinutes iš to siuntėjo. Pavyzdžiui, vartotojas gali sekti paskutinius šimtą kaimyninių mazgų, kurie jam persiuntė žinutę iš Alisos. Jei vartotojas stebi tris siuntėjų adresus (Alisos, Bobo ir Evelinos) ir taip pat jis turi penkis kaimyninius mazgus, kurie sunumeruoti nuo 1 iki 5, maršrutizavimo lentelė turėtų atrodyti taip:

7.1 lentelė. Mute maršrutizavimo lentelės pavyzdys.

Nuo:	*Alisos*	*Bobo*	*Evelinos*
	1	*5*	*4*
	2	*1*	*4*
	2	*2*	*4*
	1	*1*	*3*
	4	*1*	*4*
	1	*5*	
	1	*5*	
	2		
	2		
	1		

Iš maršrutizavimo lentelės modelio galima matyti, jog dauguma žinučių iš Evelinos pasiekė vartotoją per ketvirtą kaimyninį mazgą. Galima spėti, kad žinutės atgal Evelinai grįš per tą patį ketvirtą kaimyninį mazgą.

Jei vartotojas gaus pranešimą adresuotą Alisai, jo Mute klientinė programa pasirinks Alisos stulpelį iš lentelės, atsitiktinai pasirinks vieną iš kaimyninių mazgų ir nukreips tos žinutės maršrutą per tą mazgą. Šioje schemoje teikiama pirmenybė teikia žinučių nukreipimui per „populiariausius“ kelius, bet kartais žinučių nukreipimai atliekami per nelabai populiarius kelius.

Paieška Mute tinkle

Mute failų dalinimosi tinklas naudoja paskirstytą paiešką, kuri naudoja valdomą potvynių (angl. *flooding*) algoritmą failams surasti pagal vardą remiantis laisvos formos užklausų vardų sritimis (angl. *string*). Tuo tarpu daugelyje kitų P2P tinklų potvyniai kontroliuojami TTL schemomis, tačiau Mute pristato daug veiksmingesnį ir keičiamo dydžio mechanizmą, kuris vadinasi: universalus skaitiklis (angl. *utility counter*).

Mute klientinė programa ir jos bandymas

Mute failų dalinimasis yra taško į tašką tinklas (P2P), kurio pagalba galima anonimiškai ieškoti ir siųsti įvairius failus apsaugant savo privatumą. Tinklo apsaugos principas yra vengti tiesioginių ryšių tarp siunčiančiųjų vienas iš kito. Mute tinkle norima informacija gaunama iš kaimyninių mazgų, kurie tą informaciją gaunas iš savo kaimyninių mazgų ir t.t.

Mute naudojimas

Parsiuntus programą Windows ar Mac platformoje, galima iškart ją suinstaliuoti. Unix vartotojams reiktų dar šią programą sukompiliuoti. Instaliavus programą atsiranda Mute aplankas, kuriame yra visi konfigūraciniai ir paleidimo failas. Skirtingai nuo kitų panašių programų, Mute tinklo klientinėje programoje nustatymai keičiami ne per vartotojo sąsają, o norimu teksto redaktoriumi redaguojant nustatymų failus.

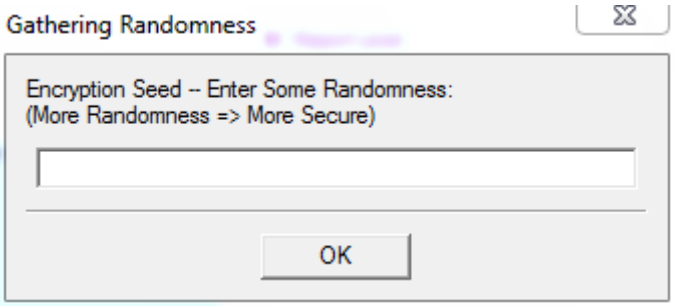
Tam kad Mute prisijungtų prie tinklo reikia nurodyti tam tikrus „hosts“ adresus. Juos įrašyti reikia į nustatymų (angl. *settings*) aplanke esantį failą – „webHostCaches“. Šiuose failuose saugomi aktyvių arba neseniai aktyviais buvusių tinklo mazgų IP adresai. Jie nurodomi taip:

http://mcache.northcountrynotes.org/mcache.php

http://reezer.freeshell.org/mcache/mcache.php

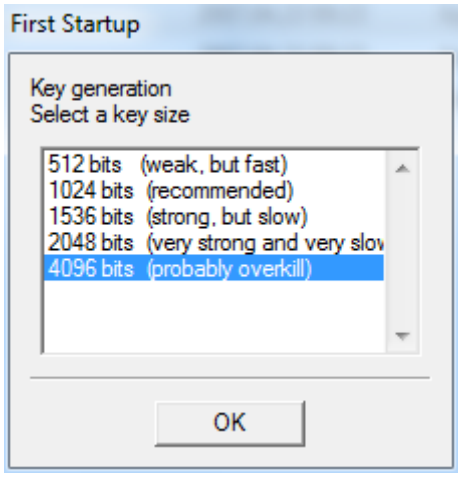
http://mcache.mccarragher.com/mcached/mcache.php

Išsaugojus failą paleidžiama Mute anonimiško failų keitimosi programa. Pirmą kartą paleidus programą klausiama ar naudojama ugniasienė (tokiu atveju geriausia ugniasienę sukonfigūruoti darbui su Mute, nes kitaip programa gali neprisijungti prie tinklo), tada prašoma įvesti eilė bet kokių simbolių (7.3 pav.), pagal kuriuos bus generuojamas šifro raktas (kuo ilgesnė atsitiktinių simbolių eilė, tuo saugesnis šifro raktas). Pagal poreikį, yra galimybė pasirinkti rakto bitų kiekį (7.4 pav.).



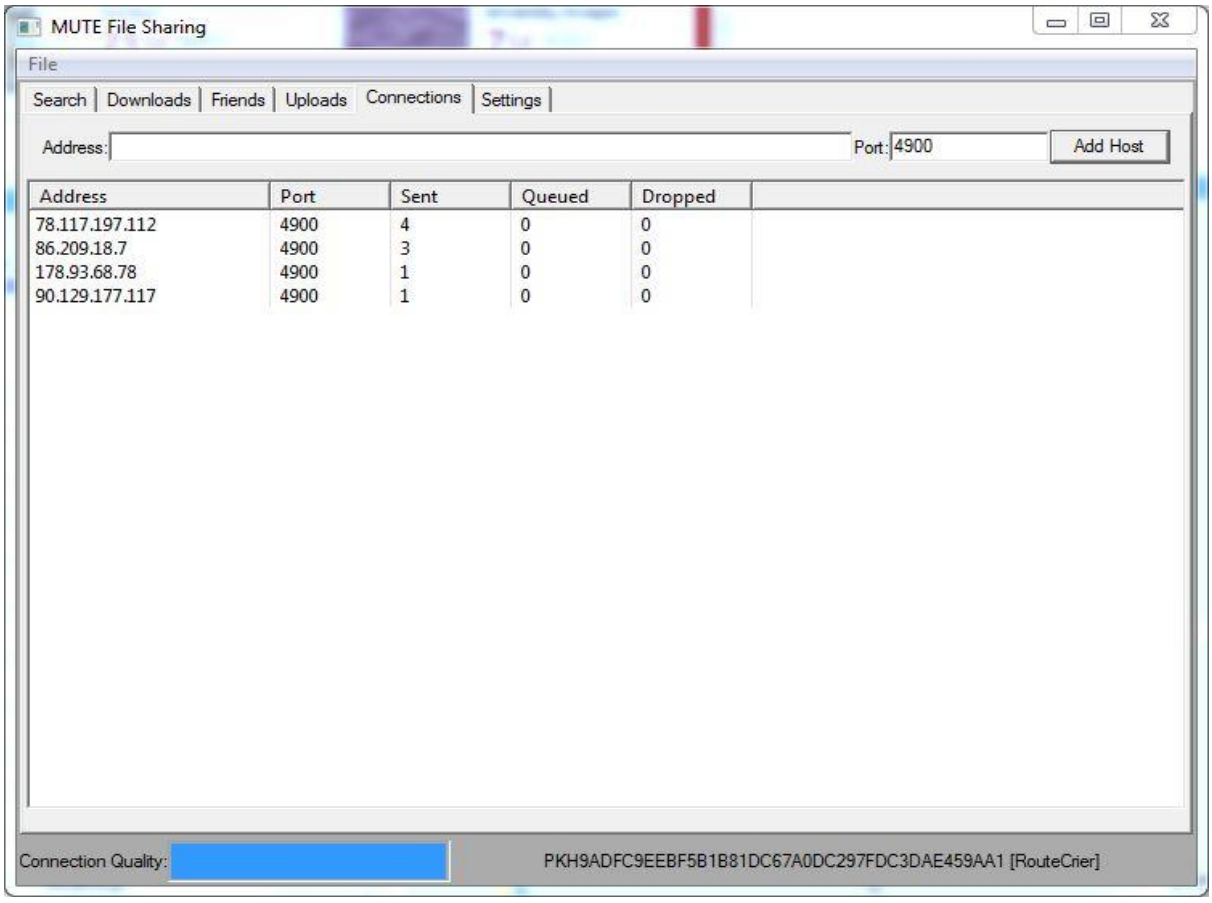
7.3 pav. Rakto generavimas.

Dažniausiai pakanka 1024 bitų ilgio rakto. Kuo ilgesnis raktas, tuo ilgiau jis generuojamas, tačiau tuo jis stipresnis. Sugeneravus raktą reikia pasirinkti aplanką, kuriuo nori dalintis su kitais Mute klientais.



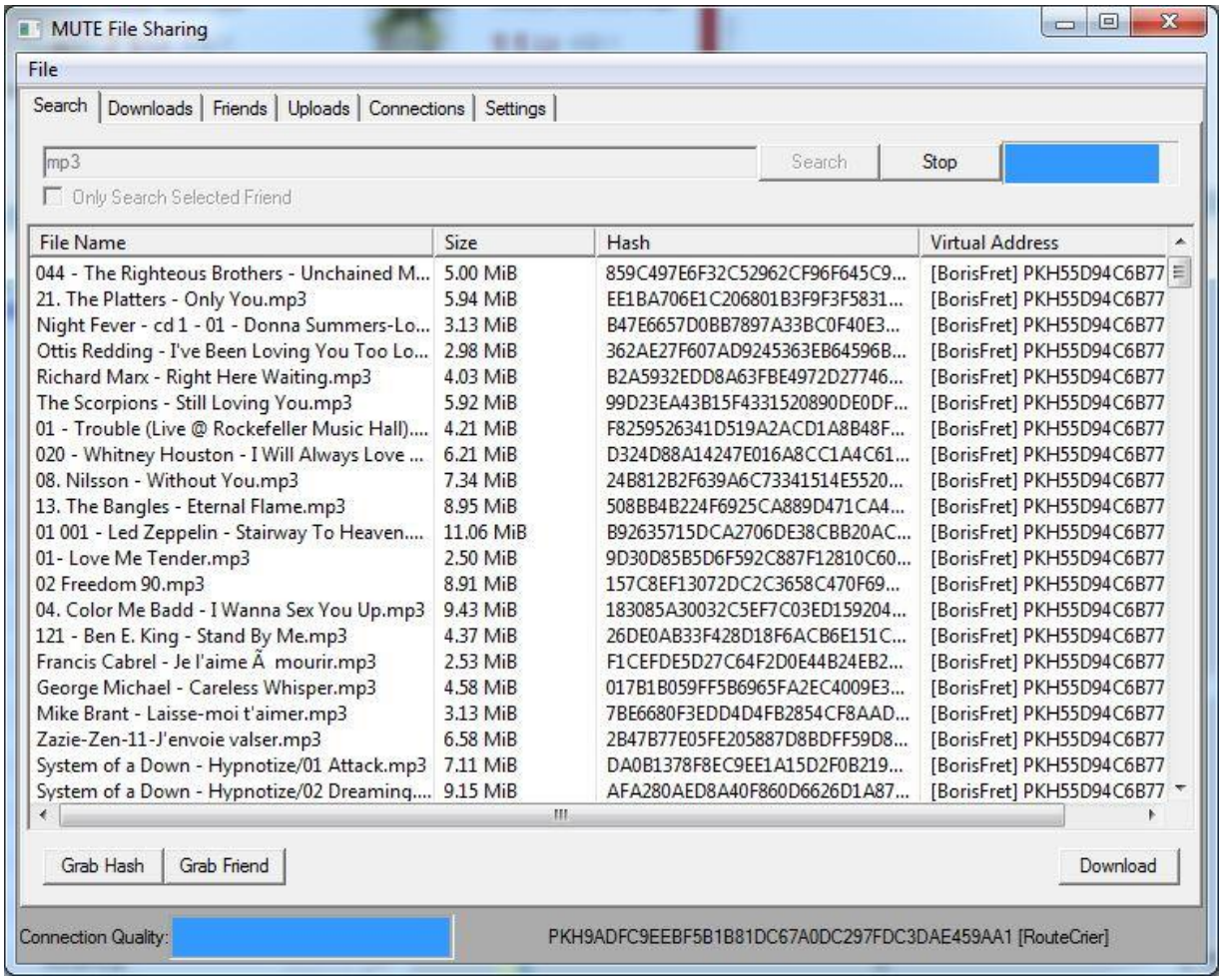
7.4 pav. Rakto ilgio pasirinkimas.

Kai Mute pagrindinis langas pasileidžia ir klientinė programa sėkmingai prisijungia prie kokio nors mazgo, „Connections“ kortelėje (7.5 pav.) galima matyti mazgus, prie kurių esame prisijungę. Lango apačioje rodo IP adresą, prie kurio bandoma prisijungti. Daugelis Mute vartotojų prie tinklo prisijungia trumpam, parsisiunčia norimą informaciją ir išjungia programą. Tačiau jų IP adresas į duomenų bazę įrašomas kaip aktyvus ir kurį laiką ten lieka, nors tas mazgas realiai nėra aktyvus. Todėl kol programa, ieškodama perbėga per visus adresus, prisijungimas prie aktyvaus mazgo gali užtrukti (kartais iki valandos), ypač kai vartotojas yra už ugniasienės.



7.5 pav. Prisijungimo kortelė.

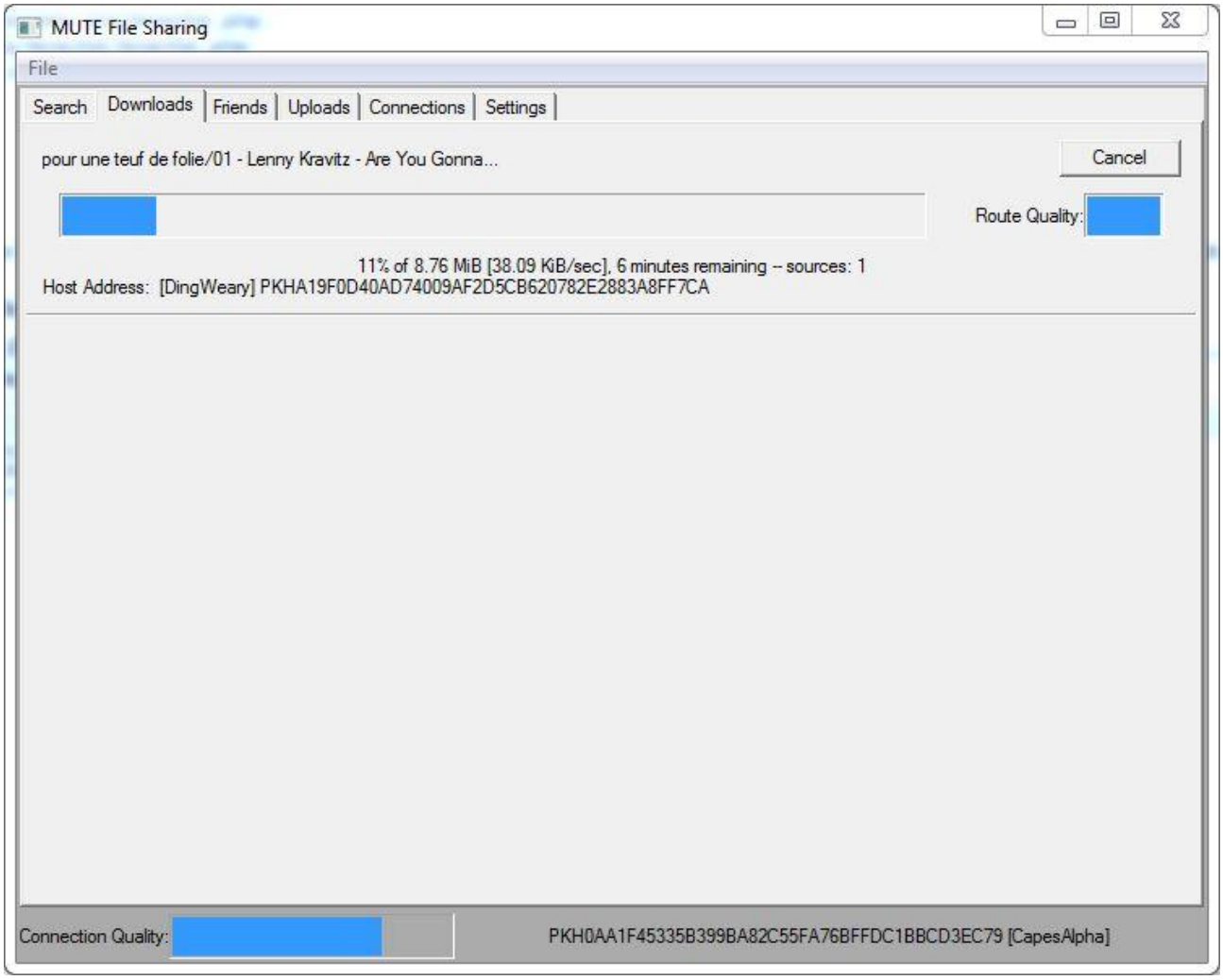
Atsiradus bent vienam prisijungimui „Connections“ sąraše, galima pereiti į „Search“ kortelę ir pradėti norimos informacijos paiešką, tačiau rekomenduojama, kad Mute klientinė programa būtų prisijungus bent prie penkių aktyvių mazgų. Tokių atveju išauga programos veikimo našumas, paieška grąžina daugiau rezultatų (didesnė tikimybė rasti norimą failą) bei siuntimas užtrunka mažiau laiko.



7.6 pav. Paieškos kortelė.

Mute paieška veikia tiesiogiai pagal simbolių seką. Pavyzdžiui ieškant „ion“ bus surasta nation.mp3 ir lions.mpg. Jei ieškoma pagal daugiau nei vieną žodį, surandami failai, kuriuose yra visi paieškoje įrašyti žodžiai. Kabutės ar žvaigždutė, kurias galima naudoti kitose paieškose sistemose, čia neveikia. Paieškos rezultatų lange prie rasto failo taip pat nurodomas to failo maišos funkcijos rezultatas ir virtualus tinklo mazgo, kuriame tas failas saugomas, adresas, kuris visiškai nesutampa su to mazgo fiziniu IP adresu).

Suradus norimą failą pasirenkame jį ir spaudžiame „Download“ mygtuką. Siuntimo statusą galima stebėti „Downloads“ kortelėje (7.7 pav.). Siuntimo greitis, palyginus su kitomis P2P programomis, yra žymiai lėtesnis. Tačiau ši programa užtikrina tam tikrą anonimiškumo ir privatumo lygį.



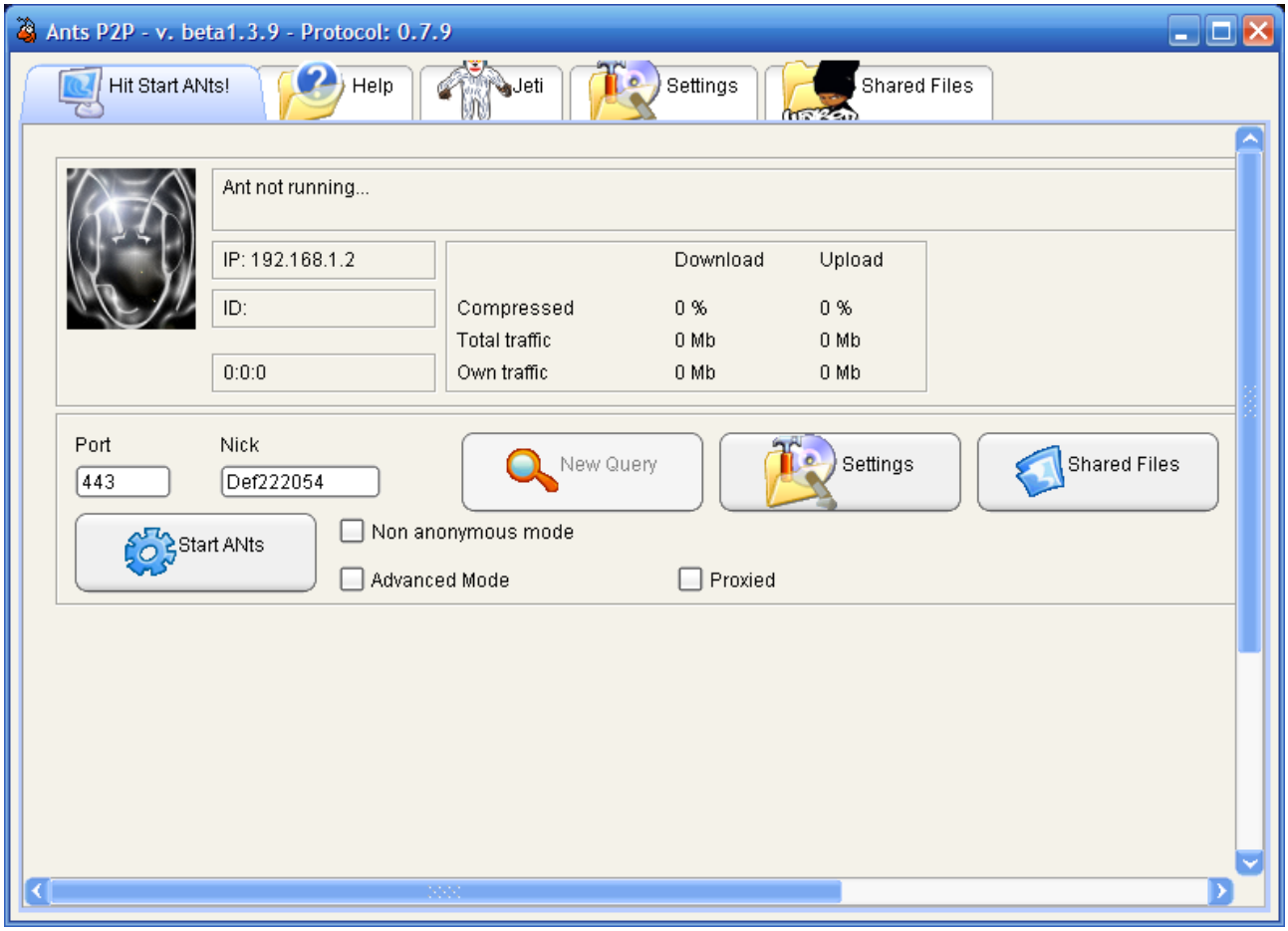
7.7 pav. Siuntimo kortelė.

Mute veikia puikiai per ugniasienę – jei ji prisijungia prie tinklo. Prisijungus, siuntimo greitis nesiskiria nuo mazgų be ugniasienės. Mazgams, kurie yra už ugniasienės yra sunkiau prisijungti dėl to, kad jie nepriima jokių prisijungimų iš išorės – ugniasienė jų nepraleidžia. Vienintelis būdas tokiems mazgams prisijungti – siųsti prisijungimo prašymus kitiems mazgams, kurie yra be ugniasienės. Taigi du mazgai, kurie yra už ugniasienės negali vienas su kitu susijungti. Mute tinklo dydis tiesiogiai priklauso nuo mazgų, kurie yra be ugniasienės.

Pabandžius dvi tam tikrą anonimiškumo lygį užtikrinančias programas (MUTE ir Freenet) galima, teigti, jog Freenet tinklas turi daug daugiau vartotojų (remiamasi tuo, kad per vienodą laiką Frenet programa prisijungė prie penkis kartus daugiau tinklo mazgų ir paieškos rezultatų kiekis buvo daug kartus didesnis negu MUTE), išstudijavus abiejų programų oficialius aprašymus, susidaro įspūdis, kad „Freenet yra saugesnis už Mute. Taip pat Freenet tinklo klientinė programa yra daug funkcionalesnė už Mute (įskiepių dėka Freenet klientinės programos funkcionalumą galima stipriai praplėsti). Mute yra pranašesnė už Freenet programą, nes turi paprastą, draugišką vartotojo sąsają bei paieškos greičiu.

Kitos anonimiškumą užtikrinančios programos

Be dviejų apžvelgtų programų, buvo dar bandoma, ANTs programa, kurios veikimo principas yra labai panašus į Mute (naudoja tą patį algoritmą). Tačiau jos išbandyti nepavyko, nes per dvylika valandų programa nerado nė vieno aktyvaus mazgo, prie kurio galėtų prisijungti (11 pav.). Taip yra dėl to, kad ANTs projektas yra jau ganėtinai senas, pati programa neišpopuliarėjo pakankamai, kad vartotojai atkreiptų į ją dėmesį, o P2P tinklų našumas ir efektyvumas yra tiesiogiai proporcingas jų vartotojų skaičiui. Tokių „beveik mirusių“ projektų kaip ANTs galima rasti ir daugiau.



7.8 pav. ANTs klientinės programos pradinis langas.