KAZIMIERAS BAGDONAS

DAUGIAMODALI SAUGUMO SISTEMA DAIKTŲ INTERNETO KOMUNIKACIJAI

DAKTARO DISERTACIJOS SANTRAUKA

TECHNOLOGIJOS MOKSLAI, INFORMATIKOS INŽINERIJA (T 007)

Kaunas
2022

KAUNO TECHNOLOGIJOS UNIVERSITETAS
VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

KAZIMIERAS BAGDONAS

DAUGIAMODALI SAUGUMO SISTEMA DAIKTŲ INTERNETO KOMUNIKACIJAI

Daktaro disertacijos santrauka
Technologijos mokslai, informatikos inžinerija (T 007)

2022, Kaunas

Disertacija rengta 2016–2021 metais Kauno technologijos universiteto Informatikos fakultete, Kompiuterių katedroje. Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu. Mokslinius tyrimus rėmė Lietuvos mokslo taryba.

Mokslinis vadovas:
Prof. dr. Algimantas VENČKAUSKAS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007)

Disertacijos santrauką redagavo: Aurelija Gražina Rukšaitė (Leidykla „Technologija“)

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:
prof. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007) — **pirmininkas**;
prof. dr. Nikolaj GORANIN (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
prof. dr. Egidijus KAZANAVIČIUS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
doc. dr. Simona RAMANAUSKAITĖ (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
dr. Anton RASSÕLKIN (Talino technikos universitetas, Estija, technologijos mokslai, informatikos inžinerija, T 007).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2022 m. rugsėjo 1 d. 14 val. Kauno technologijos universiteto „Santakos“ slėnio Posėdžių kambaryje A228.

Adresas: K. Baršausko g. 59, 51423 Kaunas, Lietuva.
Tel. (370) 37 30 00 42; el. paštas doktorantura@ktu.lt

Disertacijos santrauka išsiųsta 2022 m. rugpjūčio 1 d. Su disertacija galima susipažinti internetinėje svetainėje http://ktu.edu ir Kauno technologijos universiteto bibliotekoje (K. Donelaičio g. 20, 44239 Kaunas), bei internetinėje svetainėje https://vilniustech.lt ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, 10223 Vilnius).

KAUNAS UNIVERSITY OF TECHNOLOGY
VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

KAZIMIERAS BAGDONAS

MULTIMODAL SECURITY SYSTEM FOR INTERNET OF THINGS COMMUNICATIONS

Summary of Doctoral Dissertation
Technological Sciences, Informatics Engineering (T 007)

2022, Kaunas

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Informatics, Department of Computer Sciences during the period of 2016–2021. The studies were supported by Research Council of Lithuania. The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University.

Scientific supervisor:

Prof. dr. Algimantas VENČKAUSKAS (Kaunas University of Technology, Technology sciences, Informatics Engineering, T 007)

Dissertation summary editor: Aurelija Gražina Rukšaitė (Publishing House "Technologija")

Dissertation Defence Board of Informatics Engineering Science Field:
prof. dr. Rimantas BUTLERIS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007) — **chairperson**;
prof. dr. Nikolaj GORANIN (Vilniaus Gedimino technikos universitetas, Technological Sciences, Informatics Engineering, T 007);
prof. dr. Egidijus KAZANAVIČIUS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007);
doc. dr. Simona RAMANAUSKAITĖ (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, T 007);
dr. Anton RASSÕLKIN (Tallinn University of Technology, Estonia, Technological Sciences, Informatics Engineering, T 007).

The official defence of the dissertation will be held at 2 p.m. on 1 September, 2022 at the public meeting of Dissertation Defence Board of Informatics Engineering Science Field in the Meeting room A228 at Santaka Valley of Kaunas University of Technology.

Address: K. Baršausko st. 59, 51423 Kaunas, Lithuania
Tel. no. (+370) 37 30 00 42; e-mail doktorantura@ktu.lt

Summary of doctoral dissertation was sent on August 1st, 2022. The doctoral dissertation is available on the internet http://ktu.edu and at the library of Kaunas University of Technology (K. Donelaičio St. 20, 44239 Kaunas, Lithuania), and https://vilniustech.lt and at the library of Vilnius Gediminas Technical University Saulėtekio al. 14, 10223 Vilnius.

ABSTRACT

The emergence of Internet of Things (IoT) networks is promising to allow billions of devices to be connected in wireless *ad hoc* networks, thus providing various types of data to be generated, analyzed, and used, to create synergistic benefits that could not be obtained from isolated devices. The IoT architecture consists of low power and low computing devices and is further constrained by the limited bandwidth, thus cybersecurity solutions for IoT have to be designed and grounded with these limitations in mind. The fundamental cybersecurity tasks are object identification and authentication. More advanced cybersecurity possibilities and applications rely on these two being executed in the strongest possible way, which is often not applicable to IoT devices due to their limitations. An alternative to a single strong, but resource-intensive, the method is a multimodal solution that integrates two or more methods in conjunction to increase the overall security level.

In this thesis, we present a multimodal security system for IoT communication that shall integrate secure data streaming, a distributed software-based localization algorithm, and a power management control system for it, together with a novel multimodal localization data integration solution for object identification and authentication. Due to the limitations inherent in the IoT architecture, a set of methods have been developed to obtain the desired results. Round Trip Time (RTT) ranging measurements are used in a distributed localization algorithm for ad hoc networks that allows partial convergence of the network to generate localization solutions. A Hybrid Control System (HCS) is developed to manage power between communication, computation, and localization tasks, and it can function even with intermittent power sources, such as solar cells. A MAST-based multimodal identification and authentication method that can employ the geolocation solution of the End-Node, the geolocation information from the IoT Network Nodes (NN's) which provides measurements for the generation of said solution, and which combines it with asymmetric encryption. Finally, a lightweight secure streaming protocol for IoT is presented. The combination of these solutions creates a distributed software solution that could be deployed in IoT networks to obtain a multimodal security system for IoT objects.

I. ĮVADAS

1. Problemos formulavimas

Atsiradus ir paplitus kompiuteriams pramonėje ir asmeniniame žmonių gyvenime, šie įrenginiai tapo svarbūs daugeliu atžvilgių. Šiuo metu didelė dalis žmogaus veiklos – nuo asmeninių iki pramoninių ir net vyriausybinių funkcijų – yra papildyta ir patobulinta informacinėmis technologijomis, o kibernetinis saugumas pripažįstamas viena iš svarbiausių problemų. Atviras kompiuterių techninės ir programinės įrangos bei tinklų pobūdis kelia naujų ir didelių iššūkių, susijusių su ryšių ir duomenų tikrinimu ir patvirtinimu. Gebėjimas identifikuoti ir patvirtinti objektų autentiškumą yra pirmas ir esminis žingsnis siekiant užtikrinti IT infrastruktūros patikimumą. Atsiradus daiktų interneto (DI) tinklams, kuriuos turėtų sudaryti milijardai *ad hoc* tinkluose sujungtų prietaisų, objektų identifikavimo ir autentiškumo nustatymo sprendimai tampa sudėtingu uždaviniu. Daiktų interneto įrenginių pobūdis riboja turimą galią, skaičiavimo išteklius ir jutiklių, kuriuos galima realiai naudoti, diapazoną. Be to, išsklaidytos aparatinės įrangos architektūros ir konfigūracijos riboja bet kokius bandymus plačiai taikyti programinę įrangą, užuot rėmusis aparatinės įrangos sprendimais.

Naudojant vieno būdo identifikavimo ir autentiškumo nustatymo metodą galima suklastoti, todėl kyla pavojus saugumui ir duomenų vientisumui. Galimybė naudoti daugiamodalų objekto identifikavimo ir autentiškumo nustatymo metodą gali sumažinti šią riziką. DI atveju, kai tikimasi, kad prietaisų skaičius sieks milijardus, rankinis patikrinimas būtų praktiškai neįmanomas, todėl smarkiai sumažėtų įdiegtų sistemų patikimumas. Be to, galima tikėtis, kad konkretūs prietaisai savo funkcijas atliks konkrečioje geografinėje vietoje. Taigi duomenų tikrinimui gali kilti pavojus dėl minėtų prietaisų perkėlimo už numatomos veikimo teritorijos ribų. Šiuo metu nėra sukurto metodo, kuriuo būtų galima nustatyti daiktų interneto įrenginių daugiamodalių objektų identifikavimą ir autentiškumo nustatymą pagal buvimo vietą.

2. Darbo aktualumas

Sukurti *ad hoc* daiktų interneto tinklus, sudarytus iš milijonų ar milijardų prietaisų, kurie galėtų užtikrinti duomenų vientisumą ir patikrą, yra sudėtinga užduotis. Ne tik duomenys gali būti suklastoti, bet ir tinklo mazgai gali būti perkelti dėl išorės jėgų, todėl jų matavimai geriausiu atveju pasensta, o blogiausiu – sukelia kenkėjišką poveikį. Sukurti sistemą, galinčią generuoti geolokacinį sprendimą, visada buvo svarbus uždavinys. Tokią funkciją gali suteikti pasaulinės palydovinės navigacijos sistemos, tačiau jos integravimas gali būti neįgyvendinamas dėl daiktų interneto mazgų vieneto kainos, energijos ar skaičiavimo galios apribojimų.

Plačiai paplitusi geolokacijos paslauga ryšio kanalais naudojant standartinę įrangą dar nepasiekta ne tik dėl apribojimų, kuriuos lemia daiktų interneto įranga, bet ir dėl to, kad nėra universalių vietos nustatymo sprendimų. Galimybė integruoti identifikavimo ir autentifikavimo metodus su vietos nustatymo informacija gali gerokai padidinti duomenų patikimumą ir saugumą daiktų interneto tinkluose.

3. Darbo objektas

Šio tyrimo objektas – kibernetinio saugumo sprendimas, skirtas *ad hoc* daiktų interneto tinkluose esantiems įrenginiams.

4. Tezės tikslas

Pagrindinis šio darbo tikslas – pasiūlyti daugiamodalų daiktų interneto objektų identifikavimo ir autentiškumo nustatymo metodą, kuris sujungia lokalizaciją, duomenų srautus ir yra saugus, mažos galios ir mažo pralaidumo sprendimas.

5. Darbo uždaviniai

Pagrindiniai šios disertacijos uždaviniai:

1. Ištirti galimus belaidžio ryšio lokalizavimo metodus, pasiūlyti ir įvertinti paskirstytą *ad hoc* lokalizavimo metodą daiktų interneto tinklui.
2. Ištirti esamus valdymo metodus ir įvertinti jų tinkamumą daiktų interneto taikymams, pasiūlyti ir įvertinti daiktų interneto įrenginių galios valdymo valdymo sistemą, susijusią su lokalizacija belaidžiu ryšiu.
3. Ištirti kriptografinius metodus ir pasiūlyti lengvą daugiamodalų daiktų interneto objektų identifikavimo ir autentiškumo nustatymo sprendimą, kuriame būtų integruota lokalizacijos informacija.
4. Ištirti daiktų interneto komunikacijos metodus ir pasiūlyti saugų duomenų protokolą, skirtą duomenų srautui iš daiktų interneto tinklo mazgo.
5. Integruoti ir eksperimentiškai įvertinti pasiūlytus metodus.

6. Tyrimo metodologija

Disertacijos tikslams pasiekti taikyta ši tyrimo metodika.

1. Lyginamoji mokslinės literatūros analizė buvo naudojama lokalizavimo metodams, valdymo sistemoms, kriptografiniams metodams ir ryšių protokolams vertinti.

2. Kiekybiniai tyrimai buvo naudojami kuriant ir vertinant siūlomus lokalizavimo, kontrolės, identifikavimo ir autentiškumo nustatymo bei saugaus ryšio metodus.

3. Taikant analitinius tyrimus buvo vertinami lokalizavimo, kontrolės, identifikavimo ir autentiškumo nustatymo bei saugaus ryšio metodai.

4. Taikomieji tyrimai buvo naudojami siūlomiems įvertinti vietos nustatymo, valdymo, identifikavimo ir autentiškumo nustatymo bei saugaus ryšio metodams patvirtinti.

7. Mokslinė naujovė

Mokslinį naujumą ir šią disertaciją galima apibendrinti taip:

1. Pasiūlytas naujas paskirstytas lokalizavimo metodas, skirtas daiktų interneto tinklams.

2. Sukurta adaptyvi hibridinė valdymo sistema, skirta daiktų interneto tinklo mazgų belaidžio lokalizavimo metodų galiai valdyti.

3. Pasiūlytas naujas daugiamodalus daiktų interneto įrenginių identifikavimo ir autentifikavimo metodas.

4. Pasiūlytas saugus ir lengvas ryšio protokolas saugiam duomenų srautui, skirtas daiktų interneto taikomosioms programoms.

8. Praktinė vertė

Siūlomas lokalizavimu pagrįstas daugiamodalus daiktų interneto objektų identifikavimo ir autentiškumo nustatymo metodas įrodo, kad:

1. Programine įranga pagrįstos lokalizacijos sistemos integravimas į daiktų interneto tinklus suteikia visur esantį objektų identifikavimo ir autentiškumo nustatymo būdą;

2. Adaptyvios hibridinės valdymo sistemos taikymas daiktų interneto mazgams užtikrina stabilią lokalizaciją ir ryšio funkciją;

3. Lengvo saugaus srautinio duomenų perdavimo protokolo ir maišos pagrindu sukurtų duomenų struktūrų taikymas leidžia sukurti veiksmingą daugiamodalų daiktų interneto objektų identifikavimą ir autentifikavimą.

9. Ginamieji teiginiai

1. Siūlomas paskirstytas lokalizacijos algoritmas yra tinkamas būdas įgalinti ir skleisti lokalizacijos sprendimą *ad hoc* tinkluose.

2. Siūloma hibridinė valdymo sistema (HCS), skirta daiktų interneto įrenginių galimybėms, leidžia valdyti nuotolio matavimus, pagrįstus supakuotu apskriejimo laiku (RTT), paremtu vietos nustatymu per ryšio kanalus.

3. Siūlomas lengvas saugaus duomenų srauto protokolas padidina *ad hoc* daiktų interneto tinklų saugumą, taikant nebrangius metodus.

4. Kelių saugumo parametrų derinys pagerina identifikavimo ir autentiškumo nustatymo funkciją, nes įveda geolokacinę informaciją, kuri gali būti pritaikyta pagal pageidaujamą jautrumą.

5. Šių metodų sujungimas į daugiamodalų daiktų interneto įrenginių identifikavimo ir autentifikavimo metodą suteikia naujų funkcijų ir padidina daiktų interneto kibernetinio saugumo lygį.

10. Mokslinis patvirtinimas

Visi disertacijoje pateikti rezultatai yra originalūs ir atitinka dvi tarptautiniu mastu referuojamas „ISI Web of Science“ mokslinių žurnalų publikacijas.

Eksperimentų rezultatai buvo pristatyti ir aptarti trijose tarptautinėse konferencijose:

1. Bagdonas, Kazimieras; Jusas, Nerijus; Venčkauskas, Algimantas. "A converging distributed positioning algorithm for Internet-of-things", Elektronika ir elektrotechnika. Kaunas : KTU. ISSN 1392-1215. eISSN 2029-5731. 2017, Vol. 23, iss. 6, p. 72-76. [Science Citation Index Expanded (Web of Science); Scopus; Computers & Applied Sciences Complete] Q3 (2017, Scopus Sources)]

2. Venčkauskas, Algimantas; Morkevičius, Nerijus; Bagdonas, Kazimieras; Damaševičius, Robertas; Maskeliūnas, Rytis. "A lightweight protocol for secure video streaming", Sensors. Basel : MDPI AG. ISSN 1424- 8220. eISSN 1424-8220. 2018, vol. 18, iss. 5, art. no. 1554, p. 1-14. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] Q1 (2018, Scopus Sources)]

3. Bagdonas, Kazimieras; Venčkauskas, Algimantas. "Localization algorithm for identification of mobile objects in an Ad-Hoc internet of things network" 11th international workshop on data analysis methods for software systems,

Druskininkai, Lithuania, November 28-30, 2019 / Lithuanian Computer Society, Vilnius University Institute of Data Science and Digital Technologies, Lithuanian Academy of Sciences. Vilnius : Vilnius University, 2019. ISBN 9786090703243. eISBN 9786090703250. p. 8.

4. Bagdonas, Kazimieras.; Venčkauskas, Algimantas. "Identification of dynamic parameters and velocity control of a moving IoT node using a single ranging measurement source" 10th international workshop on data analysis methods for software systems, Druskininkai, Lithuania, November 29 – December 1, 2018. Vilnius : Vilnius University press, 2018. ISBN 9786090700433. p. 9.

5. Bagdonas, Kazimieras; Venčkauskas, Algimantas. "IoT mobile network Node's velocity estimation via curve fitting" 9th International workshop on data analysis methods for software systems, DAMSS : Druskininkai, Lithuania, November 30 – December 2, 2017 / Lithuanian Computer Society, Vilnius University, Institute of Data Science and Digital Technologies, Lithuanian Academy of Sciences. Vilnius : Vilnius University, 2017. ISBN 9789986680642. p. 6. DOI: 10.15388/DAMSS.2017.

II. DAUGIAMODALAUS SAUGUMO METODO KŪRIMAS

1. Visuotinė lokalizacija

Kad sukurtume universalų daiktų interneto *ad hoc* lokalizacijos algoritmą, tegul mūsų tinklą sudaro trijų tipų tinklo mazgai, galintys atlikti nuotolio matavimus, palaikyti lokalizacijos ryšį ir saugoti vietos bei absoliučios padėties nustatymo sprendimus. Šiame skyriuje pateiktas lokalizavimo metodas buvo paskelbtas ir žurnalo publikacijoje (Bagdonas ir kt., 2017) ir yra pagrįstas ankstesniais darbais, pateiktais tarptautinėse konferencijose (Bagdonas ir kt., 2009) ir (Bagdonas ir kt., 2008).

- Tinklo mazgas (NN) yra stacionarus daiktų interneto mazgas. Jis gali siųsti ir gauti duomenis bei atlikti RTT matavimus su kitais mazgais, esančiais jo ryšio diapazone.
- Vartotojo mazgas (VUN) yra mobilusis daiktų interneto mazgas, kuris gali judėti bet kuria kryptimi, gali siųsti ir gauti duomenis bei atlikti RTT matavimus su kitais mazgais, esančiais jo ryšio diapazone.
- Vietinė atskaitos sistema (LRF) yra mažiausiai 4 tarpusavyje sujungtų NN rinkinys. Vienas iš trijų NN negali būti tiesėje, kurioje yra kiti du, o ketvirtasis NN turi būti už plokštumos, apibrėžtos ankstesnių NN padėtimis, ribų. Tokia NN jungtis sudaro trimatį tetraedrą.

- Absoliutus padėties nustatymo sprendimas (APS) – tai padėties nustatymo sprendimas, nepriklausomas nuo LRF daiktų interneto tinkle. APS gali būti gaunamas tiesiogiai iš NN, kai yra teikiamas GNSS, jei NN turi atitinkamas technologines galimybes, arba netiesiogiai, nustatant UN ir įtraukiant UN APS koordinates į daiktų interneto vietos nustatymo algoritmą.

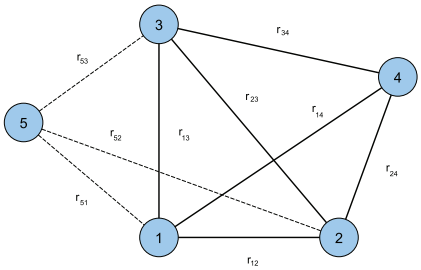
- Virtualus tinklo mazgas (VNN) – tai UN su šiuo nauju sprendimu, gautas atlikus nuotolio matavimus su vienu ar daugiau LRF. Tokiame VNN gali būti APS koordinatės, kurias pateikia nepriklausoma UN funkcija. VNN naudojamas susieti dviejų ar daugiau persidengiančių LRF įėjimo vėlyvojo LRF įsakymą su APS

- Tinklo konvergencija (NC) – tai būsena, kai du ar daugiau persidengiančių LRF sujungiami ir sukuriamas APS sprendimas. Toks NC gali įvykti vietoje arba būti viso LRF potinklio įvykis, kai LRF NN koordinatės pakeičiamos jų APS koordinatėmis. Pasiekus NC, DI padėties nustatymo sprendimas virsta APS papildymo / pakeitimo sprendimu ir yra visiškai suderinamas su APS sistema.

Kad būtų gautas trimatis sprendimas, mažiausiai keturi DI NN turi turėti tiesioginį kontaktą vienas su kitu, kad sudarytų LRF. Norint inicijuoti lokalizacijos algoritmą, kiekvienas NN turi palaikyti ryšį su visais turimais NN. Atliekami RTT matavimai ir gaunamas nuotolio įvertinimas. Toks matavimų rinkimas duoda dvejopus rezultatus abiejuose ryšio ir nuotolio matavimų galuose. Taigi gautais matavimais keičiamasi poromis, o gautą vidutinę vertę naudoja abu NN. Kiekvienas atskiras NN sukuria savo LRF ir yra žymimas kaip nulinis taškas NN_0 su koordinatėmis $x_0 = 0$, $y_0 = 0$ ir $z_0 = 0$. Atstumo matavimai tarp LRF esančių NN apibrėžiami kaip ρ_{ij}, kur i ir j yra konkrečios LRF esančių NN indeksai, kur indeksas i reiškia pradinį mazgą, o indeksas j – prijungtą mazgą RTT matavimui. Kiekvienas NN gali priklausyti ir paprastai priklauso kelioms LRF, o jo atskaitos indeksai kiekvienoje atskiroje LRF yra skirtingi, kaip parodyta 1 pav.

Gavus kiekvieno atskiro NN nuotolio matavimus ir užpildžius duomenų bazes vidutiniais nuotolio įverčiais, šios duomenų bazės dalijamos tarp NN. Po šio etapo kiekviena atskira pastaba sugretina gautus matavimus ir atlieka paiešką esamoje duomenų bazėje, kad nustatytų tokius jungiamuosius NN, iš kurių galima sudaryti bent vieną tetraedrą LRF.

Atlikus paiešką, visi dublikatai pašalinami, kad duomenų bazėse būtų tik unikalūs tetraedrai. Kiekvienas tetraedras atsiranda pagrindiniame mazge taip, kad jam priskirtos koordinatės būtų $x_0 = 0$, $y_0 = 0$ ir $z_0 = 0$. Kiekvienam LRF taikomas trilateracijos algoritmas. Pirmajam kaimyniniam NN priskiriamos tokios koordinatės, kad jis būtų ant LRF ašies X: $x_1 = \rho_1$,$y_1 = 0$,$z_1 = 0$. Antrojo ir trečiojo kaimyninių NN koordinatės apskaičiuojamos pagal lygtis nuo 1 iki 6.



1 pav. Tetraedrų formavimas NN 1

$$x_2 = \frac{\rho_{12}^2 - \rho_{01}^2}{-2\rho_{01}}, \tag{1}$$

$$y_2 = \sqrt{\rho_{02}^2 - x_2^2}, \tag{2}$$

$$z_2 = 0, \tag{3}$$

$$x_3 = \frac{\rho_{03}^2 - \rho_{13}^2 + x_2^2}{-2x_2}, \tag{4}$$

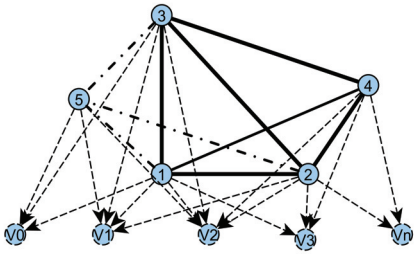
$$y_3 = \frac{\rho_{13}^2 - \rho_{23}^2 + x_2^2 + z_2^2}{2y_2} - \frac{x_2 * x_3}{y_2}, \tag{5}$$

$$z_3 = \sqrt{\rho_{03}^2 - x_4^2 - y_4^2}. \tag{6}$$

Apskaičiavus visų tetraedro NN koordinates, jos sujungiamos į LRF, apskaičiuojant dviejų tetraedrų pasukimo matricas ir konvertuojant koordinates į vieningą LRF, sudarytą iš visų unikalių tetraedrų, kuriuos priimantysis NN galėjo sudaryti su kaimyniniais NN. Apskaičiuojant pasukimo matricas taikomas linijinis mažiausių kvadratų (LLS) algoritmas, nes dėl triukšmingų matavimų negalima taikyti analitinio modelio. Nepertraukiamas padėties nustatymo sprendimas tarp nesujungtų LRF gaunamas dėl to, kad kaimyninės LRF gerokai persidengia. UN patekus į tokią persidengimo zoną, jos padėtis lygiagrečiai apskaičiuojama abiejose LRF. UN perėjimo per LRF metu sukuriama VNN_i aibė, kur $i = 0 : n$.

Pozicijos nustatymas konkrečiame mazge

Kai vartotojo mazgas (UN) patenka į teritoriją, kurioje jis gali gauti bent 3 nuotolio matavimus į 3 skirtingus LRF NN, galima atlikti trilateracijos algoritmą ir gauti jo santykinės padėties LRF viduje sprendimą.



2 pav. VNN stebėjimas per jo trajektoriją

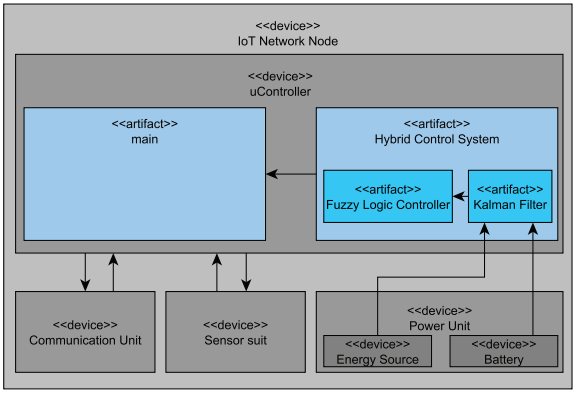
Jei UN turi nepriklausomą padėties nustatymo sprendimą, jo rezultatus galima perkelti į LRF. Tada kiekvienas atskiras matavimas traktuojamas kaip VNN, kaip parodyta 2 pav., kur V_0 reiškia įvertintą UN padėtį matavimo metu t_0, V_1 reiškia įvertintą padėtį matavimo metu t_1 ir t.t. Surinkus pakankamą kiekį matavimų, LRF galima perkelti į absoliučią koordinačių sistemą, nes kiekvienas jos NN yra sujungtas su bent trimis kitais NN. Tuomet LRF laikoma visiškai konverguota. Padėties nustatymo sprendinio kokybę galima toliau didinti papildomais matavimais ir aukštesnės geometrinės kokybės stebėjimais.

Tinklo konvergencija

Kai LRF gauna savo mazgų padėties nustatymo sprendimą absoliutinėje padėties nustatymo sistemoje, sprendimas gali būti perkeltas į kaimynines LRF. Tokią migraciją galima pasiekti, nes kaimyniniai LRF turi bendrą NN skaičių. Jei bent 3 LRF NN gauna koordinates absoliutinėje padėties nustatymo sistemoje, perėjimą iš vietinių koordinačių į absoliutines galima gauti naudojant pasukimo matricą, apskaičiuojant LRF pasukimą aplink savo pradžią. Taikant tokį metodą išvengiama drastiškų klaidų kaupimosi spąstų, kai LRF sujungiamos tik su santykinėmis padėtimis, gautomis iš nuotolio įverčių.

2. Hibridinė valdymo sistema

Daiktų interneto lokalizavimo užduočiai atlikti siūloma ši hibridinė valdymo sistema. Kalmano filtras yra skirtas daiktų interneto įrenginių energijos suvartojimui valdyti. Jis priima duomenis iš maitinimo šaltinio modulio kaip likusią baterijos galią, įėjimo galią iš atsinaujinančio energijos šaltinio ir galios suvartojimo lygį iš μKontrolerio, kaip parodyta 3 pav. Išėjimas pateikiamas neraiškiosios logikos valdikliui (FLC) integruoti, kuris nusprendžia, su kiek tinklo mazgų palaikyti ryšį ir kokiu dažniu imti mėginius. Toliau apibendrinami hibridinės valdymo



3 pav. Pasiūlyta hibridinė kontrolės sistema

sistemos elementai.

1. Kalmano filtras – daiktų interneto įrenginių energijos suvartojimui valdyti
2. Neaiški logika – tinklo mazgų, su kuriais atliekami diapazono matavimai, skaičiui nustatyti, atsižvelgiant į turimą galią ir kontroliuojant diapazono matavimų nuokrypių lygius

Kalmano filtro projektavimas

Kalmano filtras aprašomas sistemos dinaminiu modeliu, kurį sudaro visi būsenos vektoriai X_t ir būsenos perėjimo matrica A, ir matavimo modeliu, kurį sudaro stebėjimo vektorius u_t ir stebėjimo perėjimo matrica H. Kadangi Kalmanas yra tiesinis įvertis, būsenų perėjimo lygtys turi būti tiesinės. Netiesines lygtis galima pakeisti supaprastintu modeliu arba taikyti pažangesnį įvertį, pavyzdžiui, išplėstinį Kalmano filtrą. Išsamų Kalmano filtrų paaiškinimą galima rasti visoje literatūroje, tačiau šiame darbe remiamasi valdymo teorijos knyga (Grewal ir kt., 2014).

Mūsų atveju užduotis – valdyti daiktų interneto NN energijos suvartojimą taip, kad būtų galima užtikrinti nepertraukiamą veikimą. Darome prielaidą, kad mūsų prietaisas gali tiesiogiai matuoti akumuliatoriaus energijos lygį ir saulės energijos šaltinio tiekiamą energiją. Dėl saulės energijos generavimo pobūdžio, kuris priklauso nuo prietaiso geografinės padėties Žemės paviršiuje, metų ir paros laiko, lygtis, apibūdinanti didžiausią Saulės energijos generatoriaus generuojamą galią, nėra tolygi ir netiesinė. Siekiant supaprastinti daiktų interneto prietaiso

skaičiavimo naštą, ši netiesinė lygtis paverčiama tiesinių lygčių rinkiniu. Modeliuojamas daiktų interneto NN turi pastovų energijos suvartojimą (angl. *Constant Power Consumption*, CPC), kuris būtinas jo veikimui palaikyti. Darome prielaidą, kad CPC yra pastovus.

Būsenų lygtys ir būsenų vektorius

Būklės vektorių apibrėžia 7 lygtis:

$$x_t = \begin{bmatrix} PAL_t & BPL_t & IP_t & \delta BPL & \delta IP & CPC \end{bmatrix}, \tag{7}$$

čia

PAL_t – lokalizacijai skirta galia,

BPL_t – baterijos galios lygis,

IP_t – įvesties galia,

δBPL_t – baterijos galios lygio pokytis,

δIP – įvesties galios lygio pokytis,

CPC – pastovus energijos suvartojimas.

PAL_t yra galios kiekis, kurį filtras skiria lokalizavimo užduočiai atlikti, ir jis paskirstomas mazgų, su kuriais atliekama lokalizavimo užduotis, skaičiui. BPL_t yra visa baterijoje saugoma turima energija. Kadangi NN turi veikti neribotą laiką, KF turi įvertinti turimą energiją, atsižvelgdamas į bendrą energijos suvartojimą ir saulės energijos generatoriaus tiekiamą energiją. IP_t yra saulės energijos generatoriaus tiekiama galia, kurią tiesiogiai matuoja NN. δBPL_t – apskaičiuotas akumuliatoriaus galios lygio pokytis. δIP – numatomas saulės energijos generatoriaus generuojamos galios pokytis. 8, 9, 10, 11, 12 pateiktos būklės perėjimo lygtys:

$$PAL_{t+1} = IP_t - CPD_t - dBPL_t, \tag{8}$$
$$BPL_{t+1} = BPL_t + \delta BPL_t, \tag{9}$$
$$IP_{t+1} = IP_t + \delta IP_t, \tag{10}$$
$$\delta BPL_{t+1} = \delta BPL_t, \tag{11}$$
$$\delta IP_t = SMPR * \delta IP_t, \tag{12}$$
$$CPD_{t+1} = CPD_t. \tag{13}$$

Naudodamiesi perėjimo lygtimis, galime sukurti tokią matricą A, kad lygtis 15 būtų teisinga, kur x_{t+1} yra jungties įvertis laiko momentu $t+1$, A yra ši perėjimo matrica, o x_t yra būsenos vektorius laiko momentu t.

Būklės perėjimo matrica:

$$A = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & -1 & \\ 0 & 1 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$x_{t+1} = A \times x_t. \tag{15}$$

3. Matavimų vektorius ir matavimo jautrumo matrica

Matavimo vektorius 16 tiesiogiai matuoja akumuliatoriaus galios lygį BPL_t, saulės energijos generatorių įvestį IP_t ir skirtumą tarp vidinių energijos suvartojimo modelių ir matavimų metu nustatytų verčių. δIP_t rodo skirtumą tarp tiesinio saulės energijos generavimo modelio, kuris yra apibrėžtas tiesiniu modeliu, aprašytu saulėlydžio laiko, saulėtekio laiko ir vidurdienį generuojamo laiko parametrais. Šie parametrai priklauso nuo metų laiko ir NN vietos planetos paviršiuje. δBPL_t apibūdina skirtumą tarp numatomų žarnų atsargų baterijoje ir vertės, gautos atlikus tiesioginius matavimus.

Pagal šį aprašymą galime užrašyti matavimo jautrumo matricą H 17, kuri užtikrina ryšį tarp matavimo vektoriaus ir nurodyto vektoriaus, kaip apibrėžta lygtyje 18:

$$u_t = \begin{bmatrix} BPL_t & IP_t & \delta IP_t & \delta BPL_t \end{bmatrix}, \tag{16}$$

$$H = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 1 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{17}$$

$$u_t = H \times x(t). \tag{18}$$

δBPL_t modeliuojama tiesiškai, kad būtų pradedama nuo tam tikro galios lygio ir saulėtekio metu pasiektų 4% galios. Po to δBPL_t tiesiškai modeliuojama taip, kad saulėlydžio metu galia padidėtų iki didžiausios galios, o vidurnaktį tiesiškai sumažėtų iki 50%, kaip aprašyta lygtimis 19, 20, 21:

$$\delta BPL_t = \frac{BPL_{sr} - BPL_{in}}{t_{sr} - t_{st}}, t \in t < t_{sr}, \tag{19}$$

$$\delta BPL_t = \frac{BPL_{ss} - BPL_{sr}}{t_{ss} - t_{sr}}, t \in t_{ss} < t < t_{sr}, \tag{20}$$

$$\delta BPL_t = \frac{BPL_{mn} - BPL_{ss}}{t_{mn} - t_{ss}}, t \in t > t_{ss}. \tag{21}$$

čia BPL_in – akumuliatoriaus galios lygis dienos pradžioje, BPL_{sr} – akumuliatoriaus galios lygis saulėtekio metu, BPL_{ss} – akumuliatoriaus galios lygis saulėlydžio metu, BPL_{mn} – akumuliatoriaus galios lygis vidurnaktį, t_{st} – modeliavimo pradžia, t_{sr} – saulėtekio laikas, priklausantis nuo metų dienos ir NN vietos, t_{ss} – saulėlydžio laikas, priklausantis nuo metų dienos ir NN vietos, t_{mn} – vidurnaktis.

δIP_t įjungtos galios pokyčio tiesinį modelį apibrėžia lygtys 22, 23, 24, 25. Pagal netiesinį modelį įvestos galios vertės nustatomos saulėtekio, vidurdienio ir saulėlydžio momentais ir apskaičiuojamos tiesinės trajektorijos tarp šių taškų:

$$\delta IP_t = 0, t \in t < t_{sr}, \tag{22}$$

$$\delta IP_t = \frac{IP_{ss} - IP_{sr}}{t_{zn} - t_{sr}}, t \in t_{ss} < t < t_{ss}, \tag{23}$$

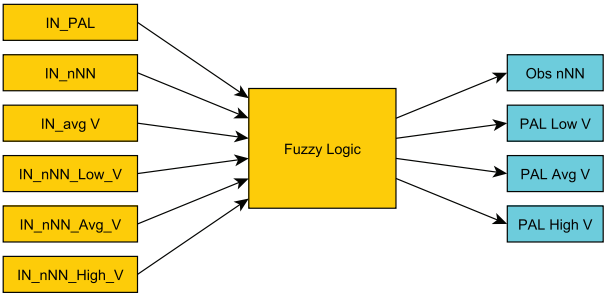
$$\delta IP_t = \frac{IP_{ss} - IP_{sr}}{t_{ss} - t_{zn}}, t \in t_{ss} < t < t_{ss}, \tag{24}$$

$$\delta IP_t = 0, t \in t > t_{ss}. \tag{25}$$

čia IP_{sr} – galia, tiekiama saulėtekio metu, IP_{zn} – galia, tiekiama vidurdienį, IP_{ss} – galia, tiekiama saulėlydžio metu, t_{sr} – saulėtekio laikas, t_{zn} – vidurdienio laikas, t_{ss} – saulėlydžio laikas.

Neaiškios logikos valdiklis

Neaiški logika – tai logikos rūšis, ji gali veikti kintamaisiais, kurių reikšmės pateikiamos kaip realieji skaičiai. Ji leidžia taikyti sprendimų priėmimo mechanizmą, kuris veikia pagal neskaitmeniškai apibrėžtas taisykles. Šiam taikymui siūloma naudoti neraiškiosios logikos valdiklį, kontroliuojantį skaičių NN, kurie atlieka atstumo matavimo užduotį. Tikslas – paskirstyti turimą galią taip, kad a) būtų galima gauti patikimai tikslų vietos nustatymo sprendimą ir b) jei turima



4 pav. Neaiškios logikos įvestys ir išėjimai

perteklinė galia būtų panaudota su konkrečiu NN susijusių matavimų dispersijai sumažinti. 4 pav. pavaizduotas neraiškiosios logikos valdiklio IO.

Neraiškusis loginis valdiklis turi šešis įėjimus ir keturis išėjimus:

- IN_PAL – nurodo galingos lokalizacijos užduoties kiekį.
- IN_nNN – nurodo stebimų NN skaičių.
- IN_avgV – nurodo vidutinį visų matavimų nuokrypio lygį.
- $IN\_nNN\_Low\_V$ – pateikia skaičių NN, kurių matavimų nuokrypių lygis yra žemas.
- $IN\_nNN\_Avg\_V$ – pateikia skaičių NN, kurių matavimų nuokrypiai yra vidutinio lygio.
- $IN\_nNN\_High\_V$ – pateikia skaičių NN, kurių matavimų nuokrypis yra aukštas.

Siūlomas neraiškiosios logikos valdiklis turi keturis išėjimus:

- Obs_nNN – nustato NN, kurie bus naudojami lokalizavimo užduočiai, skaičių.
- PAL_Low_V – nustato energijos kiekį, kuris bus skiriamas mažo nuokrypio NN.
- PAL_Avg_V – nustato energijos kiekį, kuris bus skiriamas vidutinio nuokrypio NN.
- PAL_High_V – nustato energijos kiekį, kuris bus skiriamas aukšto nuokrypio NN.

Geolokalizacijos duomenų šifravimas su kintamu tikslumu

Šiame darbe laikysime, kad geolokacinė informacija išreiškiama geografine platuma ir ilguma kaip sveikieji skaičiai, kurių didžiausi bitai žymimi laipsniais, po to minutėmis, sekundėmis ir galiausiai lanko sekundėmis. Priklausomai nuo vietos nustatymo algoritmo ir metodo tikslumo, tikimės, kad tikslumo skiriamoji geba bus 1 m. Tegul ilguma išreiškiama kaip $Y \in \mathbb{Z}$, o platuma – kaip $X \in \mathbb{Z}$. Tuomet plotą A sferoide galima apibrėžti naudojant X ir Y, taikant grindų funkciją. Grindų funkcija įgyvendinama pakeičiant nuliais iš anksto nustatytą mažiausiai reikšmingų kiekvienos koordinatės bitų skaičių. Kuo daugiau bitų pašalinama iš koordinačių, tuo didesnį plotą apibrėžia X ir Y. Sistema sutrinka, jei koordinatės atitinka ekvatorių arba pagrindinį dienovidinį. Šiam apribojimui įveikti įvedami papildomi bitai, nurodantys tikslų pašalintų bitų skaičių. Šie bitai pridedami X ir Y koordinačių pabaigoje. Iš X ir Y pašalintų bitų skaičius gali skirtis, todėl galima apibrėžti stačiakampį plotą sferoido paviršiuje. X ir Y su pašalintais bitais bus laikomi geometriniu srities A centru, o srities plotis ir aukštis bus lygus didžiausiam ilgiui tam tikroje geografinėje ilgumoje ir platumoje, išreikštam pašalinta koordinačių informacija.

Norint patikrinti, ar naujas koordinačių rinkinys X_t, Y_t patenka į sritį A, jos lyginamos su pradinėmis koordinatėmis X ir Y, pradedant nuo reikšmingiausio bito iki pirmojo pašalinto bito.

$$X_{chk} = X(MSB\ RMB) \oplus X_t(MSB : RMB), \tag{26}$$

$$Y_{chk} = Y(MSB\ RMB) \oplus Y_t(MSB : RMB). \tag{27}$$

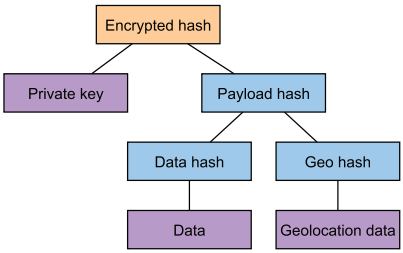
Jei ir X_{chk}, ir X_{chk} yra lygūs nuliui, tai koordinatės X_t, Y_t yra srityje A.

Norint tokią funkciją naudoti praktiškai, grindų funkcija turi būti įgyvendinta prieš taikant geolokacijos duomenims maišos funkciją. Kad ši operacija būtų sėkminga, rūko mazgas turi pranešti bitų, kurie turi būti ištrinti iš jo gauto geolokacinio sprendimo koordinačių, skaičių. Nustačius šį parametrą, jį galima naudoti tol, kol "Fog-Node" nepateikia naujo parametro. Jei grindų funkcija taikoma prieš maišos funkciją, rūko mazgas gali patikrinti MAST medį, negaudamas tikslaus geolokacinio sprendimo iš kraštinio mazgo.

4. Siūlomas daugiamodalus MAST medis

Siūlome dvi daugiamodalaus identifikavimo ir autentiškumo nustatymo metodo (MMIA) versijas. Pirmoji versija MMIA1 pavaizduota 5 pav., kurio šaknyje yra *Šifruotas maišos funkcijos rezultatas*.

MAST medis, pavaizduotas 5 pav., rodo MMIA1 struktūrą. Jis sukuriamas naudojant šifravimo, maišos ir asimetrinio šifravimo (angl. *asymmetric encryption*) funkcijas. *Turinio maiša* generuojamas sudedant *Perduodamų duomenų* ir



5 pav. MMIA1: Merkelio medžio struktūra

Geolokacijos duomenų maišas. Prieš pritaikant maišos algoritmą *Fog-End* mazgas nustatytu tikslumu užkoduoja *Geolocation data*. Grindų funkcija suteikia išlyginimo efektą, kuris naudojamas siekiant supaprastinti patikrinimą, ar perduodantis kraštinis mazgas yra gepgrafinėse ribose, apibrėžtose nustatyto tikslumo mažiausia skiriamąja geba. Sugeneravus *Turinio maišą*, jis užšifruojamas naudojant kraštinio mazgo *Privataus rakto*. Šifravimas suteikia sistemai autentiškumo nustatymo funkciją.

Algoritmas MMIA1 generuoti

Toliau išsamiai aprašomas saugaus kraštinio mazgo įrenginio *šifruotos maišos* generavimo algoritmas naudojant MMIA1 algoritmą:

1. Duomenims D turi būti pritaikyta iš anksto nustatyta maišos funkcija, kad būtų gautas maišos kodas D_h, kuris turi būti perduotas "Fog" mazgui. Maišos funkcijos išvestis bus žymima kaip dh_i:

$$D_h = h(D). \tag{28}$$

2. Vietos nustatymo sprendimo koordinatės C_X ir C_Y užmaskuojamos išvalant iš anksto nustatytą bitų skaičių, pradedant nuo mažiausio reikšmės bito. Tai pasiekiama sudarant kaukės skaičių m, kurio atskirų bitų vertė nustatoma pagal lygtį 29. Bitų, kuriuos reikia išvalyti, skaičių nustato "Fog" mazgas, ir jis žymimas kaip N_m. Kiekviena koordinatė bitais dauginama iš kaukės skaičiaus m:

$$m_i = 1 : i \geq N_m; 0 : i < N_m, \tag{29}$$

$$C_X^M(i) = C_x * m_i, i = n \ldots 0, \tag{30}$$

$$C_Y^M(i) = C_y * m_i, i = n \ldots 0. \tag{31}$$

3. Gautos užmaskuotos koordinatės C_X^M ir C_Y^M sujungiamos, kad būtų suformuoti užmaskuoti geolokacinės informacijos duomenys G^M:

$$G^M = C_X^M || C_Y^M. \tag{32}$$

4. Geolokacinės informacijos duomenims G^M turi būti taikoma iš anksto nustatyta maišos funkcija. Maišos funkcijos išvestis bus žymima G_h^M:

$$G_h^M = h(G^M) \tag{33}$$

5. Turinio maišos P_h generuojamas taikant maišos funkciją sujungtiems duomenų ir geografinės vietos maišos rezultatams:

$$P_h = D_h || G_h^M. \tag{34}$$

6. Turinio maišos kodas P_h užšifruojamas naudojant privatų kraštinio mazgo raktą K_{PR}, kad būtų gauta užšifruota maišos reikšmė EH_h, kuri gali būti perduota į rūko mazgą:

$$EH_h = E(P_h, K_{PR}). \tag{35}$$

MMIA1 algoritmas

Be to, išsamiai aprašomas miglos mazgo įrenginio gauto užšifruoto maišos identifikavimo ir autentiškumo patvirtinimo algoritmas:

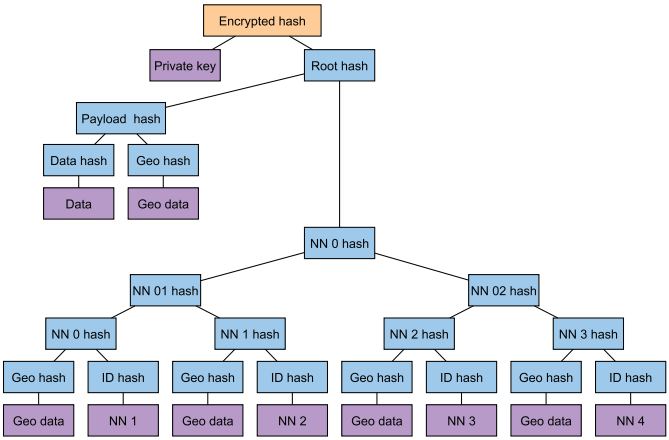
1. Gauta EH_h iššifruojamas su kraštinių mazgų viešuoju hejumi K_{PB}, kad būtų gautas naudingosios apkrovos maiša, kaip parodyta lygtyje 34:

$$P_h = D(EH_h, K_{PB}). \tag{36}$$

2. Geolokacinė maišos reikšmė sukuriama tais pačiais etapais, kaip apibrėžta lygtyse 29, 30, 31, 32 ir 33.

3. Duomenų maiša D_h apskaičiuojamas taikant maišos funkciją pranešimo duomenų segmentui, kaip apibrėžta lygtyje 28, o atkurtas naudingosios apkrovos maiša atkuriamas, kaip apibrėžta lygtyje 34.

4. Iššifruotas naudingosios apkrovos maišos turinys tada lyginamas *XOR* funkcija su atkurtu turinio maišos rezultatu. Jei rezultatas lygus nuliui, galima teigti, kad gauti duomenys yra vientisi ir jų geografinė kilmė yra teisinga.



6 pav. MMIA2: Merkelio medžio struktūra

MAST medis, pavaizduotas 6 pav., rodo MMIA2 struktūrą. Jis sudaromas naudojant šifravimo ir asimetrinio šifravimo funkcijas. *Root hash* sukuriamas susiejant *Payload hash* ir *NN 0 hash*. *Payload hash* turi identišką struktūrą, kaip ir *Root hash*, apibrėžtas MMIA1 algoritmo versijoje. Jis atlieka perduodamų duomenų vientisumo ir miglos mazgo nustatytos geografinės sąlygos tikrinimo funkciją. *Root hash* gaunamas minėtą *Payload hash* sugretinant su *NN 0 hash*. *NN 0 hash* yra šakninis maišos medžio, kuriame yra *ID hash*, ir kiekvieno *NN*, kuris buvo naudojamas kraštinio mazgo geolokacijos sprendiniui generuoti, perduotų geolokacijos *Data hash*, *Root hash*. Kad būtų galima patikrinti *NN 0 hash*, *Fog-Node* mazgas turi turėti kiekvieno *NN*, naudoto generuojant *Edge-Node* geolokacinį sprendimą, geolokacinę informaciją, o *Edge-Node* mazgas prie perduodamų duomenų turi pridėti *NN*, kurie buvo naudojami generuojant jo geolokacinę informaciją, ID. *NN's* ID turi būti pridedami tiksliai tokia pačia seka, kokia jie buvo naudojami *NN 0 hash* medžiui generuoti. Gautas *Root hash* vėliau užšifruojamas naudojant kraštinio mazgo *Privatųjį raktą*.

Pagal abu siūlomus algoritmus, kai rūko mazgas iššifruoja *Payload hash MMIA1* arba *Root hash MMIA2*, jis gali atkurti ir patikrinti MAST medžius, naudodamas tik geolokacijos sąlygą, kurią jis perdavė kraštiniam mazgui, ir gautus *NN's* ID. Patikrinimo etapą galima pagreitinti saugant iš anksto apskaičiuotas kiekvieno *NN # hash* vertes "Fog-Node" mazge.

MMIA2 algoritmas

Toliau išsamiai aprašomas saugaus kraštinio mazgo įrenginio *Encrypted hash* generavimo algoritmas naudojant MMIA2 algoritmą:

1. Duomenims D turi būti pritaikyta iš anksto nustatyta maišos funkcija, kad būtų gautas kodas D_h, kuris turi būti perduotas "Fog" mazgui. Maišos funkcijos išvestis bus žymima kaip dh_i:

$$D_h = h(D). \tag{37}$$

2. Vietos nustatymo sprendimo koordinatės C_X ir C_Y užmaskuojamos išvalant iš anksto nustatytą bitų skaičių, pradedant nuo mažiausios reikšmės bito. Tai pasiekiama sudarant kaukės skaičių m, kurio atskirų bitų vertė nustatoma pagal lygtį 38. Bitų, kuriuos reikia išvalyti, skaičių nustato "Fog" mazgas, ir jis žymimas kaip N_m. Kiekviena koordinatė bitais dauginama iš kaukės skaičiaus m:

$$m_i = 1 : i \geq N_m; 0 : i < N_m, \tag{38}$$

$$C_X^M(i) = C_x * m_i, i = n \ldots 0, \tag{39}$$

$$C_Y^M(i) = C_y * m_i, i = n \ldots 0. \tag{40}$$

3. Gautos užmaskuotos koordinatės C_X^M ir C_Y^M sujungiamos, kad būtų suformuoti užmaskuoti geolokacinės informacijos duomenys G^M:

$$G^M = C_X^M || C_Y^M. \tag{41}$$

4. Geolokacinės informacijos duomenims G^M turi būti taikoma iš anksto nustatyta maišos funkcija. Maišos funkcijos išvestis bus žymima G_h^M:

$$G_h^M = h(G^M). \tag{42}$$

5. *Payload hash* P_h generuojamas taikant maišos funkciją sujungtiems duomenims ir geografinės vietos koordinatėms:

$$P_h = D_h || G_h^M. \tag{43}$$

6. Kiekvienas NN, kuris dalyvavo generuojant lokalizacijos sprendimą kraštinio mazgo pusėje, teikia savo vietos nustatymo duomenis kraštiniam mazgui. Vėliau šie duomenys pakartotinai panaudojami generuojant kiekvieno atskiro NN *Geo hash* G_h^i, kaip parodyta lygtyje 44, kur N yra NN, kurie buvo naudojami generuojant lokalizacijos sprendimą, skaičius:

$$G_h^i = h(G^i), i = 1, 2 \ldots N. \tag{44}$$

7. Kiekvieno atskiro NN, dalyvavusio generuojant lokalizacijos sprendimą "Edge-Node" pusėje, ID; jo ID įtraukiamas į pranešimo duomenų segmentą, kuris perduodamas "Fog-Node" ir yra pranešimo dalis. Vėliau kiekvienas ID naudojamas generuojant *ID hash*:

$$ID_h^i = h(ID^i), i = 1, 2 \dots N. \tag{45}$$

8. Kiekvieno atskiro NN, kuris dalyvavo generuojant lokalizacijos sprendimą kraštinio mazgo pusėje, *Geo hash* G_h^i ir *ID hash* ID_h^i yra sujungiami ir sugretinami į vieną $NN_h^i h$ reikšmę:

$$NN_h^i hh = h(G_h^i|||ID^i), i = 1, 2 \dots N. \tag{46}$$

9. Sukurtos $NN_h^i hh$ reikšmės sujungiamos poromis ir taip suformuojamas kitas dvejetainio maišos medžio sluoksnis. Jei NN skaičius yra nelyginis, paskutinė likusi maišos vertė NN_h^N suglaudinama atskirai:

$$NN_h^{ij} = h(NN_h^i h|||NN_h^j hh), i = 1, 3 \dots N, j = i + 1. \tag{47}$$

10. Žingsnis 9 kartojamas tol, kol maišos reikšmių skaičius tampa lygus 1. Tada ši vertė žymima kaip NN_h^0.

11. *Payload hash* reikšmė P_h yra sujungiama su NN_h^0 reikšme ir sugretinama, kad būtų suformuota MAST medžio šakninė vertė:

$$Root_h = h(P_h||NN_h^0). \tag{48}$$

12. Gauta MAST medžio šaknis vėliau užšifruojamas naudojant kraštinio mazgo privatų raktą:

$$EH_h = E(P_h, Root_h). \tag{49}$$

MMIA2 autentifikavimas

Rūko mazgo įrenginio gauto užšifruoto identifikavimo ir autentiškumo patvirtinimo algoritmas:

1. Gautas EH_h iššifruojamas naudojant kraštinių mazgų viešąjį hešą K_{PB}, kad būtų gauta originali šakninė maiša. D lygtyje 50 reiškia dešifravimo funkciją:

$$Root_h = D(EH_h, K_{PB}). \tag{50}$$

2. *Geo hash* reikšmė sukuriama tais pačiais etapais, kaip apibrėžta lygtyse 38, 39, 40, 41 ir 42.

3. Duomenų maiša D_h apskaičiuojama taikant maišos funkciją pranešimo duomenų segmentui, kaip apibrėžta lygtyje 37, o atkurtos naudingosios apkrovos maiša P_h atkuriama, kaip apibrėžta lygtyje 43.

4. Kiekvienam ID, kuris buvo pridėtas prie pradinio pranešimo, *Geo hash* G_h^i, *ID hash* ID_h^i ir $NN_h^i h$ maišos vertės apskaičiuojamos pagal lygtis 44, 45 ir 46. Geolokaciniai duomenys gaunami iš vidinės rūko mazgo duomenų bazės arba gali būti prašomi perduoti iš NN^i kraštinio mazgo.

5. Atliekant rekursinį procesą, sugeneruoti NN^i maišos medžiai yra sujungiami poromis ir sugrupuojami į aukštesnius dvejetainio maišos medžio kamienus. Kai lieka tik viena maišos reikšmė, ji žymima kaip NN_h^0.

6. Apskaičiuotos naudingosios *Payload hash* vertės P_h ir NN_h^0 yra sujungiamos ir apskaičiuojama MAST medžio šaknies maišos vertė.

7. Apskaičiuotai *Root hash* yra taikoma išskirtinė arba funkcija su $Root_h$ verte, gauta naudojant lygtį 50. Jei šios operacijos rezultatas lygus 0, patikrinama kraštinio mazgo identifikacija ir autentiškumo patvirtinimas, kraštinio mazgo geolokacija ir duomenų vientisumas. Be to, patikrinamas NN, kuriuos kraštinis mazgas naudojo geolokacijos sprendiniui gauti, patikimumas.

5. Lengvas saugus srautinio perdavimo protokolas

Siekiant užtikrinti duomenų srautinį perdavimą scenarijuose, kuriuose toleruojamas nedidelis paketų praradimas, pavyzdžiui, duomenų srautinis perdavimas mažo pralaidumo tinkluose ir ribotų išteklių įrenginiuose, siūlome lengvą saugų srautinio perdavimo protokolą (LSSP), kuris būtų naudojamas duomenų srautinio perdavimo programose tarp galutinio įrenginio ir miglos galinio sluoksnio įrenginių. Protokolas, kuris pristatomas šiame skyriuje, anksčiau buvo paskelbtas žurnalo publikacijoje (Venčkauskas ir kt., 2018).

Protokolas – aprašytas šiomis pavadinimo savybėmis:

- Autentifikuotas duomenų srautas be ryšio užmezgimo.

- Atskiri veikimo režimai, skirti protokolui, kad būtų galima užtikrinti skirtingus saugumo lygius: duomenų siuntėjų autentiškumo nustatymas, duomenų vientisumas ir autentiškumo nustatymas, perduodamų duomenų konfidencialumas ir atsparumas daliniam duomenų praradimui perdavimo metu.

- Galimybė atnaujinti saugumo savybes net ir po duomenų srauto sąveikos, kai tarp siuntėjo ir gavėjo neperduodami jokie papildomi duomenys ir neatliekami jokie papildomi veiksmai, kuriuos turi atlikti bet kuris iš jų.
- Duomenų pridėtinių sąnaudų panaikinimas dėl to, kad visi įvesti saugumo duomenys įterpiami į modifikuotų UDP paketų antraštes. Duomenys, esantys UDP paketų viduje, protokole lieka nemodifikuoti.
- Protokolas palaiko daugialypės transliacijos ir transliavimo funkciją duomenų sraute.
- Saugios ir paprastos maišos, simetrinio šifravimo ir HMAC funkcijos yra pagrindinės technologijos, leidžiančios naudoti siūlomus protokolus. Šios funkcijos gali būti lengvai realizuojamos miglos galiniuose įrenginiuose.

Siūlomam protokolui įgyvendinti naudojami šie metodai:

1. Srautiniams duomenims perduoti iš galutinio įrenginio į rūko mazgą naudojami modifikuoti paketai iš UDP transporto antraštės (Zander ir kt., 2007). Duomenų paketų saugumo patikros ir eiliškumo keitimas atliekamas naudojant UDP paketo antraštės laukuose įterptą informaciją.
2. Nuolatinis autentiškumo nustatymas (Xie ir kt., 2015) tarp srautinio perdavimo įrenginio ir duomenų srautą priimančio įrenginio grindžiamas galutinio įrenginio autentifikatoriumi (Venčkauskas ir kt., 2012). Šiai funkcijai pasiekti naudojami slaptažodžiais pagrįsti pranešimų autentiškumo patvirtinimo kodai (HMAC), laiko žymos ir saugios slaptažodžių funkcijos. Duomenų konfidencialumui užtikrinti naudojami kiti saugūs algoritmai.
3. Laiko žymėjimas ir slaptųjų raktų naudojimas.
4. Kontrolinės sumos (**?**Performance of checksums and CRCs over real data**?**, 1998) ir perteklinių duomenų įtraukimas, pavyzdžiui, klaidų taisymo kodai (Ishengoma, 2014), (Reed ir kt., 1960), (Hamming, 1950) gali būti naudojami siekiant padidinti duomenų srauto patikimumą neidealioje tinklo aplinkoje, kad būtų sumažintas paketų praradimas.

Kad pasiūlytame protokole būtų užtikrintas veiksmingas ir lankstus saugus bendravimas, apibrėžiame tris saugumo režimus:

- 1 režimas – duomenų šaltinio autentiškumo nustatymas.
- 2 režimas – duomenų šaltinio autentiškumo patvirtinimas ir turinio vientisumas.
- 3 režimas – duomenų šaltinio autentiškumo patvirtinimas, turinio vientisumas ir konfidencialumo funkcijos.

Pakeistas UDP saugiam duomenų srautui

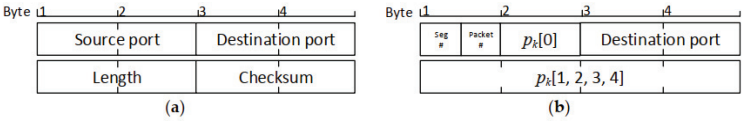
Kad būtų pasiektos protokole numatytos funkcijos, siūlome lengvą saugų srautinio perdavimo protokolą (LSSP), sukuriamą modifikuojant standartinius UDP paketus papildoma autentiškumo patvirtinimo informacija, kuri įtraukiama į UDP paketų antraštes. Originalaus UDP paketo struktūra nepažeidžiama, tačiau kai kurių laukų funkcionalumas pakeičiamas. Reikiama šaltinio autentiškumo patvirtinimo informacija (duomenų srauto autentifikatoriai ir dinaminiai įrenginiai) sukuriama iš įrenginių saugaus identifikatoriaus ir laiko žymos maišos rezultato. Tada sugeneruota autentifikavimo informacija atskiriama ir paskirstoma tarp UDP paketų antraščių. Dėl UDP protokolo konstrukcijos nėra paketų pristatymo garantijos, informacijos apie eiliškumą , neišvengiama dubliuojančių paketų perdavimo, todėl į juos įtraukėme LSSP duomenų segmentų ir paketų numerius. Papildoma kokia autentiškumo patvirtinimo informacija gali būti atkurta naudojant klaidų taisymo kodus. Šie klaidų taisymo kodai skirti tik autentifikavimo informacijai ir neatkuria perduotų duomenų iš srauto.

Pridedant pranešimo autentiškumo patvirtinimą, kurį jie ką tik švytėjo į perduodamą duomenų centrą, pasiekiamas autentiškumas ir duomenų vientisumas. Ši viršenybė apskaičiuojama perduodančiame įrenginyje naudojant saugų įrenginį identifikatorių (sid) ir saugias maišos funkcijas (h). Kadangi turima UDP antraštės erdvė yra ribota, duomenų srauto paketai paskirstomi įvesti duomenų segmentus s_i. Kiekvienam atskiram segmentui priskiriame eilės numerį: $i = 0, 1, ...n$.

Pranešimo autentiškumo patvirtinimo kodui generuoti naudojamo kodo algoritmo tipas daro įtaką segmentų ilgiui ir yra paskirstomas į n paketų. Papildoma perteklinė informacija pridedama prie duomenų srauto, kad būtų galima ištaisyti klaidas. Panašiai kiekvienas duomenų paketas $p_{i,j}$, $j = 0, 1, ...n - 1$, priklausantis tam pačiam duomenų segmentui si, surūšiuojamas ir jam suteikiamas eilės numeris i.

Kiekviename perduodamame pakete yra segmento numeris i ir atgalinis numeris j, kuris apibrėžia jo vietą segmente. Šie numeriai įtraukiami į pakeistą UDP antraštę. Priėmimo įrenginyje paketų numeriai naudojami gautiems paketams segmente išdėstyti, o segmentų numeriai naudojami skirtingiems duomenų srauto segmentams identifikuoti.

Paveikslėlyje 7 pavaizduota modifikuota UDP paketo struktūra. Šis metodas pagrįstas slaptųjų kanalų mikroprotokolo koncepcija. Modifikuoto UDP paketo antraštėje išlieka nepakitęs tik paskirties prievado laukas, palyginti su pradine UDP paketo antrašte (Zander ir kt., 2007). Šaltinio prievado pirmasis baitas suskaidomas į du 4 bitų mažylius, iš kurių pirmasis naudojamas segmento numeriui i saugoti, o paketo segmento sekos numeris j saugomas antrajame mažylyje. Antrajame baite šaltinio prievado, kontrolinės sumos laukai ir paketo ilgis naudojami



7 pav. Standartinio vartotojo datagramų protokolo (UDP) paketo (**a**) ir modifikuoto UDP paketo (**b**), kuriame yra autentifikavimo duomenys ir segmentų bei paketų numeriai, palyginimas

penkiems autentifikavimo duomenų baitams saugoti.

Jei autentiškumas apskaičiuojamas naudojant HMAC-SHA1 algoritmą, duomenų segmento dydį lemia $n = 4 + 1$ duomenų paketai. SHA1 algoritmu pagrįstos išklotinės dydis yra tik 160 bitų, kuriuos galima sutalpinti į keturias UDP paketų antraštes, dar vieno paketo reikia pridedant kontrolinę sumą, generuojamą XOR klaidų taisymo kodu. UDP antraštės laukai buvo pasirinkti remiantis šiomis prielaidomis:

- Šaltinio prievadas nėra svarbus laukas, nes siunčiantis įrenginys identifikuojamas pagal paskirties prievadą ir autentifikuojamas naudojant kitą metodą;
- UDP duomenų ilgis iš esmės yra nereikalingas laukas, nes duomenų ilgis gali būti apskaičiuotas pagal IP antraštės informaciją;
- Kontrolinės sumos laukas UDP antraštėje nėra privalomas duomenų vientisumas tikrinamas duomenų perdavimo sluoksnyje. Be to, duomenų vientisumas gali būti tikrinamas LSSP protokole.

Laisvas kai kurių UDP antraštės laukų modifikavimas gali sukelti tam tikrų problemų sudėtinguose tinkluose. tačiau pagrindinis LSSP protokolo tikslas yra ryšys su kitais tinklais, kuriuose naudojami maršrutizatoriai, ugniasienės ir t. t. tarp rūko mazgų ir galutinių įrenginių, kai naudojami tik OSI 2 lygio tinklo infrastruktūros įrenginiai. Mūsų stebėjimai rodo, kad modifikuoti UDP antraštės laukai nesukelia jokių papildomų problemų operacinių sistemų (Windows ir Linux) tinklo kaupyklėje, jei naudojamos žemo lygio tinklo bibliotekos (pvz., libpcap, winpcap (team, [s. a.]) ir t. t.).

Saugių prietaisų identifikatorių generavimas ir galutinių prietaisų registravimas

Pirmasis protokolo etapas – naujo rūko galinio įrenginio registracija rūko mazge ir saugaus įrenginio identifikatoriaus (sid), kurį žino tik duomenis siunčiantis galutinis įrenginys ir vienas (ar daugiau) iš rūko mazgų, priimančių duomenis

ir tikrinančių jų saugumo savybes, sukūrimas. Saugus įrenginio identifikatorius perduodamas rūko mazgams saugiu kanalu ir saugomas rūko mazge. Norint užregistruoti naują rūko galinį įrenginį rūko mazge, pirmiausia turi būti sukurtas pradinis saugus kanalas. Kadangi pradinės belaidžio ryšio sąsajos gali būti nesaugios, reikia sukurti alternatyvų saugų ryšio kanalą. Tiesioginis laidinis ryšys tarp dviejų komponentų, pavyzdžiui, naudojant USB arba eternetą, galėtų užtikrinti pakankamą apsaugą. Siūlome naudoti laidinį ryšį galutiniam įrenginiui registruoti, o tolesniam ryšiui – belaidį ryšį.

Autentifikavimo informacija (LSSP šifravimo raktai) generuojama iš saugaus įrenginio identifikatoriaus (sid). Todėl šis identifikatorius turi būti neloguojamas, geros kokybės, generuojamas tikrai atsitiktine tvarka, turėti pakankamą entropiją, būti pakankamo ilgio ir nesaugomas galutiniame įrenginyje. Šiam tikslui naudojamos fizinės neklonuojamos funkcijos (angl. *Physical Unclonable Functions*, PUF) (Hamming, 1950), tačiau PUF paprastai realizuojamos specialia aparatine įranga. Mes sukūrėme slapto šifravimo rakto generavimo algoritmą naudodami įterptosios sistemos parašą (Venčkauskas ir kt., 2012). Siūlomas metodas veiksmingai generuoja aukštos kokybės raktus be jokių papildomų techninės įrangos ir infrastruktūros sąnaudų, o tai labai svarbu ribotus išteklius turintiems įrenginiams. Siūlome šį algoritmą naudoti saugiems įrenginių identifikatoriams generuoti.

Be to, išsamiai aprašytas saugaus įrenginio identifikatoriaus (sid) generavimo algoritmas naudojant galutinių įrenginių parašą:

1. Sukurkite galutinio įrenginio komponentų parašų rinkinį $ES = es_i, i = 1, ...n$. Parašas sukuriamas taikant pardavėjo ID (cv_i), tipo ID (ct_i), modelio ID (cm_i) ir serijos numerio (csn_i) eilutę:

$$es_i = cv_i|||ct_i|||cm_i|||csn_i \tag{51}$$

   Atliekant 2 – 6 veiksmus, sukuriamas komponentų parašų poaibis. Šie parašai bus naudojami galutinio įrenginio parašui apskaičiuoti.

2. Apskaičiuokite įrenginio įterptosios programos antraštės maišos rezultatą $ph = h(pk|||psn)$.

3. Sukurkite $n\ kartų m$ matricą $MH = mh_{ij}$ iš prietaiso įterptosios programos antraštės maišos baitų $mh_{ij} = eb(ph, (i-1) \times j+i)$, kur n yra galutinio prietaiso parašų skaičius, o $m = eb(ph, n) mod n$.

4. Apskaičiuokite s_j matricos MH stulpelių elementų sumą, $s_j = \Sigma_{i=1}^{n} mh_{ij}, j = 1, ..., m$.

5. Sukurkite komponentų paraščių indeksų masyvas $IND = ind_j$, kur $ind_j = s_j mod n$, ir ištrinkite pasikartojančius indeksus, $ind_j \neg ind_i, \forall i \in 1...j-1$.

6. Sukurkite komponentų parašų poaibį $\widetilde{ES} \subseteq ES, \widetilde{es_i} = es_j$, kur $j = ind_k, \forall ind_k \in IND, k = 1, ..., m$, iš kurio bus sukurtas galutinio įrenginio parašas.

7. Sukurkite galutinio įrenginio parašą $ss_i = sign(\widetilde{ES})$.

8. Sukurkite slaptąjį įrenginio identifikatorių $sid = fsid(ss, salt, iteration\_count, key\_length)$, kur $salt = eb(ph, n)modn, iteration\_count = count(\widetilde{ES})$.

LSSP 1 režimas: šaltinio autentifikavimas

Siekiant palengvinti duomenų srauto duomenų šaltinio autentiškumo patvirtinimą, visų duomenų paketų antraštėse yra segmento pranešimas apie autentiškumo patvirtinimo kodo suvestinę ir duomenų segmento klaidų taisymo kodo suvestinę.

Perduodamose UDP antraštėse yra segmento ir paketo ID numeriai. Vėliau šie numeriai naudojami gautiems paketams sutvarkyti pagal viršelio fragmentų seką. Šie viršelio fragmentai paskirstomi tarp skirtingų to paties segmento paketų. Iš perduodančiojo įrenginio nereikalaujama atlikti jokių perduodamų duomenų pakeitimų ar skaičiavimų.

Duomenų viršenybės vertė nepriklauso nuo duomenų turinio; duomenų šaltinio autentiškumo patvirtinimas atnaujinamas atliekant šiuos skaičiavimus:

1. $mac1_i = HMAC(sid, ts|||i)$, kur sid yra saugus šaltinio identifikatorius, ts yra dabartinė laiko žyma, o i yra perduoto segmento numeris.

2. Skaitmeninis failas padalijamas į fragmentus $p_k = submac(mac1_i)$, kur $k = 1...m, m = lenght(mac1_i)/5$.

3. Apskaičiuojamas $mac1_i$ klaidų taisymo kodas: $ecc_i = fecc(p_1...p_k)$, kur $fecc$ yra pasirinkta klaidų taisymo funkcija.

4. p_k ir ecc_i įterpimas į UDP antraštes. Po įterpimo paketai yra paruošti perduoti gavėjui.

Kad duomenų srautų šaltinis būtų autentifikuotas, priimantis įrenginys turi surinkti visus pk fragmentus iš to paties segmento ir sudaryti bendrą $mac1_i$ sumą. Jei ne visi paketai buvo gauti, juos galima atkurti naudojant klaidų taisymo kodą. Priimantis įrenginys taip pat turi apskaičiuoti savo atitinkamos funkcijos versiją, kad gautų $mac1_r$. Duomenų autentiškumo patvirtinimas patikrinamas, jei abi vertės yra vienodos.

LSSP 2 režimas: šaltinio ir turinio autentiškumo nustatymas

Siuntimo mazge turi būti laikomasi šios šaltinio ir turinio autentiškumo patvirtinimo procedūros:

1. Kiekvienam naujam srauto segmentui s_i turi būti generuojamas naujas autentifikavimo raktas k_i pagal lygtį $k_i = H(sid|||ts|||i)$, kur i yra segmento numeris perdavimo metu, laiko žyma žymima ts, o sid yra saugus šaltinio identifikatorius.

2. Kai visi segmentų duomenų fragmentai yra sujungti, $HMAC$ *digest* galima apskaičiuoti naudojant visus į paketus įtrauktus duomenis, taikant k_i raktą, o duomenų sujungimas $mac2 = HMAC(ki, duomenys)$ atliekamas, kai į duomenų segmentą, įtraukti duomenų paketai.

3. Skaidymas atliekamas naudojant $p_k = submac(mac1_i)$, kur $k = 1...m, m = lenght(mac1_i)/5$.

4. Klaidų taisymo kodas $mac1_i$ apskaičiuojamas naudojant $ecci = fecc(p_1...p_k)$, kur $fecc$ yra norima taisymo funkcija.

5. Atnaujinus UDP antraštes su p_k ir ecc_i, duomenų segmentas gali būti perduotas į gavėjo duomenų srautą.

Priimantis įrenginys turi surinkti visus duomenų paketus, sudarančius duomenų segmentą. Kai visi jie bus surinkti, $mac2_s$ suvestinę vertę galima atkurti išgaunant duomenis iš atitinkamų paketų antraščių". Jei perduodant duomenis paketas buvo prarastas, tam tikriems trūkstamiems fragmentams atkurti galima naudoti klaidų taisymo kodą.

Gavėjo $mac2_r$ digest versija apskaičiuojama iš gauto rakto k_i reikšmių, naudojant duomenis, gautus iš srauto. Jei siuntėjo $mac2_s$ vertė atitinka gavėjo $mac2_r$ vertę, tuomet siuntėjo tapatybė patvirtinta ir patikrintas duomenų vientisumas. Tokiu atveju duomenų paketai paliekami nepakeisti visame duomenų segmente.

LSSP 3 režimas: šaltinio autentiškumo nustatymas, turinio autentiškumo nustatymas ir konfidencialumas

Siekiant toliau tobulinti šį pasiūlyto protokolo variantą, galima naudoti simetrinį šifravimą, kuris padidina saugumo savybes ir užtikrina duomenų konfidencialumą. Turinio vientisumas ir šaltinio autentiškumo patvirtinimas užtikrinamas taikant identišką procedūrą, kaip ir 2 režimo algoritmo versijoje. Padidinimas pasiekiamas naudojant simetrinį šifrą, kuriuo užšifruojami visi duomenys (pvz., AES) CBC režimu, po to, kai apskaičiuojama santrauka.

Duomenų paketas nepriklausomai užšifruojamas siunčiančiajame įrenginyje, naudojant iniciacijos vektorių iv_j, $j = 0, 1, ..., n-1$ ir slaptą šifravimo raktą ek_i. Visi paketai turi tą patį i-ojo segmento šifravimo raktą. Slaptasis šifravimo raktas generuojamas pagal šią lygtį: $ek_i = H(sid|||jjj|||ts|||jjj|||i)$, kur sid yra saugus šaltinio identifikatorius, ts – dabartinė laiko žyma, o H – ta pati saugi maišos funkcija, kuri naudojama HMAC skaičiavimams. Jei gauto slaptojo rakto ilgis yra per didelis pasirinktam šifravimo algoritmui, slaptasis raktas sutrumpinamas, kad atitiktų naudojamo šifravimo algoritmo reikalavimus. Turi būti pasirinkta pakankamai saugi glaudinimo funkcija, kad būtų įvestas pakankamai ilgas glaudinimo funkcijos rezultatas, siekiant sugeneruoti pakankamai saugų šifravimo raktą. Jei šifravimui pasirinktas AES256, tai, siekiant reikiamo saugumo lygio, HMAC skaičiavimams atlikti turėtų būti naudojamas bent jau SHA256 algoritmas.

CBC šifravimo modelyje naudojamas iniciacijos vektorius turi būti skirtingas kiekvienam atskiram duomenų paketui ir turi būti apskaičiuojamas pagal šią lygtį:

$$iv_j = H(sid||i|||j). \tag{52}$$

Toks saugumo parametrų išvedimo protokolas užtikrina, kad priimančioji šalis galės iššifruoti duomenis net ir tuo atveju, jei duomenys segmento viduje bus prarasti. Prarastus paketus galima atkurti tik tada, kai yra pakankamai klaidų taisymo duomenų. Viso segmento atkurti neįmanoma.

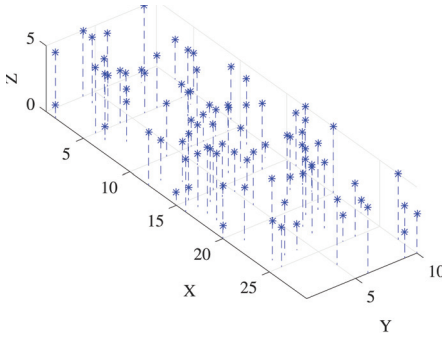
Jei reikia, atsparumą duomenų praradimui galima padidinti įvedant tropines modifikacijas, t. y. su papildomais duomenų paketais įvedant perteklinę klaidų taisymo informaciją. Jei tokia informacija įvedama, reikia apskaičiuoti tų duomenų paketų klaidų taisymo kodą.

III. METODAI

1. Simuliacijos sąranka paskirstymo vietos nustatymo algoritmui patikrinti

Modeliavimas atliktas MATLAB aplinkoje. 100 mazgų rinkinys buvo atsitiktinai paskirstytas 30m x 10m x 5m dydžio erdvėje, kaip parodyta 1 pav. Buvo įvestas maksimalus ryšio atstumas ir apskaičiuoti atstumai buvo veikiami 2% atsitiktinio triukšmo, kai atstumai vertinami naudojant RTT, kada ryšio atstumas yra 5m.

Inicijuojant modeliavimą, tinkle nėra padėties nustatymo informacijos. Pirmiausia kiekvienam atskiram NN sukuriami LRF. Gavus nuotolio nustatymo informaciją, kiekvienas mazgas sukuria individualią duomenų bazę. Modeliavimo tikslais duomenų bazė buvo sujungta į vieną struktūrą. Nors iš tikrųjų kiekvieno



1 pav. Simuliuojamo tinklo mazgai

atskiro LRF formavimo trukmė gali skirtis, modeliavimo tikslais visas LRF apskaičiuojame per t_0. Apskaičiuotos LRF nėra išlygintos pagal absoliučią padėties nustatymo sistemą ir yra atsitiktinai orientuotos.

Apskaičiavus LRF, į imituojamą aplinką išleidžiamos imituojamos UN. UN judėjimas imituojamas laike, kai mėginių ėmimo dažnis yra 1 s, kaip diskretiškas nuotolio matavimų rinkinys su vienodais apribojimais, kurie buvo taikomi NN. Kiekvienas UN juda tiesine trajektorija 1 m/s greičiu. VN sukuriami tik tada, kai su atskiru LRF atliekami 3 nuotolio matavimai. Jei UN yra dviejų ar daugiau LRF persidengimo zonoje ir su kiekvienu atskiru LRF atliekami bent 3 nuotolio matavimai, VN gauna padėties nustatymo sprendimus visuose LRF. Kai JN baigia savo maršrutus, tinklų būklė tiriama atsižvelgiant į VN padėties nustatymo tikslumą ir LRF konvergenciją su absoliutine padėties nustatymo sistema. Algoritmų veikimas buvo įvertintas atlikus 100 simuliacijų su atsitiktinėmis trajektorijomis.

2. Simuliavimo sąranka HCS patikrinti

Siekiant patvirtinti siūlomą hibridinę valdymo sistemą, buvo sukurtas MATLAB modeliavimas. Modeliuojant sukuriamas turimų NN skaičius ir kiekvienam atskiram NN priskiriamas pradinis dispersijos lygis. Viso modeliavimo metu hibridinei valdymo sistemai pateikiamas atsitiktinis šių NN rinkinys. Modeliuojant imituojama saulės energijos gamyba, priklausanti nuo geografinės ilgumos, metų dienos, paros laiko ir triukšmo lygio parametrų.

Siekiant patvirtinti siūlomą hibridinę valdymo sistemą, vertinamas jos gebėjimas išlaikyti funkcionalumą per 24 valandų ciklą, kartu sumažinant prastovos laiką, maksimaliai padidinant lokalizacijos tikslumą ir išvengiant akumuliatoriaus persotinimo energija.

24 valandos modeliavimo laikotarpis padalijamas į 864 imčių, kurių kiek-

viena sudaro 100 sekundžių. Kiekvieno mėginio ėmimo metu Kalmano filtras generuoja galingos lokalizacijos užduoties įvertį, o Neaiškios logikos valdiklis organizuoja ryšius su atskirais NN. Nustatomas toks pastovus energijos suvartojimas, kad, neskiriant galios lokalizavimui, daiktų interneto mazgas galėtų atlikti fonines užduotis dvi dienas, jei pradėtų veikti su pilnu akumuliatoriumi ir negautų energijos iš saulės energijos generatoriaus.

3. Daugiamodalaus identifikavimo ir autentiškumo nustatymo metodo tikrinimo modeliavimo sąranka

Siekiant patikrinti pasiūlytą daugiamodalaus identifikavimo ir autentiškumo nustatymo algoritmą, MATLAB programa buvo sukurtas modeliavimas. Modeliavimą sudaro MN su priskirtais privačiais raktais ir geolokacinėmis statinėmis koordinatėmis. Modeliuojama, kaip mobilusis MN keliauja per geolokacinę vietą, kurioje nustatyta konkreti dominanti sritis. Nurodyta zona MN nežinoma ir ją patikrina tik MN perduotų duomenų gavėjas. MN nepertraukiamai perduoda duomenis dabartiniu pastoviu mėginių ėmimo dažniu, kurie yra pasirašyti siūlomu daugiamodaliu identifikavimo metodu.

Maišos medį sudaro MN viešasis raktas, kuris naudojamas generuojant *Top hash* ir *Payload hash*. Šis raktas yra pastovus, ir jo nereikia iš naujo apskaičiuoti. *Payload hash* generuojamas sugretinant *ID hash* ir *Geolocation hash*. *ID hash* generuojamas sugretinant *Data hash* su siuntėjo viešuoju raktu. Jei MN yra duomenų šaltinis, šiame etape naudojamas jo paties viešasis raktas. *Data hash* generuojamas apdorojant perduodamus duomenis maišos funkcija.

Modeliuojant tiriami įvairūs scenarijai, pagal kuriuos generuojant *Top hash* įtraukiami neteisėti slaptažodžiai. Pirmiausia tiriama galimybė nurodyti geografinę vietą, tada tiriama, ar į geolokacijos duomenų generavimą neįtraukiami neteisėti NN.

4. Eksperimentinė lengvojo saugaus srautinio perdavimo protokolo patikrinimo sąranka

Kokybinė analizė

Siūlomas protokolas, palyginti su DTLS, turi papildomų privalumų, pavyzdžiui, supaprastinta naujų įrenginių registracija srautinio perdavimo sesijai. DTLS protokole taikoma rankų sukrėtimo procedūra, kurios metu nustatomas kliento ir serverio autentiškumo patvirtinimas. Rankų sukrėtimo procedūros metu naudojami x.509 sertifikatai. Norint naudoti DTLS protokolą, x.509 sertifikatai turi būti sukurti, pasirašyti ir perduoti visiems tinklo mazgams. Generuojant x.509 sertifikatus reikia griežtai valdyti, saugoti ir atšaukti visus tinkle išduotus sertifikatus.

Norint užmegzti naują ryšį naudojant LSSP protokolą tarp kliento ir ser-

verio, nereikia jokių specialių etapų. Jei naudojama protokolo versija yra M1 – paprasčiausias variantas, į tinklu perduodamus paketus neįvedama jokių duomenų pridėtinių sąnaudų, todėl duomenų autentiškumui nustatyti duomenų srautas yra nulinis.

Be to, DTLS naudojamai rankų sukrėtimo procedūrai užbaigti reikia daug laiko. Rankų sukrėtimo procedūros trukmė priklauso nuo tinklo kokybės. Jei prarandama daug duomenų, rankos paspaudimo procedūros trukmė gerokai pailgėja. Be to, apsidraudimo procedūra reikalauja dvikrypčio kliento ir serverio ryšio, todėl dar labiau padidėja tinklo duomenų srautas. Ši tinklo apkrova gali turėti neigiamos įtakos, jei rankinio apsikeitimo procedūrą tenka periodiškai kartoti. DTLS paketų duomenų laukai naudojami papildomai, į juos įterpiant protokolo ir saugumo duomenis. Šie papildomi duomenys dar labiau padidina bendrą duomenų srautą.

Kadangi DTLS protokole dirbantys darbuotojai gauna laiko informaciją, dėl ilgos duomenų srauto sesijos pertraukos tenka atlikti papildomas rankos suvedimo procedūras. LSSP protokolas tokio trūkumo neturi, nes duomenų srauto sesijos atnaujinimas nereikalauja jokių papildomų veiksmų. Gavėjas gali automatiškai atlikti šaltinio autentiškumo patvirtinimą iš karto po to, kai baigiamas priimti visas pirmasis segmentas.

Kadangi DTLS protokolas yra "taškas-taškas" architektūros, jis negali būti veiksmingai naudojamas daugiaadresiniam duomenų siuntimui daugeliui gavėjų vienu metu. Jei reikia tokios funkcijos, su kiekvienu imtuvu reikia užmegzti atskirus ryšius. Dėl to dėl galios ir pralaidumo apribojimų DTLS negali būti naudojamas daugiaadresinio duomenų siuntimo programose, skirtose kompiuteriams. Siūlomas LSSP tokių trūkumų neturi, todėl gali būti efektyviai naudojamas tais atvejais, kai reikalingas daugiaadresinis duomenų perdavimas. Kadangi LSSP nenaudoja jokių rankų sukrėtimo metodų, jis taip pat gali būti naudojamas transliavimo programoms, nes nereikia patvirtinimo iš gavėjų ar ryšio su jais.

Siūlomas LSSP protokolas taip pat pranašesnis už DTLS, kai duomenys prarandami triukšmingai ir tinkle, nes DTLS protokole nėra jokio mechanizmo, kuris tokiomis sąlygomis padėtų spręsti klaidų ar pristatymo sutrikimų problemas. DTLS užtikrina duomenų vientisumą tik rankų suvedimo etape, jei paketas prarandamas žemesniuose UDP/IP kamino lygiuose. Didesnis atsparumas duomenų praradimui dėl klaidų, atsirandančių žemesniame tinklo kaupyklės lygmenyje, gali būti tvarkomas LSSP protokolu naudojant ECC, kad būtų galima patvirtinti naudingojo krūvio duomenų autentiškumą.

LSSP protokolo trūkumai, palyginti su standartiniais UDP arba DTLS protokolais, yra didesni perdavimo įrenginio atminties reikalavimai, nes segmento duomenys turi būti visiškai agreguoti atmintyje, kad būtų galima atlikti reikiamas duomenų autentiškumo patvirtinimo ir ECC sąlygas. Priklausomai nuo naudojamų kodų santraukos ilgio, ši atminties apkrova gali turėti neigiamos įtakos. Jei naudojamas HMAC-SHA1, buferį reikia padidinti penkis kartus, palyginti su standarti-

niais reikalavimais UDP perdavimams.

Reikalavimas sujungti visų šių segmentų duomenis padidina duomenų srauto per tinklą vėlavimą, nes skaičiavimus galima atlikti tik tada, kai visi segmentų duomenys įkeliami į atmintį. Poveikis vėlavimui yra didesnis, jei naudojamas "lėtas" duomenų srautas. Didesnio vėlinimo sumažinimo galima pasiekti sumažinus paketų dydį.

Vykdymo analizė

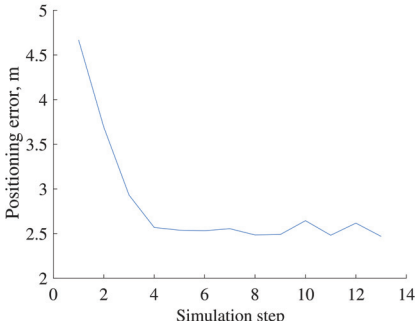
Siekiant įvertinti siūlomo metodo našumo charakteristikas, buvo sukurtas srautinio perdavimo klientas ir priėmimo klientas. "Galutinio įrenginio" prototipui įgyvendinti buvo naudojamas įterptinis "Raspberry Pi" kompiuteris (B modelis, 2 versija, BCM2835 procesorius, 512 MB RAM), kuriame įdiegta "Raspbian GNU/Linux 9 (stretch)". Rezultatai gauti atliekant matavimus siunčiančiajame įrenginyje. Kaip duomenų srautą priimanti šalis buvo naudojamas standartinis "Windows 10" kompiuteris. LSSP protokolui įgyvendinti naudota "Java" programavimo kalba. Atvirojo kodo saugumo bibliotekos iš Bouncy Castle (Bouncy Castle Inc., 2018) buvo integruotos į įgyvendinimą. jnetpcap java biblioteka (Technologies, 2018) buvo naudojama siekiant gauti prieigą prie UDP paketų antraščių. libpcap (team, [s. a.]) ir winpcap sisteminės bibliotekos buvo naudojamos siekiant gauti sąsają su žemuoju lygiu. Bandymai, susiję su DTLS, buvo atlikti naudojant "Java" gimtąjį DTLS protokolą, kurį pateikė "Bouncy Castle (*Legion of the Bouncy Castle Inc. Java (D)TLS API and JSSE Provider. User Guide.* 2018)".

IV. REZULTATAI

1. Paskirstytosios lokalizacijos algoritmo patikrinimo rezultatai

1 pav. pavaizduota vidutinė atsitiktinai sugeneruotų JN, judančių tinkle tiesine trajektorija, padėties nustatymo paklaida. JN tinklo aprėpties zoną įveikė mažiau nei per 14 žingsnių. Kiekvieno žingsnio metu VNN buvo kuriami, jei buvo LRF. Pradiniuose modeliavimo etapuose padėties nustatymo paklaida yra gerokai didesnė dėl nepalankios geometrijos, nes JN patenka į modeliuojamą tinklą, o VN yra tik priešais ją. Toliau UN judant modeliuojamoje erdvėje, ji patenka į tinklą ir tampa apsupta NN. Kai NN yra aplink JT, gaunamas tikslesnis padėties nustatymo sprendimas.

Atlikus modeliavimą, NN pasiekė konvergaciją. Nors visiška tinklo konvergacija nebuvo pasiekta, pažymėtina, kad tik su 3 UN, turinčiais APS funkciją ir einančiais atsitiktiniu tiesiaeigiu keliu, keturiose zonose įvyko vietinis tinklo konvergavimas. Papildomos iteracijos gerokai pagerina tinklų konvergavimo lygį ir padėties nustatymo tikslumą, kaip parodyta 0 lentelėje. Visiška tinklo konvergacija nepasiekta, nes tinklo pakraščiuose esantys NN nesugebėjo suformuoti LRF.



1 pav. Vidutinė visų imituojamų UN padėties nustatymo paklaida joms kertant imituojamą tinklą

0 lentelė. Paskirstytųjų lokalizacijos algoritmų modeliavimo rezultatų santrauka

Iteracijų skaičius	NC lygis, %	Vidutinė padėties nustatymo paklaida, m
1	26	$2,8$
2	63	$2,2$
4	70	$1,9$
8	74	$1,9$

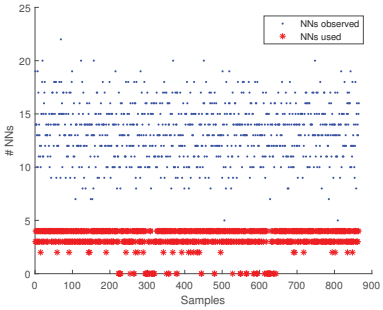
2. Galios valdymo HCS lokalizavimo funkcijoms patikrinti rezultatai

2 pav. mėlynais taškeliais pavaizduotas turimų NN skaičius kiekvienu imties metu, o raudonomis žvaigždutėmis pavaizduotas panaudotų NN kiekis, kurį nurodo neraiškiosios logikos valdiklis. Šis skaičius generuojamas atsitiktinai priskiriant iš turimų NN. Konkretus NN gali atsirasti, dingti ir vėl atsirasti per visą modeliavimą. Kiekvienas NN identifikuojamas pagal savo ID, todėl jį galima lengvai priskirti prie turimų NN. NN surūšiuojami pagal variantiškumo lygius, ir neaiškios logikos valdiklis iš kiekvienos grupės atsitiktinai pasirenka tam tikrą NN skaičių, priklausantį nuo turimos galios.

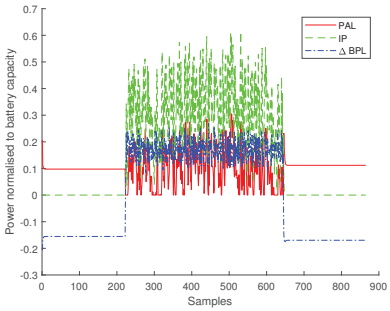
3 pav. pavaizduota saulės energijos generatoriaus tiekiama galia – žalia brūkšninė linija, lokalizacijai priskirta galia – raudona tiesi linija, o akumuliatoriaus energijos lygio pokytis – mėlyna brūkšninė linija.

4 pav. parodytas tiesinis akumuliatoriaus galios lygio BPL_{Proj} įverčio modelis per modeliavimo laikotarpį ir faktiškai gautas akumuliatoriaus galios lygis BPL, valdomas Kalmano filtru.

5 pav. parodyta, kaip laikui bėgant keičiasi atskiro NN dispersijos lygis,

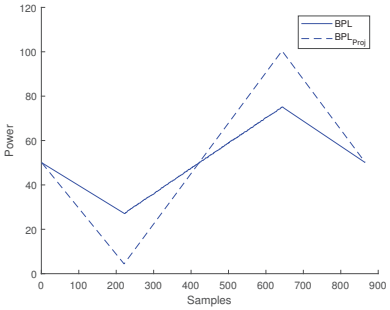


2 pav. Kraštinio mazgo stebimų NN skaičius ir neraiškiosios logikos valdiklio vietos nustatymo matavimams priskirtų NN skaičius

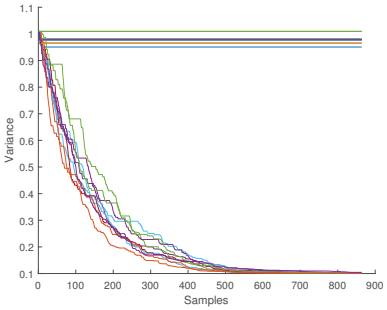


3 pav. Galios įvestis į Kalmano filtrą. PAL – lokalizacijai priskirta galia, IP – įėjimo galia, ΔBPL – akumuliatoriaus galios lygio pokytis

kai hibridinė valdymo sistema paskirsto galią, kad su ja atliktų nuotolio matavimus. Šiame modeliavime nuotolio matavimų dispersijos pokytis modeliuojamas kaip visos sukauptos galios logaritminė funkcija. Nors šis modelis neatspindi visų apibendrinto tikslumo niuansų, palyginti su vidutiniais rezultatais, jis naudojamas kaip iliustracija.



4 pav. Kalmano filtro valdomas akumuliatoriaus energijos lygis per imituojamą 24 valandų laikotarpį

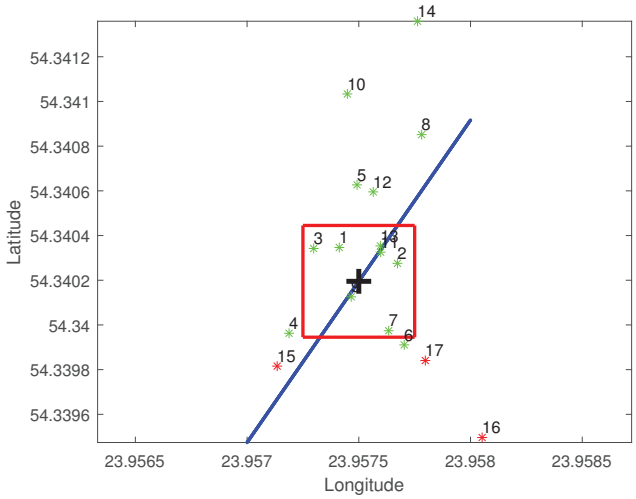


5 pav. Pasikeitusi atskirų NN RTT nuotolio sprendimo dispersija, susidariusi dėl matavimų vidurkinimo

3. Daugiamodalaus identifikavimo ir autentiškumo nustatymo metodo patikrinimo rezultatai

6 pav. pavaizduotas išjungtas modeliavimas. Mėlyna linija vaizduoja MN trajektoriją, žalios žvaigždutės – galiojančius NN, o raudonos žvaigždutės – negaliojančius NN, kurie bus naudojami lokalizacijai. Juodas kryžius žymi paskirtą geolokaciją, iš kurios laukiama duomenų. Raudonas kvadratas žymi tikslumo apribojimus. Duomenys, perduodami iš kvadrato ploto, laikomi galiojančiais, o duomenys, perduodami už nustatytos teritorijos ribų, atmetami.

7 pav. parodytas neteisėtų NN, aptiktų šaltinio perduotame paraše, skaičius. Todėl duomenys, kuriuose yra vietos nustatymo sprendimas, pagrįstas neteisėtų NN atstumo matavimais, atmetami kaip negaliojantys.



6 pav. Simuliacinis MN, važiuojančio per nustatytą zoną, scenarijus

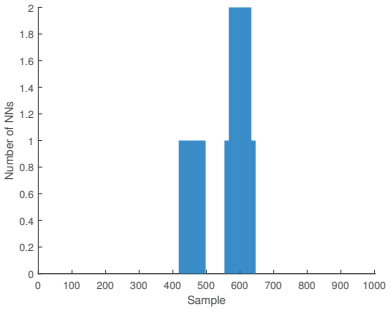
4. Lengvojo saugaus duomenų srautinio perdavimo protokolo patikrinimo rezultatai

Šešių analizuotų protokolų našumo palyginimas, kai nuosekliai perduodama 10 MB duomenų, atsižvelgiant į perdavimo trukmę, naudojant skirtingo ilgio paketus, pateikiamas 8 pav.

Nepriklausomai nuo perduodamo paketo ilgio, UDP protokolas užtikrina trumpiausią perdavimo laiką tarp analizuotų protokolų. Jų LSSP protokolo M1 versijos našumas yra labai panašus į UDP rezultatus. Pagrindinis skirtumas tas, kad M1 papildomai perduoda duomenų šaltinio autentiškumo patvirtinimo funkciją, kurios nėra UDP protokole. Nedidelį šių dviejų protokolų skirtumą galima paaiškinti didesniais skaičiavimo reikalavimais, kuriuos kelia M1 protokole numatyti skaičiavimai, juos reikia atlikti perduodančiame įrenginyje.

Tačiau bendras belaidžiais kanalais perduodamų pasiūlymų kiekis yra lygiavertis tiek UDp, tiek M1 atvejais, o vienintelis veiksmingas šių dviejų protokolų skirtumas yra ilgesnis siųstuvų apdorojimo laikas, reikalingas reikiamiems skaičiavimams atlikti prieš siunčiant paketus. Šį skirtumą galima pastebėti paveikslėlyje kaip $0,5$ s perdavimo laiko skirtumą, perduodant M1 protokolu 10 MB duomenų 512 B dydžio duomenų paketuose.

Siūlomo protokolo našumui įvertinti kaip etaloną galima naudoti DTLS2 ir M2 protokolų palyginimą, jie abu gali užtikrinti panašų našumo lygį ir pasižymi



7 pav. Neteisėti NN, aptikti lokalizacijos sprendime

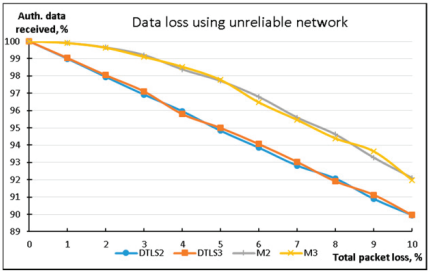
8 pav. Laikas, reikalingas 10 MB duomenų perdavimui kiekvienu iš analizės protokolų

panašiomis duomenų ir šaltinio autentiškumo nustatymo savybėmis.

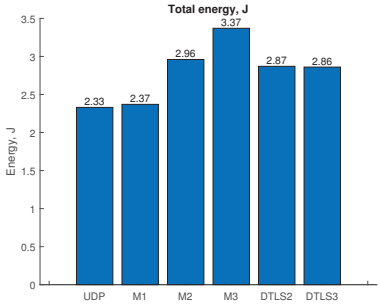
M3 yra lėtesnis nei DTLS3, nes papildomas duomenų paketas su ECC informacija duomenims siunčiamas kiekviename duomenų protokole.

Siekiant įvertinti, kaip metodai veikia realiuose tinkluose, buvo naudojamas NetEM (Network Emulation) įrankis. NetEM įrankis suteikia galimybę emuliuoti įvairias sąlygas ir tinklo funkcionalumą. Mes naudojome NetEM (Hemminger ir kt., 2005) perdavimo įrenginyje (Raspberry Pi), kad imituotume atsitiktinį paketų praradimą tinklo techninėje įrangoje. Priimdami ir siekėme surinkti visus turimus duomenų paketus. Prarastus duomenų paketus, jei įmanoma, atkurdavome. 9 pav. pavaizduoti gauti šio eksperimento rezultatai.

Siekiant įvertinti siūlomų metodų energijos vartojimo efektyvumo lygį, išmatuotas bendras to svajojančio įrenginio suvartojamos energijos kiekis, kai 10 MB duomenų buvo perduodami 256 B paketais. Mokymo įrenginį sudarė Raspberry Pi kompiuteris ir prie jo prijungtas USB WiFi adapteris. Gauti rezultatai pavaizduoti 10 pav.



9 pav. Eksperimentiniai praktinio našumo rezultatai neoptimalioje tinklo infrastruktūroje

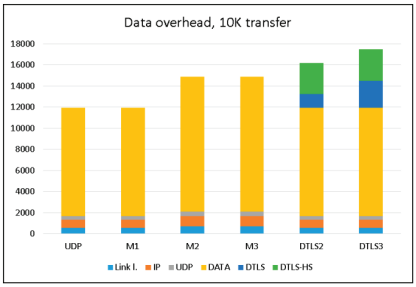


10 pav. Eksperimentiniai energijos suvartojimo bandymo rezultatai

Digitus Wireless 150N USB adapteris buvo naudojamas perdavimo įrenginio energijos suvartojimui matuoti naudojant srovės šuntą ir stalinį multimetrą. Duomenų perdavimo eksperimentų metu energijos suvartojimo matavimai buvo renkami naudojant standartinį "Windows 10" kompiuterį. Į skaičiavimus neįtraukta USB "Wi-Fi" adapterio suvartojama energija.

LSSP ir atitinkami DTLS protokolai, kuriuose naudojami identiški kibernetinio saugumo algoritmai, buvo įgyvendinti naudojant "Java" kriptografines bibliotekas. Šio žingsnio buvo imtasi siekiant užtikrinti skirtingų protokolų palyginimo pagrįstumą, kartu užtikrinant identiško lygio perduodamų duomenų saugumą. Šio eksperimento tikslas buvo išmatuoti ir įvertinti energijos sąnaudų skirtumą tarp skirtingų protokolų, atliekant empirinius matavimus.

Siekiant palyginti visus tiriamus protokolus, iš srautinio perdavimo įrenginio buvo perduota 10 K duomenų, o serverio įrenginyje naudojant "Wireshark" programinę įrangą priimti paketai. Tada apskaičiuotas bendras belaidžiais kanalais perduotų duomenų kiekis. 11 pav. pavaizduoti gauti rezultatai.



11 pav. Eksperimento rezultatai duomenų perdangai palyginti

Šioje diagramoje nagrinėjami tik paketai, kurie buvo perduoti iš perdavimo įrenginio, ir neatsižvelgiama į DTLS protokole naudojamus rankų suvedimo paketus, kurie buvo perduoti iš serverio atgal į perdavimo įrenginį.

Lygindami gautus rezultatus galime patvirtinti, kad LSSP M1 modifikacija nesukuria papildomų duomenų ir yra lygiavertė standartiniam UDP protokolui. LSSP M2 ir M3 režimuose papildomi paketai naudojami ECC informacijai perduoti, todėl, palyginti su M1 versijos visais standartiniais UDP perdavimais, pridėtinės išlaidos sudaro 25%. $DTLS2$ ir $DTLS3$ protokolo versijos atlieka rankų valdymo funkciją (etiketė – DTLS-HS) ir kiekviename perduodamame duomenų pakete pateikia pridėtinių duomenų, susijusių su DTLS protokolu (etiketė – DTLS).

V. IŠVADOS

1. Sukurtas paskirstytas lokalizacijos metodas, pagrįstas RTT diapazono matavimais, skirtas *ad hoc* daiktų interneto tinklams. Paskirstytasis metodo pobūdis leidžia jį sklandžiai diegti neturint išankstinių žinių apie daiktų interneto tinklų mazgų buvimo vietą. Patikrinus modeliavimo būdu, paaiškėjo, kad jis gali veikti su visiška arba daline tinklo konvergencija, kai tinklo kišenėse gaunamas absoliutus padėties nustatymo sprendimas, o izoliuotuose arba neturinčiuose pakankamo ryšio tinkluose veikia vietinio atskaitos rėmo režimu. Atlikus 8 iteracijų su tiesinėmis trajektorijomis, imituojamas tinklas pasiekė 74% konvergavimo lygį, o vidutinė padėties nustatymo paklaida – $1,9m$. Įvedus virtualius tinklo mazgus, absoliutaus padėties nustatymo sprendinys gali būti skleidžiamas laike, kai yra galimybė atlikti matavimus.

2. Sukurta specializuota hibridinė valdymo sistema, skirta energijos suvartojimui lokalizacijos ir ryšio užduotims valdyti. Sistemos patikrinimai rodo,

kad ji gali paskirstyti turimą galią, atsižvelgdama į numatomą suvartojamą galią, esamą akumuliatoriaus įkrovos lygį ir turimų RTT matavimų kokybę bei skaičių. Siūlomas Kalmano filtras gali paskirstyti galią 24 val. trukmės langui su 50% baterijos atsargų, kurias galima naudoti kitą dieną. Neraiškiosios logikos valdiklis vidutiniškai parinko 4 NN matavimams, o jų nuokrypiai nuo nuotolio įvertinimo per 900 imčių sumažėjo vidutiniškai iki $0,1$ m.

3. Dvi daugiamodalaus identifikavimo ir autentiškumo nustatymo metodo versijos: Siūlomos dvi dvi modifikavimo ir identifikavimo bei identifikavimo sistemos. *MMIA1* leidžia identifikuoti ir autentifikuoti naudojant skaitmeninį parašą ir MAST medžio geplokacinę šaknies maišą. Jį galima gauti tris kartus pritaikius maišos funkciją ir vieną kartą pritaikius šifravimo funkciją sugeneruotai maišai. *MMIA2* papildomai apima daiktų interneto tinklo mazgų, kurie buvo naudojami lokalizavimo sprendimui gauti, ID ir goelokacines maišas. Abiejų pasiūlytų metodo variantų rezultatas yra tokio paties dydžio *Root hash*, tačiau antrajame variante DI tinklo mazgui reikia atlikti papildomus maišos skaičiavimus. Patvirtinimo etape *MMIA2* sėkmingai pašalino pavyzdžius, kurie geolokacijos sprendimui generuoti naudojo nepageidaujamus NN 100% laiko.

4. Sukurtas ir pasiūlytas lengvas saugus srautinio duomenų perdavimo protokolas, skirtas duomenų srautiniam perdavimui iš daiktų interneto įrenginio. Apibrėžti trys saugumo režimai. 1 režimas užtikrina duomenų šaltinio autentiškumo patvirtinimą. 2 režimas užtikrina duomenų vientisumą. 3 režimas prideda konfidencialumo funkcijas. Atlikus empirinius eksperimentus nustatyta, kad M1 protokolo našumas yra palyginamas su UDP protokolo našumu su autentiškumo nustatymo funkcijos priedu, o perdavimo laikas dėl apdorojimo reikalavimų skiriasi $0,5$ s ir reikalauja nedidelio $2,5\%$ bendros galios padidėjimo. Siūlomas M2 protokolas, palyginti su DTLS2, pasižymi panašiu našumu ir panašiomis šaltinio autentiškumo nustatymo savybėmis, o jo bendra galia padidėja 3%. Siūlomas M3 protokolas, palyginti su DTLS3, yra lėtesnis dėl papildomo duomenų paketo su ECC informacija. Palyginti su DTLS3, M3 protokolui reikia papildomai 18% visos energijos.

5. Siūlomų metodų derinys pateikia sprendimų rinkinį, kurį galima sujungti į praktinį vietos nustatymu pagrįstą daugiamodalų identifikavimo ir autentifikavimo metodą, skirtą daiktų interneto *ad hoc* tinklams. Šiame etape ribojantis veiksnys yra RTT nuotolio matavimų tikslumas, kurį galima pasiekti dabartiniais metodais ir technine įranga. Būtina atlikti papildomus galimų sprendimų, kaip padidinti turimą matavimų tikslumą, tyrimus, tačiau jie nepatenka į šio darbo apimtį.

LITERATŪRA

BAGDONAS, Kazimieras; BORRE, Kai, 2008. Ubiquitous WiFi/GNSS positioning system-TOA based distance estimation. In: *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, p. 1773–1779.

BAGDONAS, Kazimieras; JUSAS, Nerijus; VENCKAUSKAS, Algimantas, 2017. A Converging Distributed Positioning Algorithm for Internet-of-Things. *Elektronika ir Elektrotechnika*. T. 23, Nr. 6, p. 72–76.

BAGDONAS, Kazimieras; SCHIØLER, Henrik; BORRE, Kai, 2009. Range estimation for indoor positioning via drifting clocks. In: *Proceedings of the 22nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2009)*, p. 516–526.

BOUNCY CASTLE INC., Legion of the, 2018. *BC-FJA (Bouncy Castle FIPS Java API) User Guide.* [`https://downloads.bouncycastle.org/fips-java/BC-FJA-SecurityPolicy-1.0.0.pdf`]. Accessed: 2018-03-08.

GREWAL, Mohinder S; ANDREWS, Angus P, 2014. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons.

HAMMING, Richard W, 1950. Error detecting and error correcting codes. *The Bell system technical journal*. T. 29, Nr. 2, p. 147–160.

HEMMINGER, Stephen ir kt., 2005. Network emulation with NetEm. In: *Linux conf au*. Citeseer. T. 5, p. 2005.

ISHENGOMA, Fredrick R, 2014. The art of data hiding with reed-solomon error correcting codes. *arXiv preprint arXiv:1411.4790*.

Legion of the Bouncy Castle Inc. Java (D)TLS API and JSSE Provider. User Guide. 2018 [`https://downloads.bouncycastle.org/fips-java/BC-FJA-(D)TLSUserGuide-1.0.3.pdf`]. Accessed: 2018-03-08.

Performance of checksums and CRCs over real data, 1998. *IEEE/ACM Transactions on Networking*. T. 6, Nr. 5, p. 529–543.

REED, Irving S; SOLOMON, Gustave, 1960. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*. T. 8, Nr. 2, p. 300–304.

TEAM, The Tcpdump, [s. a.]. *Tcpdump & Libpcap.* [`http://www.tcpdump.org/`]. Accessed: 2018-03-08.

TECHNOLOGIES, Sly, 2018. *jNetPcap API.* [`http://jnetpcap.com`]. Accessed: 2018-03-08.

VENČKAUSKAS, Algimantas; JUSAS, Nerijus; MIKUCKIENĖ, Irena; MACIULEVIČIUS, Stasys, 2012. Generation of the secret encryption key using the signature of the embedded system. *Information technology and control*. T. 41, Nr. 4, p. 368–375.

VENČKAUSKAS, Algimantas; MORKEVICIUS, Nerijus; BAGDONAS, Kazimieras; DAMAŠEVIČIUS, Robertas; MASKELIŪNAS, Rytis, 2018. A lightweight protocol for secure video streaming. *Sensors*. T. 18, Nr. 5, p. 1554.

XIE, Haijiang; ZHAO, Jizhong, 2015. A lightweight identity authentication method by exploiting network covert channel. *Peer-to-Peer Networking and Applications*. T. 8, Nr. 6, p. 1038–1047.

ZANDER, Sebastian; ARMITAGE, Grenville; BRANCH, Philip, 2007. A survey of covert channels and countermeasures in computer network protocols. *IEEE Communications Surveys & Tutorials*. T. 9, Nr. 3, p. 44–57.

UDK 004.056+004.738.5](043.3)

SL 344. 2022-07-22, 3 leidyb. apsk. l. Tiražas 42 egz. Užsakymas 146.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas