**KAUNO TECHNOLOGIJOS UNIVERSITETAS**
INFORMATIKOS FAKULTETAS
INFORMACIJOS IR INFORMACINIŲ TECHNOLOGIJŲ SAUGA (6211BX008)

MARIUS TAPARAUSKAS

# TRANSPORTO PRIEMONIŲ LOKALAUS BELAIDŽIO TINKLO SAUGIOS KOMUNIKACIJOS METODAS

Baigiamasis magistro projektas

Vadovas
Prof. A. Venčkauskas

Kaunas, 2023

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**
INFORMATIKOS FAKULTETAS
INFORMACIJOS IR INFORMACINIŲ TECHNOLOGIJŲ SAUGA (6211BX008)

MARIUS TAPARAUSKAS

# TRANSPORTO PRIEMONIŲ LOKALAUS BELAIDŽIO TINKLO SAUGIOS KOMUNIKACIJOS METODAS

Baigiamasis magistro projektas

Vadovas
Profesorius A. Venčkauskas
(parašas) (data)

Recenzentas
Docentas T. Adomkus
(parašas) (data)

Studentas
M. Taparauskas
(parašas) (data)

Kaunas, 2023

**Kauno technologijos universitetas**

Informatikos fakultetas

Marius Taparauskas

# Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodas

## Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

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

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

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

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

Marius Taparauskas

*Patvirtinta elektroniniu būdu*

Taparauskas, Marius. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodas. Magistro krypties studijų baigiamasis projektas, vadovas prof. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informacijos ir informacinių technologijų sauga.

Reikšminiai žodžiai: belaidžio tinklo sauga, transporto priemonių tinklo sauga, „Bluetooth“ sauga, HMAC, HKDF, AES.

Kaunas, 2023. 90 p.

## Santrauka

Tobulinant transporto priemones ir jų sistemas, inžinieriai integruoja kompiuterinius ir išmanius modulius, kurių tikslas stebėti ir valdyti transporto priemonės veikimą ar transportuojamo krovinio būklę. Šie moduliai ir sistemos tampa patraukliu objektų įvairiems nusikaltėliams, kurie gali sukelti žalą transporto priemonei, vairuotojui, keleiviams ir pašaliniams žmonėms ir jų turtui. Todėl transporto priemonių sauga yra itin svarbus aspektas šiandieninėje visuomenės infrastruktūroje.

Šio darbo tikslas – suprojektuoti saugų komunikacijos metodą, kuris būtų taikomas transporto priemonėse ir palyginti suprojektuotą metodą su rinkoje esančiais sprendimais.

Darbe išanalizuotos transporto priemonėms kylančios saugumo problemos, naudojamos komunikacijos technologijos, duomenų apsaugojimo metodai. Saugaus komunikacijos metodo panaudojimui ir tyrimui nuspręsta realizuoti demonstracinį transporto priemonės tinklą naudojantį suprojektuotą komunikacijos saugumo metodą. Šiame tinkle komunikuoja „ESP32“ mikrovaldikliai naudodami „Bluetooth Low Energy“ komunikacijos technologiją. Tam, kad būtų galima ištirti suprojektuoto metodo efektyvumą ir naudojimo galimybes, buvo specifikuota tyrimo metodika ir tyrimo parametrai.

Siūlomo sprendimo prototipe yra vienas įrenginys atsakingas už kriptografinių sesijos raktų generavimą ir paskirstymą. Taip pat, šis įrenginys siunčia duomenų užklausas kitiems tinklo mazgams. Kriptografinis sesijos raktas yra 128 bitų ilgio ir keičiamas periodiškai. Kriptografiniam sesijos raktui apskaičiuoti naudojama HKDF funkcija, o duomenų šifravimui naudojama „AES-CBC“ algoritmą. Kiekviena žinutė tinkle turi autentifikacijos reikšmę (MAC), kuri apskaičiuojama HMAC funkcija. Kiekvienas tinkle komunikuojantis įrenginys patikrina žinutės autentifikacijos reikšmę prieš atliekant darbus su šifruotais duomenimis. Atlikus transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo, projektuoto šiame darbe, tyrimus ir palyginimus su rinktoje esančiais TLS ir DTLS metodais pastebėta, kad siūlomas sprendimas yra efektyvesnis energijos sąnaudomis, atminties išteklių sąnaudomis ir sparta laiko atžvilgiu. Tačiau, šiems rezultatams pasiekti yra naudojama silpnesnė tinklo kriptografinė apsauga. Eksperimentų ir tyrimų metu gauti rezultatai išanalizuoti. Pateiktos siūlomo saugios komunikacijos metodo išvados.

Taparauskas, Marius. A Method for Secure Communication of Vehicular Local Area Wireless Networks. Master's Final Degree Project, supervisor prof. Algimantas Venčkauskas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Information and Information Technology Security.

Keywords: wireless network security, vehicular network security, „Bluetooth“ security, HMAC, HKDF, AES.

Kaunas, 2023. 90 pages.

## Summary

During the development of vehicles and their systems, engineers integrate computers and other intelligent modules aimed at monitoring and controlling the performance of the vehicle or the condition of the transported cargo. These modules and systems become attractive objects for various criminals who can cause damage to the vehicle, the driver, passengers and outsiders or their property. Therefore, vehicle safety and security is an extremely important aspect of society and today‘s public infrastruture.

The aim of this work is to design a secure communication method that would be applied in vehicles and to comapre the designed method with solutions available on the market.

The paper analyzes security problems arising for vehicles, used communication technologies and data protection methods. For the use and research of the secure communication method, it was decided to implement vehicle network prototype which uses the designed comunication security method. Multiple „ESP32“ microcontrollers are communicating in this network using „Bluetooth Low Energy“ communication technology. In order to investigate the effectiveness and usability of the designed method, the research methodology and research parameters where specified in this paper.

In the prototype of the proposed solution, there is one device responsible for generating and distributing cryptographic session keys. Also, this device sends data requests to other network nodes. The cryptographic session key is 128 bits in length and is changed periodically. The cryptographic session key is calculated using HKDF function and the data is encrypted using „AES-CBC“ algorithm. Each message on the network has the message authentication code (MAC) which is calculated using HMAC function. Each device communicating in the network checks the message authentication code before performing any operations with encrypted data. After completing research and comparing the secure communication of vehicular local area wireless network method with TLS and DTLS methods, it was observed that the proposed solution is more efficient in terms of energy consumption, memory resource consumption and speed in terms of time. However, weaker network cryptographic protection is used to achieve these results. The results obtained during the experiments and the research are analyzed and the conclusions of the proposed secure communication method are presented.

**Turinys**

## Lentelių sąrašas

## Paveikslų sąrašas

## Santrumpų ir terminų sąrašas

**Santrumpos:**

1. SRAM (Static random-access memory) – įrenginio atmintis, kurią galima perrašyti.
2. ROM (Read Only Memory) – įrenginio atmintis, kurios negalima perrašyti programos veikimo metu.
3. RTC (Real Time Clock) – realaus laiko laikrodis.
4. GPIO (General Purpose Input/Output) – fizinės jungtys palaikančios skaitmeninio signalo nuskaitymą ar perdavimą tarp išorinių įrenginių.
5. SPI (Serial Peripheral Interface) – sinchroninės serijinės trumpo nuotolio komunikacijos protokolas.
6. I2C (dar žinomas kaip IIC – Inter-Integrated Circuit) – sinchroninės daug valdiklių - daug valdomų įrenginių (master/slave) serijinės trumpo nuotolio komunikacijos protokolas.
7. I2S (Inter-IC Sound) – elektrinės serijinės magistralės sąsajos standartas skirtas sujungti skaitmeniniams garso įrenginiams.
8. UART (Universal asynchronous receiver-transmitter) - prietaisas, kuris verčia lygiagrečius duomenų bitus į nuoseklius duomenų bitus.
9. AES (Advanced Encryption Standart) – blokinis simetrinis kriptografinis algoritmas.
10. GATT (General Attribute profile) – „Bluetooth“ „Bluedroid“ palaikoma komunikacijos ir duomenų perdavimo specifikacija kiekvienam kliento ar serverio komunikacijos sąsajai atskirai suteikiant galimybę aprašyti ir sudaryti daugiau nei vieną „Bluetooth“ komunikaciją tarp įrenginių.
11. TLS (Transport Layer Security) – kriptografinis protokolas, numatantis apsaugotą duomenų perdavimą tarp mazgų pasauliniame kompiuterių tinkle internete. Pagrindinis TLS protokolo tikslas yra suteikti privatumą ir duomenų integralumą tarp dviejų bendraujančių kompiuterinių programų.
12. DTLS (Datagram Transport Layer Security) – komunikacijos protokolas užtikrinantis „datagrama“ pagrįstų programų saugumą, apsaugant nuo pasiklausymo, klastojimo ar pranešimų klastojimo. Veikimas paremtas TLS protokolu.
13. HMAC (Hash based Message Authentication code) – kriptografinė autentifikacijos funkcija naudojanti santrauką ir slaptą raktą.
14. HKDF (HMAC based Key Derivation Function) – kriptografinių raktų išvedimo funkcija, kuri apskaičiuoja naują raktą naudojant slaptą informaciją.
15. AES (Advanced Encryption Standart) – duomenų šifravimo algoritmų šeima, kuriai priklauso daug skirtingų AES šifravimo algoritmo variacijų.
16. Tekstograma – nešifruoti duomenys.
17. Šifrograma – šifruoti duomenys.

## Įvadas

Šiandien transporto priemonių kibernetinis saugumas tampa vis svarbesnis. Taip yra, nes kompiuterinės technologijos tobulėja sparčiai ir yra plačiai integruojamos į transporto priemones. Informacinėmis technologijomis sukuriami kompleksiški transporto priemonių lokalūs tinklai, kurie atsakingi už itin svarbius sistemos darbus. Tarp šių darbų įeina stabdymo, apšvietimo, variklio valdymo ir kitų sistemų valdymas pagal jutiklių pateikiamus duomenis. Todėl informacinių technologijų, naudojamų transporto priemonėse, sukompromitavimo keliamos problemos, kelia pavojų ir transporto priemonių keleivių ir pėsčiųjų sveikatai. Taip pat, transporto priemonėse pradedant naudoti belaidžius jutiklius, įsilaužimo į lokalų tinklą, galimybė didėja. O naudotojui suteikiant galimybę prijungti išmaniuosius įrenginius prie transporto priemonės lokalaus tinklo, atveria papildomus kelius prie sistemos. Tokios problemos kelia pavojų naudotojų privačiai informacijai. Tačiau, saugumo sprendimai belaidžiuose jutikliuose nėra stiprūs arba visai nenaudojami, o transporto priemonių keliami apribojimai įrangos dydžiui ir energijos sąnaudoms papildomai kenkia lokalaus tinklo saugumui. Todėl, šiandien reikia metodo, suteikiančio saugumą transporto priemonių lokaliam tinklui atsižvelgiant į ribotų išteklių aplinką.

Šio projekto tikslas yra suprojektuoti, realizuoti ir išanalizuoti transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodą.

Tikslui pasiekti reikia atlikti šiuos uždavinius:

1. Atlikti transporto priemonių lokalaus tinklo saugumo problemos, komunikacijos technologijų, autentifikacijos ir šifravimo algoritmų analizę.
2. Suprojektuoti transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodą.
3. Realizuoti transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodą.
4. Išanalizuoti ir ištirti transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodą.

# 1. Transporto priemonių lokalaus belaidžio tinklo saugumo analizė

Šiame skyriuje atliekama transporto priemonių lokalaus belaidžio tinklo saugumo problemos analizė. Analizuojamos belaidžių tinklų įterptinėse sistemose problemos, trūkumai ir galimos atakos. Analizuojami įrenginių autentifikavimo ir autorizavimo metodai, kurie tiktų ribotų išteklių sistemoje. Taip pat analizuojami saugūs duomenų siuntimo belaidžiu ryšiu metodai ribotų išteklių sistemoms.

## 1.1. Transporto priemonių lokalaus belaidžio tinklo saugumo problemos analizė

Šiandien, tobulėjant kompiuterinėms technologijoms, transporto priemonės, jose esantys jutikliai ir valdikliai yra stipriai priklausomi nuo kompiuterinių technologijų. Populiarėjant belaidėms technologijoms, transporto priemonių jutiklius galima gaminti belaidžius, taip palengvinant transporto priemonės surinkimą, kadangi nereikia rūpintis laidų išvedžiojimu [1]. Taip pat belaidžių jutiklių naudojimas palengvina ir taisymą, kadangi sutrikus jutiklio veikimui nereikia rūpintis jutiklio laidų kokybe. Šios technologijos suteikia ir papildomų galimybių, pavyzdžiui padangų oro slėgio jutikliai, kurie siunčia duomenis belaidžiu tinklu. Be to, šios technologijos suteikia patogumų naudotojui, kuris gali su mobiliuoju įrenginiu prisijungti prie transporto priemonės sistemos tam, kad valdyti suteikiamas programas.

Pagrindine problema yra kompiuterinio skaičiavimo galios trūkumas, kadangi transporto priemonė yra ilgaamžė ir turi atlaikyti aukštas temperatūras ir vibracijas palyginus su įprastais stacionariaisiais ar nešiojamaisiais kompiuteriais [2]. Taip pat kompiuterinė sistema transporto priemonėse yra mažų matmenų ir turi ribotą maitinimo šaltinį, dėl to mažėja skaičiavimo galia. Visi šie faktoriai padidina tikimybę, kad transporto priemonės sauga bus nulaužta. Taip yra todėl, kad ribotų išteklių sistemose naudojant pakankamai ilgus kriptografinius raktus ar sudėtingus kriptografinius algoritmus, sistemos veikimas sulėtinamas ir sistema gali veikti per lėtai. Taip pat, transporto priemonių tinklas gali būti atakuojamas naudojant daugybę skirtingų atakų, naudojant belaides komunikavimo technologijas ar transporto priemonės vidinę įrangą. Tai sukelia grėsmę naudotojų sveikatai, kadangi kelių jutiklių duomenų pakeitimas ar žinučių modifikavimas gali pakenkti transporto priemonės veikimui, sukeliant grėsmę keleiviams ir pėstiesiems [2].

Taip pat verta paminėti, kad transporto priemonių tinklas nėra atnaujinamas kaip įprastų kompiuterinių sistemų, todėl atsiradus naujoms atakoms ar atradus saugumo spragų esamuose sprendimuose, nėra kaip šias klaidas pašalinti atnaujinant programinę įrangą. Be to, transporto priemonių vidinis tinklas yra itin kompleksiškas, šį tinklą sudaro daug skirtingų posistemių, kurios turi savas saugumo spragas. Tobulėjant naudotojų išmaniesiems įrenginiams, transporto priemonėse, suteikiama galimybė prisijungti prie vidinio tinklo, kas sukelia papildomų saugumo spragų [3]. Esant silpnai naudotojo įrenginių ir transporto priemonės vidinio tinklo komunikacijos apsaugai, tinklas gali būti pažeistas iš išorės ar pačio naudotojo įrenginio. Taip pat naudotojo įrenginiui gali būti suteikiama prieiga prie viso transporto priemonės vidinio tinklo neizoliuojat specifinių tinklo funkcijų. Tai sukelia grėsmę ne tik naudotojo privatiems duomenims, bet ir sveikatai, įgijus prieigą ir kontrolę prie transporto priemonės tinklo valdomų jutiklių.

Transporto priemonėse naudojami jutikliai ir valdikliai, kurie dirba belaidžiu tinklu dideliais atstumais, sukelia papildomų saugumo spragų. Įrenginiai, kaip GPS ar GSM, gali būti apgaunami jiems siunčiant nekorektiškus duomenis. Taip pat, šie įrenginiai gali suteikti nuotolinę prieigą prie tinklo neautorizuotiems naudotojams. O transporto priemonės jutikliai, siunčiantys duomenis

belaidžiu ryšiu, suteikia prieigą prie transporto priemonės tinklo atliekant ataką naudojant šių jutiklių tinklą [4].

Transporto priemonėje jutikliams komunikuojant naudojant belaidį tinklą iškyla daugybė saugumo problemų. Kadangi jutiklių duomenys yra siunčiami visomis kryptimis, su specialia įranga ir pakankamai artimu atstumu, šiuos duomenis galima stebėti neautorizuotam objektui. Problema ypač pavojinga, jeigu jutiklių duomenys nėra šifruojami. Tokiu atveju, atliekant atvirkštinę inžineriją, galima sukompromituoti sistemos veikimą tinklui siunčiant neteisingus duomenis, apsimetant autorizuotais jutikliais. Jeigu tinklo posistemės nėra izoliuotos ir komunikuoja tik su vienu centriniu įrenginiu, visa sistema gali būti paveikta atakuojančio objekto. Taip pat, jeigu sistema priima bet kokius įrenginius nereikalaujant specifinės autorizacijos, išoriniam objektui itin lengva įsilaužti į lokalų tinklą. Dauguma problemų gali būti išspręstos lokalų transporto priemonės tinklą paskirsčius į atskiras posistemes, kuriuos viena su kita komunikuotų šifruotu ir autorizuotu komunikacijos tuneliu, kurio kriptografijos raktai būtų periodiškai keičiami.

### 1.2. Projekto įgyvendinimui keliami reikalavimai

Atlikus problematikos analizę pastebėta, kad vidiniam lokaliam transporto priemonės tinklui kyla nemažai saugumo problemų, ypač kai lokalus transporto priemonės tinklas turi belaidžių komunikacijos technologijų posistemes. Kadangi transporto priemonės lokalus tinklas dirba išteklių ribotoje sistemoje, projektui keliami reikalavimai atsižvelgiant į tai, kad sistema negali suteikti itin daug skaičiavimo galios kriptografiniams algoritmas.

Projekto įgyvendinimui iškeliami šie reikalavimai:

1. Komunikacijos technologija turi būti belaidė ir trumpo nuotolio. Atsižvelgiant į tai, kad projekto metu sprendžiamos problemos susijusios tik su lokaliu transporto priemonės tinklu, belaidės komunikacijos technologijos turi veikti tik trumpu nuotoliu. Tai sumažina tikimybę, kad tinklui bus pakenkta naudojant išorinį, transporto priemonės tinklui, objektą.

2. Komunikacijos technologijos veikimas turi būti izoliuojamas transporto priemonės lokaliame tinkle. Jeigu jutikliui būtina komunikuoti su išoriniu objektu, kaip GPS jutikliui su palydovu, komunikacijos technologija turi suteikti galimybę šią sistemos dalį atskirtį į atskirą izoliuotą posistemę. Taip pat tinklas neturi turėti jokių išorinių objektų tinklo veikimui.

3. Autentifikacijos, autorizacijos ir šifravimo algoritmai turi būti lengvi skaičiavimo galios atžvilgiu, tačiau turi užtikrinti duomenų saugumą, vientisumą ir konfidencialumą. Taip pat naudojamas metodas turi užtikrinti periodišką kriptografijos modifikavimą, kuris apsaugotų nuo tinklo atakų, kurių metu naudojami raktai ar kita informacija būtu rasta ar atkuriama.

### 1.3. Belaidėms technologijoms taikomų atakų analizė

Šiandien egzistuoja daugybė skirtingų belaidžio ryšio technologijų, kurios gali būti naudojamos transporto priemonių komunikacijai su naudotojo įrangą, kitomis transporto priemonėmis ar aplinkine infrastruktūra. Šioms technologijoms egzistuoja skirtingos saugos problemos ir atakos. Šiame poskyryje aptariamos šios atakos, kurios skirstomos į transporto priemonės vidinio tinklo atakas, transporto priemonės su viskuo tinklo atakos, transporto priemonės su kita transporto

priemone tinklo atakos ir kitos atakos. Šio projekto analizė atliekama transporto priemonės vidinio, lokalaus tinklo problemoms ir galimoms atakoms aptarti.

### 1.3.1. Nuotolinė transporto priemonės jutiklių ataka

Transporto priemonės skirtingi elektroniniai komponentai, tokie kaip kameros, ultragarsiniai radarai ir skirtingi jutikliai yra sujungti vieno transporto priemonės lokalaus tinklo. Kiekvienas jutiklis ar kitas elektroninis komponentas turi savo stipriąją pusę, tačiau taip pat ir silpnąją pusę, atkreipiant dėmesį į komponento veikimo atstumą, aptikimo pajėgumą ir komponento patikimumą. Taip pat, naujesni jutiklių ir komponentų modeliai gali suteikti belaidę prieigą išoriniam subjektui. Tai suteikia galimybę perimti jutiklių informaciją ir pateikti blogą informaciją, kuri nebūtų atpažinta kaip neteisingą ar iš išorės pateikta [4].

### 1.3.2. GPS klastojimo ataka

Pagrindinė transporto priemonės navigacijos sistema paremta GPS technologija, kuri gauną transporto priemonės lokacijos informaciją ir laiką. Šios atakos metu, išorinis subjektas pateikia, stipriu GPS signalu, netikrus GPS duomenis. Kadangi duomenis yra pateikiami stipresniu nei tikruoju palydovų signalu, GPS imtuvas ignoruoja tikrąjį signalą. Šios atakos metu galima nukreipti transporto priemonę bloga trajektorija, kas itin pavojingą automatizuotoms transporto priemonėms [4].

### 1.3.3. Artimo nuotolio pažeidžiamumų atakos

Artimo nuotolio pažeidžiamumai yra sukeliamai trumpo nuotolio komunikacijos mechanizmų, kurie veikia transporto priemonės lokaliame tinkle. Tokios silpnybės gali būti sukeltos veikiančio „Bluetooth“ tinklo, padangų oro slėgio stebėjimo sistemų ar be rakto veikiančio užrakto ir užvedimo mechanizmo sistemos. Šiuo metu žinoma, kad „Bluetooth“ technologija nėra saugi nuo atminties atakų, kurių metu galima paleisti išorinį kodą iš susieto „Bluetooth“ įrenginio. Šiuo atveju pažeistas įrenginys gali atakuoti transporto priemonės variklio valdymo informaciją. Šios atakos metu, transporto priemonės naudotojas nepastebi vykdomos atakos. Nors šiandiena yra pasiūlyti skirtingi kriptografiniai algoritmai „Bluetooth“ saugumui užtikrinti, jie yra neefektyvūs, todėl nėra taikomi komerciniuose produktuose. Taip pat šie algoritmai neišsprendžia atminties atakos problemos [4].

Padangų oro slėgio stebėjimo sistemos paremtos paprastais protokolais, kurie nėra paremti kriptografiniais metodais. Todėl, šios sistemos gali būti išanalizuojamos naudojant atvirkštinę inžineriją. Taip pat šios sistemos yra pažeidžiamos klastojimo ir baterijos eikvojimo atakų. Naudojant itin pigią ir lengvai prieinamą įrangą, šių sistemų siunčiamos žinutės gali būti pasiekiamos iki 10 metrų atstumu, o naudojant žemo triukšmo stiprintuvą (low noise amplifier), žinutės gali būti gaunamos iki 40 metrų atstumu. Tai suteikia galimybę rinkti duomenis ir juos klastoti. Taip pat, kiekvienas padangoje laikomas jutiklis turi savo išskirtinį identifikatorių, šį identifikatorių galimą panaudoti aptikti ar sekti transporto priemonės lokaciją [4].

Be raktų veikianti prieigos sistema gali būti blokuojama išorinio subjekto naudojant specialią įrangą, taip neleidžiant užrakinti transporto priemonės. Tokia įranga gali būti paslėpta transporto priemonių stovėjimo aikštelėse, blokuojant daugiau nei vieną transporto priemonę. Taip pat be raktų veikianti sistema yra pažeidžiama kopijavimo atakos, kurios metu transporto priemonės atrakinimo signalas gali būti nukopijuojamas ir panaudojamas gauti prieigai prie transporto priemonės [4].

Valdiklio srities tinklo (CAN - Controler Area Network) pagrindinė problema yra autentifikacijos ir šifravimo trūkumas. Prie šio tinklo be autorizacijos gali prisijungti nauji mazgai, kurie gauna prieigą prie visų CAN tinklo žinučių, kurios nėra šifruojamos. Taip atakuojantis įrenginys gali rinkti asmeninę transporto priemonės naudotojo informaciją: mobilaus telefono numerį, lokaciją, saugomus adresus ir kita. Taip pat, atakos metu tinklui galima pateikti CAN žinutes. Kadangi šiame tinkle pirmenybė taikoma įrenginiui su didžiausiu prioritetu, atakos metu įterptas mazgas gali atlikti DOS ataką, kol įrenginys nenutraukiamai dirba aukščiausiu prioritetu [4].

Pakartojimo atakos metu, transporto priemonei yra atsiunčiami paketai, kurie buvo nuskaityti ar sudaryti specialiomis sąlygomis. Paketai gali būti pagaminti iš anksto ar sukuriami komunikacijos metu atsižvelgiant į dabartinę situaciją. Tokie paketai gali pateikti informaciją, kad priešais transporto priemonę esantis objektas stabdo ar atlieka kitą veiksmą, kuris neatitinka realios situacijos, taip priverčiant transporto priemonės sistemą reaguoti į naujai gautą informaciją [4].

Naudotojo įrenginiai taip pat gali būti panaudoti transporto priemonės lokalaus tinklo atakai. Jeigu naudotojo įrenginys yra užkrėstas ir naudotojas šį įrenginį sujungia su transporto priemonės lokaliu tinklu, iš šio naudotojo įrenginio gali būti atliekama ataka. Nuo šios atakos apsisaugoti ypač sunku, kadangi naudotojas, pridedant prie tinklo, patvirtino įrenginį kaip saugų.

Transporto priemonėje veikiantis „Wi-Fi“ tinklas gali būti atakos objektas. Dažniausiai „Wi-Fi“ SSID ir slaptažodžiai nėra šifruojami, kadangi naudotojai neatlieka specifinių konfigūracijų „Wi-Fi“ prieigos taške. Tai suteikia galimybę imituoti tinklą išoriniam subjektui, prie kurio prisijungia transporto priemonės naudotojai. Naudojant tokią „Evil Twin“ ataką galima stebėti naudotojų veiksmus. Taip pat tokie tinklai gali būti atakuojami DOS, radijo bangų trukdymo, pakartojimo, žmogaus viduryje (man in the middle) ir kitų atakų [2].

### 1.4. Duomenų saugumui pažeisti naudojamų atakų analizė

Tinkle siunčiami duomenys gali būti pažeidžiami naudojant skirtingas atakas, bandant atspėti naudojamą kriptografinį raktą ar išnaudojant žinomas šifravimo algoritmų silpnybes. Šiame poskyryje analizuojamos atakos, kurios gali būti pritaikomos šifruotų duomenų saugumui pažeisti.

#### 1.4.1. Žinomos ir pasirinktinos tekstogramos atakų analizė

Žinomos tekstogramos ataka yra kriptoanalizės modelis, kurio metu yra žinoma tekstograma ir šifrograma po tekstogramos šifravimo. Naudojant šiuos žinomus parametrus galima atkurti kitus slaptus parametrus, kritpografinius raktus ar kodų knygas lyginant ir ieškant skirtingų sutapimų tarp skirtingų šifrogramų ir tekstogramų [5].

Pasirinktinos tekstogramos ataka yra kriptoanalizės modelis, kurio metu galima gauti šifravimo algoritmo sukurtą šifrogramą. Taip pat, kaip žinomos tekstogramos atakoje, kriptografiniai raktai gali būti atkuriami lyginant skirtingas šifrogramas ir ieškant sutapimų [5].

#### 1.4.2. Susijusio rakto atakos analizė

Susijusio rakto ataka, yra kriptoanalizės modelis, kurio metu galima stebėti šifravimo algoritmo atliekamas operacijas naudojant keletą skirtingų raktų. Šių raktų reikšmės gali būti nežinomos, tačiau jas turi sieti koks nors matematinis sąryšis. Pavyzdžiui, atakos metu naudojamų raktų paskutiniai ar

pirmieji kriptografinio rakto bitai yra visada vienodi. Naudojant tokią ataką šifravimo algoritmo sukurtoje šifrogramoje gali kartosis tam tikra struktūra [6].

### 1.4.3. Slydimo atakos analizė

Slydimo ataka yra vienas iš kriptoanalizės metodų, skirtų paneigti idėją, kad didinant šifravimo ciklų skaičių, bet koks šifravimo algoritmas gali tapti saugiu ir atspariu diferencialinėms atakoms. Ši ataka dirba taip, kad ciklų skaičius tampa beprasmis. Slydimo atakos metu analizuojamas raktų valdymo mechanizmas ir bandoma išnaudoti šio mechanizmo silpnybes. Vienas iš dažniausiai pritaikomų atakos tipų yra raktų pakartojimas ciklu. Tam, kad slydimo ataką būtų galima sėkmingai pritaikyti šifravimo algoritmo nulaužimui, reikia, kad šifravimo algoritmas naudotų tą pačią funkciją visais šifravimo ciklais. Taip pat, ši funkcija turi būti pažeidžiama ir žinomos tekstogramos atakos, tai smarkiai prisideda prie sėkmingo atakos įvykdymo [7].

### 1.4.4. Kubo atakos analizė

Kubo (cube) ataka leidžia apskaičiuoti kintamų polinomų (daugianarių) sudėtinius skaičius, kurie buvo sudauginti, sudėti, atimti ar pakelti natūrinio skaičiaus laipsniu. Kubo atakos smarkiai pasižymi apskaičiuojant atsitiktinių skaičių polinomus. Taip pat, ši ataka gali būti pritaikoma bet kokiam blokiniam šifrui ar srautiniam šifrui tol, kol bent vienas išvesties bitas gali būti atvaizduojamas kaip mažo laipsnio polinomas slaptuose ir viešuose kintamuosiuose parametruose [8].

## 1.5. Belaidžių komunikacijų technologijų naudojamų transporto priemonėse analizė

Šiame poskyryje analizuojamos belaidės komunikacijos technologijos, kurios gali būti naudojamos transporto priemonių lokaliame tinkle. Analizuojamas technologijų veikimas, topologijos, trūkumai, galimos grėsmės. Šiame poskyryje analizuojamos: „ZigBee“, „Bluetooth“, „DSRC“, „GSM“, „WiMAX“ technologijos.

### 1.5.1. „ZigBee“ technologijos analizė

„ZigBee“ technologijos moduliai naudoja „IEEE 802.15.4“ transliacijos standartą. „IEEE 802.15.4“ – „LR-PWAN“ (Low Rate Personal Wide Area Network) gali dirbti dvejomis topologijomis – žvaigždės ir P2P (peer-to-peer). Šios technologijos tinklas sudaromas iš koordinuojančio įrenginio, maršrutizatoriaus įrenginio ir galinių (klientų) įrenginių. Koordinatoriaus įrenginys gali būti tik vienas. Šis įrenginys prie tinklo pridedamas pirmasis, kuris parenka unikalius ID numerius ir leidžia kitiems įrenginiams prisijungti prie tinklo. Taip pat šis įrenginys gali suteikti saugumo servisus. Maršrutizatoriaus įrenginys skirtas siųsti žinutes „ZigBee“ tinkle. Šis įrenginys suteikia galimybę prisijungti kitiems įrenginiams, kaip pratesimo įrenginiams, taip padidinant tinklo pasiekiamumą ir funkcionalumą. Galinio mazgo (kliento) įrenginys, yra įrenginys, kuris daugiausiai laiko praleidžia miego rėžime ir duomenis siunčia tada, kada gauna užklausą arba nurodytu laiko momentu [9].

Žvaigždės topologija „ZigBee“ tinkle sudaryta iš kelių įrenginių sujungtų tik su jam priskirtu maršrutizatoriaus įrenginių. Sujungiant kelis maršrutizatorių įrenginius galima sudaryti „medžio“ formos topologiją. Tai gali padaryti tinklą mažiau patikimą, kadangi iš vienos medžio šakos esančio mazgo žinutė turi pereiti per kelis skirtingus maršrutizatorius tol, kol pasiekia galutinį tikslą. Tokioje topologijoje, vienam maršrutizatoriui sugedus, visi jo mazgai bus nepasiekiami [9]. Tokia topologija

nėra patraukli naudoti lokaliame transporto priemonės tinkle, kadangi sugedus maršrutizatoriaus įrenginiui, visi jo mazgai, šiuo atveju tai gali būti jutikliai, bus nepasiekiami.

„Peer-to-peer“ topologijoje kiekvienas tinklo įrenginys gali komunikuoti su visais kitais įrenginiais, kurie yra radijo komunikacijos diapazone [9]. Ši topologija suteikia lankstesnį tinklą, kadangi sugedus vienam įrenginiui visi kiti įrenginiai vis tiek gali pasiekti visą likusi tinklą ir jo įrenginius. Tokioje topologijoje, galima lengviau atrasti sugedusius įrenginius, o tikimybė prarasti didžiąją dalį tinklo įrenginių sugedus vienam, yra itin maža. Tokia topologija yra labiau patraukli lokaliam transporto priemonės tinklui įgyvendinti.

Technologijos saugumas yra paremtas keturiomis sąvokomis: saugumo lygis, pasitikėjimo centras, autentifikacija, duomenų šifravimas, vientisumas ir duomenų naujumas. „ZigBee“ tinkle yra du saugumo lygiai: aukšto saugumo ir standartinio saugumo, abu lygiai skiriasi raktų apsikeitimo metodika. Pasitikėjimo centras yra įrenginys, tinkle atsakingas už saugumo valdymą. Šis įrenginys parūpina tris skirtingus raktus: tinklo raktą, pagrindinį raktą, nuorodos raktą. Tinklo raktas naudojamas visų įrenginių, nuorodos raktas naudojamas dviejų komunikuojančių tarpusavyje įrenginių. Nuorodos raktas sukuriamas iš pagrindinio rakto. Duomenys šifruojamas AES 128 bitų šifravimo algoritmu naudojant CCM (Counter with Cipher Block Chaining Message Authentication Code), suteikiant autentifikacijos ir šifravimo galimybę. Duomenų vientisumas užtikrinamas naudojant MIC (Message Integrity Code) eilutę, kuri suskaičiuojama duomenų šifravimo metu. Gavus paketą kitam įrenginiui, MIC reikšmė suskaičiuojama iš naujo ir yra palyginama su gauta. Jeigu reikšmės sutampa, paketo vientisumas išlaikytas [10].

„ZigBee“ tinklui galima pakenkti skirtingomis atakomis. Tinklo raktas yra siunčiamas neužšifruotas, todėl visi įrenginiai tinkle ši raktą gauną, kas suteikia galimybę naudotis tinklu pilnavertiškai, ši problema išsprendžiama tiktais ranka įrašius raktą į įrenginio atmintį. Taip pat, tinklui galima atlikti pasiklausymo, paketų iššifravimo, duomenų manipuliacijos ir injekcijos ataktas. Be to, galima atlikti DOS atakas arba tinklo mazgo sabotažo ataką, kurios metu įrenginiui siunčiami paketai, kuriuos gavęs įrenginys negali „užmigti“ ir taupyti energijos. Jeigu įrenginys dirba nuo baterijos, tokios atakos metu, baterija išeikvojama ir įrenginys nustoja dirbti. Šias problemas galima išspręsti sukonfigūruojant įrenginį pačiam „atsikelti“ ir išsiųsti duomenis ar užsiklausti kitų įrenginių, ar yra naujos informacijos, taip dirbant nustatytu periodu taupant energijos šaltinį [10].

### 1.5.2. „Bluetooth“ technologijos analizė

„Bluetooth“ technologija dirba 2.4 GHz bangų dažnio diapazone. Sistema naudoja siuntimo-gavimo dažnius, kurie yra keičiami naudojant dažnio keitimo algoritmą tam, kad butų pašalintas komunikacijos metu esantis triukšmas komunikavimo metu naudojamo dažnio diapazone. Vidutiniškai „Bluetooth“ ryšiu galima perduodi duomenis 1 Mbps sparta. „Bluetooth“ tinkle gali būti sudaroma „piconet“ topologija, kurioje įrenginiai gali būti sujungti panašiai kaip žvaigždės topologijoje. Tokioje topologijoje įrenginiai gali būti klientais arba serveriais. Serverio įrenginys yra pagrindinis įrenginys, kuris valdo dažnių keitimo veikimą ir kuris komunikuoja su visais prie jo prijungtais klientų įrenginiais. Klientų įrenginiai negali komunikuoti tarpusavyje, jie gali komunikuoti tik su serverio įrenginiu. Taip pat, įrenginiai sinchronizuoja savo veikimą su serverio įrenginiu. Tokie „piconet“ topologijos tinklai gali būti keli ir jie gali būti sujungti vienas su kitu arba šių skirtingų tinklų serverių įrenginiai gali komunikuotu su tuo pačiu įrenginiu [9]. Tokia topologija vadinama „scatternet“. Antroji topologija, kurioje skirtingi „piconet“ tinklai yra sujungti, yra

patrauklesnė transporto priemonių lokalaus tinklo įgyvendinimui, kadangi galima įgyvendinti patvaresnį tinklą, kuriame vienam tinklo įrenginiui sugedus, likęs tinklas ir jo įrenginiai butų pasiekiami kitu maršrutu.

„Bluetooth“ tinklui galima pakenkti skirtingomis atakomis. Stebėjimo atakomis galima skenuoti tinklą ir nuskaityti tinkle esančius adresus ir gaminius, suteikiant informacijos tolimesnėms atakoms. Tinklo praplėtimo atakomis įprastas „Bluetooth“ tinklas praplečiamas naudojant išorines antenas ir įrenginius, taip suteikiant galimybę įprastai už tinklo esantiems įrenginiams prisijungti prie tinklo ar įvykdyti ataką. Obfuskacijos atakos metu tinklo įrenginiai gali būti klonuojami, jų pasiklausoma. Į tinklą gali būti įterpiami paketai su neteisingais duomenimis. Taip pat „Bluetooth“ tinklui gali būti atliekama DOS ataka ar netgi „Man in the middle“ ataka [11].

### 1.5.3. „DSRC“ technologijos analizė

„DSRC“ technologija dirba 5.9 GHz bangų dažnio diapazone. Su šia technologija įrenginiai gali komunikuoti iki 1000 metrų atstumu. Visas tinklas sudarytas iš atskirų automatizuotų tinklo mazgų. Toks tinklas gali būti paskirstomas tik į dvi kategorijas: statinis tinklas ir mobilus tinklas. Statiniame tinkle, tinklo mazgo pozicija negali būti keičiama, kai šis mazgas yra pridedamas prie tinklo. Mobiliame tinkle mazgų pozicija gali būti keičiama, tokioje topologijoje mazgai dinamiškai sudaro tinklą keistis informacija. Mobiliame tinkle maršrutizavimo užduotis yra itin sudėtinga. Reikia atkreipti dėmesį į tai, kad „DSRC“ technologija labiau tinka ne vidinio tinklo komunikacijai sudaryti, o transporto priemonės su išorine infrastruktūra tinklui sudaryti. „DSRC“ tinkle duomenys gali būti perduodami trimis būdais. Pirmasis „Peer-to-Peer“ – du tinklo mazgai esantys vienas kito diapazone komunikuoja be tarpinių mazgų. Antrasis „Remote-to-Remote“ – du tinklo mazgai esantys už vienas kito diapazono ribų komunikuoja pasitelkiant tarp jų esančius tarpinius mazgus. Trečiasis „Dynamic Traffic“ – tinklo mazgams keičiant savo pozicija tinkle, mazgų maršrutai turi būti perskaičiuojami, ko pasėkoje gali atsirasti komunikacijos trikdžių [9]. Ši technologija nėra tinkama vidinio transporto priemonės tinklo sudarymui.

### 1.5.4. GSM technologijos analizė

GSM yra standartas sukurtas aprašyti protokolus skirtus antros kartos mobilaus (2G) tinklo, naudojamo mobiliuosiuose įrenginiuose. GSM technologija gali dirbti keturiuose bangų diapazonuose: 450 MHz, 900 MHz, 1800 MHz, 1900 MHz priklausomai nuo šalių reguliuojamų įstatymų. GSM tinklas sudarytas iš stacionarios infrastruktūros ir prie jos prisijungiančių mobilių įrenginių. Mobilūs įrenginiai yra skirti naudotojams pasiekti tinklą. Tinklui yra skiriamos antenos, mobiliems įrenginiams prisijungti prie tinklo. Vienas mobilus įrenginys gali būti prisijungęs tik prie vienos antenos vienu metu, o prisijungimas prie kitos antenos yra atliekamas automatiškai. GSM tinklas gali palaikyti didelį kiekį duomenų, o tinklo plėtimas nesudaro daug tinklo trikdžių [9]. Tokia technologija tinka transporto priemonės tinklo komunikacijai su išoriniu tinklu, tačiau netinka vidiniam tinklui, kadangi visa komunikacija turi būti atliekama per tinklą pasiekiamą stacionarios antenos, todėl dingus ryšiui, pavyzdžiui aklosiose zonose, transporto priemonės vidinis tinklas sutriktų.

### 1.5.5. „WiMAX“ technologijos analizė

Didėjant plačiajuosčio belaidžio ryšio (Broadband Wireless Access) populiarumui buvo sukurtas „IEEE 802.16“ standartas dar žinomas kaip „WiMAX“. Nors ši technologija mažiau naudojama nei

„IEEE 802.11“, „WiMAX“ yra labiau tinkanti naudoti lauke. Naujausia technologijos versija „IEEE 802.16m-2011“ suteikia 100 Mbps tinklo spartą mobiliems tinklo mazgams ir iki 1Gbps stacionariems tinklo mazgams. Kadangi komunikacijai naudojami 2-11 GHz bangų dažniai, tarp tinklo mazgų gali būti kliūčių, kurios tinklo veikimui netrukdys [12].

„WiMAX“ kaip ir kitos belaidės technologijos yra labiau pažeidžiamos negu laidinės, tokios atakos kaip „man-in-the-middle“, DOS ir pakartojimo atakos gali būti atliekamos šiam tinklui pažeisti. Taip pat problema kyla mobilių įrenginių palaikyme, kadangi tinklas gali būti atakuojamas iš naujai prie tinklo pridėto įrenginio ir iš bet kurios tinklo dalies. Tačiau, reikia paminėti tai, kad šioje technologijoje galima naudoti šifruojamas kontrolės žinutes ir naudoti „PKMv3“ (Privacy and Key Management) protokolą, kuris naudoja „X.509“ sertifikatus ir dviejų raktų šifravimą komunikavimo raktams apsikeisti. PKM gali būti padalinta į tris dalis: autorizacija, raktų išvedimas, rankos paspaudimo (handshake) įvykdymas. Autorizacija atliekama pirmoji. Pradžioje išsiunčiama autentifikacijos informacijos žinutė, kurioje pateiktas gamintojo sertifikatas. Po šios žinutės išsiunčiama autorizacijos užklausa, kurioje pateiktas gamintojo sertifikatas, įrenginio palaikomi kriptografiniai algoritmai ir atsitiktiniai sugeneruotas skaičius. Šios žinutės tikslas yra gauti autorizacijos raktą. Atsakyme pateikiamas sertifikatas patvirtinti žinutės autentiškumą, užšifruotas autorizacijos raktas, rakto gyvavimo laikas, saugumo identifikatorius, tas pats atsitiktinis raktas nurodyti kuriai užklausai gautas atsakymas, naujas atsitiktinis skaičius ir „RSA“ parašas visai žinutei. Gavus atsakymo žinutę išsiunčiamas autorizacijos patvirtinimas. Antras žingsnis – raktų išvedimas. Raktai yra sukuriami naudojant „Dot16KDF“ funkciją. Trečias žingsnis rankos paspaudimas – atliekamas užtikrinti, kad abiejų įrenginių naudojamas autorizacijos raktas sutampa. Atlikus šiuos veiksmus ir patvirtinus, kad įrenginys ir pagrindinė stotis prie kurios įrenginys prisijungęs sėkmingai autorizavosi, pagrindinė stotis sukuria kriptografinį raktą, kuriuo abu įrenginiai šifruos žinutes vienas kitam. [12].

## 1.6. Įrenginių autentifikavimo ir autorizavimo metodų analizė

Ribotų išteklių sistemose stiprūs, šiandien kompiuteriuose naudojami, autentifikavimo ir autorizavimo metodai gali reikalauti per daug resursų, taip lėtinant sistemos veikimą. Todėl, tokioms sistemoms kuriami išskirtiniai metodai, kurie išspręstų resursų naudojimo ir saugumo efektyvumo problemą. Šiame poskyryje analizuojami šiandien naudojami autentifikacijos ir autorizacijos algoritmai „Bluetooth“ ir LTE tinkluose, bei analizuojami mažai resursų reikalaujantys metodai skirti autentifikavimui, autorizavimui ir privatumui užtikrinti.

### 1.6.1. „Bluetooth“ autentifikacijos metodas ir autorizacijos lygiai

Autorizacija ir autentifikacija „Bluetooth“ technologijoje atliekama raktų apsikeitimu ir jų generavimu. Komunikacijos pradžioje, komunikaciją inicializuojantis įrenginys privalo sugeneruoti inicializavimo raktą. Šis raktas susideda iš atsitiktinio skaičiaus, kuris nusiunčiamas kitam įrenginiui, jeigu nustatyta, susideda ir iš PIN kodo (jeigu nenustatyta, naudojamas „0000“ PIN kodas). Matyti, kad jeigu įrenginyje nėra nurodyto PIN kodo, inicializavimo raktas yra silpnas, kadangi atsitiktinis raktas yra siunčiamas kitam įrenginiui neužšifruotas. Abudu komunikuojantys įrenginiai turi sugeneruoti vienodą inicializavimo raktą. Tai atlikus, įvykdomas abipusis autentifikavimo metodas. Įrenginys prie kurio bandoma prisijungti – įrenginys „B“, atsiunčia savo „Bluetooth“ adresą, tuomet šį adresą gavęs įrenginys – įrenginys „A“, sugeneruoja atsitiktinį skaičių – iššūkį (challange) ir nusiunčia jį įrenginiui iš kurio gavo „Bluetooth“ adresą – įrenginiui „B“. Abu įrenginiai, „A“ ir „B“,

naudodami tą patį vieno įrenginio „B“ išsiųstą „Bluetooth“ adresą, atsitiktinį skaičių ir inicializavimo raktą, suskaičiuoja iššūkio atsakymą. Tuomet įrenginys „B“ nusiunčia atsakymą įrenginiui „A“, abu atsakymai turi sutapti. Tai pakartojama dar kartą įrenginiui „A“ nusiuntus savo „Bluetooth“ adresą įrenginiui „B“. Kai abu įrenginiai sukuria lygius atsakymus, abipusė autentifikacija yra sėkminga. Toliau atliekamas raktų generavimas sudaryti ir šifruoti komunikacijos tuneliui [13].

Autorizacijos proceso metu, „Bluetooth“ įrenginys nustato ar kitas „Bluetooth“ įrenginys gali gauti prieigą prie specifinių funkcijų. Ši autorizacija yra paremta dviem konceptais: pasitikėjimo santykiais ir servisų saugumo lygiais. Be to, autorizacija yra priklausoma nuo autentifikacijos, kadangi autentifikacijos metu nustatoma įrenginio tapatybė, kuri naudojama nustatyti prieigos lygmenis. „Bluetooth“ technologijoje yra trys pasitikėjimo lygmenys [14]:

1. Patikimas – įrenginys autentifikavosi ir turi prieigą prie prisijungto įrenginio funkcijų.
2. Nepatikimas – įrenginys autentifikavosi, bet prieiga prie funkcijų ribota.
3. Nežinomas – įrenginys nebuvo autentifikuotas ir laikomas nepatikimu.

### 1.6.2. Mobilaus tinklo autentifikacija

LTE mobiliuosiuose tinkluose autentifikacijos mechanizmui naudojamas EPS-AKA protokolas, kuriame EAP (Extensible Authentication Protocol) yra autentifikacijos karkasas. Autentifikacijos procesas prasideda iškarto po ryšio sudarymo tarp naudotojo įrenginio ir mobiliojo tinklo valdytojo. Valdytojas išsiunčia identifikacijos užklausą naudotojo įrenginiui, kuris turi atsakyti pateikdamas savo IMSI (International Mobile Subscriber Identity) reikšmę. Ši reikšmė yra išskirtinis skaičius, kuris identifikuoja kiekvieną įrenginį esantį mobiliame tinkle. Mobiliojo tinklo valdytojui gavus IMSI reikšmę, išsiunčiama nauja užklausa gauti EPS autentifikavimo vektoriui iš HSS (Home Subscriber Service) abonementų paslaugų valdymo duomenų bazės, naudojant naudotojo įrenginio pateiktą IMSI reikšmę. HSS sukuria atsitiktinį iššūkio (challange) skaičių ir sugeneruoja autentifikavimo vektorių, kuriame saugoma laukiama naudotojo įrenginio atsakymo į iššūkį reikšmė. Šis vektorius nusiunčiamas tinklo valdytojui, kuris naudotojo įrenginiui išsiunčia atsitiktinį skaičių ir kitus autentifikavimui reikalingus parametrus, tačiau nepateikia atsakymo į iššūkį reikšmės. Naudotojo įrenginys tuomet suskaičiuoja savo autentifikacijos parametrus, kuriuos sulygina su gautais iš tinklo valdytojo. Jeigu parametrai sutampa, įrenginys suskaičiuoja atsakymo reikšmę, kurią nusiunčia tinklo valdytojui. Valdytojas patikrina gautą atsakymo reikšmę su saugoma iššūkio atsakymo reikšme, jeigu jos sutampa, abipusis autentifikavimas yra baigtas. Po šių veiksmų valdytojas įrenginiui nusiunčia autentifikacijos sėkmės arba nesėkmės žinutes [15].

### 1.6.3. „LEAP“ lengvas šifravimo ir autentifikavimo protokolas

LEAP (Lightweight Encryption and Authentication Protocol) susideda iš dviejų mechanizmų: raktų valdymo mechanizmo ir žinučių šifravimo ir autentifikavimo mechanizmo. Raktų valdymo mechanizmas atsakingas už sesijos raktų generavimą ir jų paskirstymą komunikuojantiems įrenginiams. Žinučių šifravimo ir autentifikavimo mechanizmas naudoja sugeneruotus raktus užtikrinti žinučių konfidencialumą ir autentiškumą [16].

Raktų generavimo mechanizmas yra aukščiausio prioriteto tam, kad užtikrinti saugumo reikalavimus. Ilgaamžiai simetriniai raktai naudojami skirtingų įrenginių yra sugeneruojami pagaminimo metu arba kai įrenginys yra pakeičiamas sistemoje. Sesijos raktai yra išvedami iš

įrenginio simetrinio rakto periodiškai ir yra naudojami žinučių šifravime ir autentifikavime. Kiekvienai komunikacijai tarp skirtingų įrenginių sudaromi skirtingi sesijų raktai – kiekvienas komunikacijos tunelis turi savo sesijos raktus. Šių sesijų raktų saugumas patobulinamas naudojant du metodus. Pirmasis metodas – sesijos raktai yra žinomi tik įrenginių kurie komunikuoja tarpusavyje. Taip sukompromituotas įrenginys sistemoje negali apgauti kitų įrenginių, kurie su juo nekomunikuoja. Antras metodas – sesijos raktai yra atnaujinami periodiškai tam, kad pasipriešinti brutalios jėgos (brute-force) atakai [16].

Visi ilgalaikiai simetriniai raktai yra saugojami saugiame įrenginyje – raktų valdytojas, kuris turi pakankamai resursų jiems saugoti ir aukštą saugumo lygį. Taip pat šis įrenginys dirba kaip centralizuotas valdytojas, užtikrinantis raktų valdymo mechanizmo saugumą. LEAP protokole naudojamas AES (Advanced Encryption Standard) kaip simetrinis kriptografijos algoritmas. SHA (Secure Hash Algorithm) naudojamas kaip raktų santraukų funkcija. Sesijų raktai yra atnaujinami ir paskirstomi šešiais žingsniais [16]:

1. Saugus įrenginys (raktų valdytojas) saugantis simetrinius raktus sugeneruoja atsitiktinį skaičių iš kurio sugeneruoja simetrinį sesijos raktą naudojant ilgaamžį simetrinį raktą.

2. Raktų valdytojas užšifruoja sesijos raktus su ilgaamžiu simetriniu raktu tam, kad sukurtų šifrą.

3. Raktų valdytojas sugeneruoja MAC reikšmę, kuri yra įrenginio identifikacinio numerio ir šifro sąjunga, naudojant ilgaamžį simetrinį raktą. Tuomet raktų valdytojas išsiunčia raktų atnaujinimo užklausą komunikaciją sudarantiems įrenginiams. Šioje užklausoje pateikiamas identifikacinis numeris, šifras ir MAC reikšmė.

4. Įrenginiai turi patvirtinti gautą MAC reikšmę tam, kad autentifikuoti raktų atnaujinimo užklausą. Užklausą autentifikavus, įrenginys iššifruoja gautą šifrą naudojant savo ilgaamžį simetrinį raktą. Taip atkuriant sesijos raktą.

5. Įrenginys sugeneruoja naują MAC reikšmė naudojant identifikacinį numerį ir sesijos raktą ir išsiunčia rakto atnaujinimo atsakymą atautentifikavimui raktų valdytojui.

6. Raktų valdytojas patikrina gautas MAC reikšmes iš abiejų komunikaciją sudarančių įrenginių raktų atnaujinimo atsakymo autentifikavimui.

Žinučių šifravimo ir autentifikavimo mechanizmas naudoja vieną iš žinomiausių RC4 (Rivest Cipher) srauto šifro algoritmą, kuris susideda iš raktų planavimo KSA (Key Scheduling Algorithm) ir pseudoatsitiktinio generavimo PRGA (Pseudo-Random Generation Algorithm) algoritmų. KSA iš rakto sugeneruoja pirminę *m* baitų ilgio kombinaciją. Įprastai kombinacija būna tarp 5 ir 64 baitų ilgio. Pagrindinė RC4 dalis yra PRGA, kuri sukuria vieno baito išvestį kiekviename žingsnyje. Teksto šifravimas atliekamas XOR binarinės operacijos metu sujungiant pseudoatsitiktinę skaičių eilę sujungus su tekstu. Atlikus raktų atnaujinimo ir paskirstymo procesą, komunikuojantys įrenginiai turi simetrinius sesijų raktus, kuriuos panaudojus galima užšifruoti ir autentifikuoti žinutes naudojant RC4 algoritmą vietoje AES algoritmo, ir MAC reikšmėmis paremtais būdais. Tai atliekama keturiais žingsniais [16]:

1. Įrenginys įvykdo RC4 algoritmą tam, kad sugeneruoti srauto raktą naudojant sesijos raktą.

2. Įrenginys sujungia identifikacinį numerį su raktų srautu naudojant XOR binarinę operaciją sukuriant naują identifikacinį numerį.

3. Įrenginys nusprendžia į kuria duomenų lauko vietą įterpti identifikacinį numerį. Kadangi identifikacinis numeris 11 bitų ilgio, o žinutė yra 64 bitų ilgio, galimos 54 pozicijos.

4. Įrenginys naujai gautą duomenų srautą sujungia su raktų srautu naudojant XOR binarinę operaciją. Rezultatų duomenys įdedami į žinutę, kuri yra išsiunčiama.

Kadangi kiekvienai žinutei sugeneruojamas naujas srauto raktas, žinutės siuntėjo įrenginys ir gavėjo įrenginys bendrai valdo žinučių skaitiklį tam, kad komunikacija butų sinchronizuota. Abu įrenginiai saugo šio skaitiklio reikšmę, kuri yra atstatoma raktų atnaujinimo ir paskirstymo metu. Šis skaitiklis padidinamas vienu tada, kai siuntėjas išsiunčia žinutę arba gavėjas sėkmingai gauna žinutę. Prieš gaunant žinutę, gavėjas sugeneruoja srauto raktą naudojant sesijos raktą ir skaitiklio reikšmę, tuomet žinutė yra iššifruojama naudojant sugeneruota srauto raktą. Atlikus iššifravimą patikrinimas identifikacinis numeris pateiktas žinutėje su gautos žinutės identifikaciniu numeriu, jeigu jis sutampa žinutė autentifikuota sėkmingai. Jeigu identifikaciniai numeriai nesutampa žinutė pašalinama ir skaitiklis nepadidinamas [16].

### 1.6.4. Lengvas autentifikacijos protokolas siūlomas nešiojamiems įrenginiams

Šis protokolas suprojektuotas leisti abipuse autentifikaciją tarp įrenginių ir pagrindinės stoties. Įrenginiams sėkmingai įvykdžius abipusę autentifikaciją, įrenginiai sugeneruoja sesijos raktus, kuriais apsaugo komunikacijos tunelį. Šis metodas yra lengvas todėl, kad naudoja lengvas kriptografines operacijas – negrįžtamos santraukos funkcija ir XOR binarinė operacija. Autentifikacija atliekamas keturiais žingsniais [17]:

1. Pasirengimo diegti etapas – šio etapo metu, įrenginio atmintyje yra išsaugomi autentifikavimo parametrai. Įrenginiui yra parenkamas unikalus identifikacijos numeris ir sugeneruojamas atsitiktinis skaičius. Šios dvi reikšmės yra sujungiamos santraukos funkcija, kurios rezultatas yra pseudo-tapatybes identifikatorius. Taip pat sugeneruojama laikina atsitiktinė reikšmė laikinam tapatybės identifikavimui. Toliau sugeneruojamas atsitiktinis skaičius, kuris naudojamas kaip slaptas raktas su laikina tapatybės reikšme ir laiko žyma, taip sukuriant laikiną kredencialą. Kredencialo, laikinos tapatybės ir pseudo-tapatybės reikšmės išsaugomos įrenginio atmintyje.

2. Registracijos etapas – šio etapo metu, pagrindinės stoties įrenginio atmintyje išsaugomi autentifikavimui ir sesijos raktų generavimui reikalingi parametrai. Tai gali būti slaptažodis, kriptografinis raktas ar biometriniai duomenys iš kurių yra sugeneruojami tapatybės duomenys kiekvienam prie soties prijungtam įrenginiui.

3. Prisijungimo etapas – šio etapo metu, įrenginys pateikia autentifikavimo parametrus pagrindinei stočiai, kur turi juos patvirtinti. Sėkmingai patvirtinus duomenis, stotis išsiunčia prisijungimo žinutę.

4. Autentifikacijos ir raktų apsikeitimo etapas – kai prisijungimo etapas sėkmingai įvykdomas, įrenginys patikrina gautą prisijungimo žinutę ir ją autentifikuoja. Jei žinutė patvirtinta sėkmingai, įrenginys nusiunčia žinutę stočiai, kuri turi ją autentifikuoti. Abiem įrenginiams autentifikavus žinutes, abu įrenginiai gali sugeneruoti sesijos raktus.

### 1.6.5. DTLS – Datagram Transport Layer security protokolas

DTLS protokolas suteikia saugumą datagramomis paremtoms programoms, suteikiant galimybe komunikuoti apsaugant nuo pasiklausymo, klastojimo ar žinučių modifikavimo. DTLS paremtas srautu orientuoto TLS protokolu ir yra skirtas suteikti panašias saugumo garantijas. Šis protokolas palaiko transporto greitį nesukeliant vėlavimo, tačiau kadangi naudojamas UDP protokolas komunikacijai, programos turi rūpintis paketų pertvarkymu, praradimu ir duomenimis, kurie yra per dideli vienam paketui.

## 1.7. Įrenginių duomenų apsaugojimo metodų analizė

Įrenginių siunčiami duomenys apsaugomi naudojant skirtingus kriptografinius algoritmus. Perduodant duomenis tinkle, duomenys yra šifruojami. Šifravimui skirti algoritmai gali būti padalinti į tris grupes: simetriniai algoritmai, asimetriniai algoritmai ir kriptografiniai protokolai. Pagal tai, kiek raktų algoritmas naudoja duomenų šifravimui ir iššifravimui, algoritmai grupuojami į dvi klasifikacines grupes. Pirmoji grupė – simetriniai algoritmai, naudoja tik vieną raktą šifravimui ir iššifravimui. Antroji grupė – asimetriniai algoritmai, naudoja du raktus, iš kurių vienas yra viešai žinomas ir vienas slaptas. Šiuose algoritmuose viešas raktas naudojamas šifravimui, o privatus iššifravimui [18].

Ribotų išteklių sistemose, sudėtingi šifravimo metodai ir sudėtingi duomenų siuntimo metodai gali reikalauti per daug išteklių, taip lėtinant sistemos veikimą. Todėl, tokiomis sistemos reikalingi išskirtiniai metodai, leidžiantys apsaugoti duomenų konfidencialumą ir vientisumą išlaikant saugumo ir resursų efektyvumą. Šiame poskyryje analizuojami skirtingi mažai resursų reikalaujantys šifravimo algoritmai, santraukų algoritmai ir saugūs duomenų perdavimo metodai.

### 1.7.1. „DESL“ ir „DESXL“ lengvų šifravimo algoritmų analizė

DESL yra resursų atžvilgiu lengvesnė klasikinio DES algoritmo versija, o DESXL yra resursų atžvilgiu lengvesnė DESX algoritmo versija, kurioje naudojamas tik vienas pakeitimo blokas vietoje aštuonių. Kadangi naudojamas tik vienas pakeitimo blokas, įrenginyje sutaupoma atminties ir šis blokas suteikia pasipriešinimą prieš įprastas kriptoanalizės atakas [18].

Pagrindinės DESL ir DESXL šifravimo algoritmo idėjos yra tos pačios kaip DES arba išvestos iš jų:

1. Suteikia galimybę naudoti įrenginių serijinę architektūrą, kas sumažina kompleksiškumą.
2. Papildomai suteikia galimybę pritaikyti „raktų balinimą“ tam, kad pasunkinti arba visai panaikinti brutalios jėgos atakas.
3. Papildomai pakeičia aštuonis pakeitimo blokus į vieną taip sumažinant kompleksiškumą.

Pritaikius pirmąją idėją, gaunama lengvesnė DES implementacija, kuri suvartoja 35 procentais mažiau resursų nei geriausios AES implementacijos. Taip pat šios idėjos įgyvendinimas sumažina algoritmo naudojamus skaičiavimo ciklus vienam blokui 86 procentais palyginus su serijiniu AES algoritmu. Tačiau, naudojami raktai yra trumpi – 56 bitų ilgio, todėl pritaikyti brutalios jėgos ataką nesunku, o visas galiams rakto kombinacijas galima suskaičiuoti per kelias dienas, todėl šis sprendimas tinka tik trumpo saugumo reikalaujančioms sistemoms. Tam, kad padidinti saugumą galima pritaikyti antrąją idėją. Rakto balinimas atliekamas sujungus DES raktą naudojant XOR

binarinę operaciją su nauju raktu ar bitų seka, taip pakeičiant rakto ilgį ir atakos metu reikalingų skaičiavimų kiekį. Jeigu reikia itin lengvo algoritmo, pritaikoma trečioji idėja, kurios metu naudojamas tik vienas pakeitimo blokas. Tai sumažina viso algoritmo kompleksiškumą [19].

DESL algoritmas yra saugus prieš tam tikrus tiesines ir diferencialines kriptoanalizės metodus ir „Davies-Murphy“ ataką, kadangi algoritmas naudoja tik vieną pakeitimo bloką kartojant aštuonis kartus. Todėl yra sumažinama kolizijų galimybė šių blokų rezultate ir visos funkcijos rezultate [18].

### 1.7.2. „Katan“ ir „Ktantan“ blokinių šifravimo algoritmų analizė

„Katan“ ir „Ktantan“ yra šešių blokų šifrai orientuoti į aparatinę įrangą ir yra padalinti į tris „Katan“ šifrus: „Katan32“, „Katan48“, „Katan64“ ir tris „Ktantan“ šifrus: „Ktantan32“, „Ktantan48“, „Ktantan64“. Skaičiai prie algoritmo pavadinimo nurodo blokų dydį bitais. Visi algoritmai naudoja 80 bitų ilgio raktą. Skirtumas tarp šių algoritmų yra tai, kad „Ktantan“ yra kompaktiškesnis aparatinės įrangos atžvilgiu. Šiame algoritme raktas yra įrašomas įrenginio atmintyje ir negali būti pakeičiamas. Todėl „Ktantan“ šifrai yra maži blokiniai šifrai palyginus su „Katan“ ir yra naudojami įrenginiuose, kurie yra inicializuojami tik su vienu raktu. Algoritmai naudoja mažai resursų dėl šių trijų priežasčių [18]:

1. Pradinės būsenos dydis yra lygus viso algoritmo bloko dydžiui. Algoritmas naudoja postūmio registrus ir grįžtamojo ryšio (feedback) funkcijas, kurias yra nesunku implementuoti aparatinėje įrangoje.

2. Algoritmai apdoroja mažus blokus duomenų, kurių dydis yra tarp 32 ir 64 bitų ilgio.

3. „Ktantan“ rakto planas yra lengvas, kadangi raktas nekinta palyginus su „Katan“, kuriame raktas yra transformuojamas.

„Katan“ ir „Ktantan“ algoritmai yra saugūs prieš tiesinius ir diferencialinius kriptoanalizės metodus, tačiau yra pažeidžiami slydimo atakų (slide attack). Šios atakos metu randamos dvi žinutės, kurių šifravimo procesai yra itin panašūs. Šią ataką galima pritaikyti jeigu algoritme parinktas mažas šifravimo ciklų skaičius. Taip pat „Ktantan“ šeimos algoritmai yra pažeidžiami kubo atakų [3].

### 1.7.3. „Present“ lengvo šifravimo algoritmo analizė

„Present“ yra vienas iš lengviausių algoritmų, kuris gavo ISO/IEC standartą už lengviausią kriptografiją. Šis algoritmas yra paremtas transformacijos lygmenimis iš „Serpent“ ir DES algoritmų. Transformacijos lygmenys buvo detaliai išanalizuoti, ypač atkreipiant dėmesį į saugumą ir aparatinės įrangos panaudojimo efektyvumą. Šiam algoritmui reikia itin mažai skaičiavimo ciklų, įvykdomas 31 ciklas su 64 bitų ilgio duomenų blokais naudojant XOR binarinę operaciją. Algoritmas leidžia naudoti 80 arba 128 bitų raktus. „Present“ algoritmas buvo sukurtas aparatinės įrangos įgyvendinimui, tačiau gali būti įgyvendintas ir programiškai, tačiau naudojamas šifruoti tik mažam arba vidutiniškam duomenų kiekiui [18].

„Present“ algoritmas yra labiau pažeidžiamas tiesinių atakų negu diferencialinių atakų. Taip yra dėl bitinės permutacijos (matematinė funkcija nustatanti galimų simbolių sandaros kiekį, kai simbolių išsidėstymas yra svarbus) naudojamos algoritme. Taip pat šis algoritmas yra pažeidžiamas slydimo atakų ir susijusio rakto atakos, kurios metu naudojami skirtingi raktai šifruoti vienai tekstogramai stebint šifravimo rezultatus [18].

### 1.7.4. „Hummingbird“ hibridinio šifravimo algoritmo analizė

„Hummingbird“ yra hibridinis algoritmas sudarytas iš blokinio šifro ir srautinio šifro. Šis šifras duomenis šifruoja 16 bitų ilgio blokais ir naudoja 256 bitų ilgio raktą. Taip pat, turi 80 bitų vidinę būseną ir paprastas logines ir aritmetines operacijas. Todėl, kad naudoja mažo ilgio blokus šifravimo metu, šis algoritmas turi minimalų atsakymo laiką ir energijos suvartojimo reikalavimus. Dėl šios priežasties tinka belaidžių jutiklių komunikacijai be papildomų modifikacijų. Nors ir operacijos atliekamos su trumpais 16 bitų blokais, palyginus su „Present“ algoritmu, „Hummingbird“ turi didesnį vėlavimo ir vykdymo laiką. Todėl šiame algoritme šifravimas yra lėtesnis, todėl „Hummingbird“ algoritmas yra mažiau efektyvus [18]. Šis algoritmas aktyviai naudoja $2^{16}$ modulio sumos funkciją tam, kad sumaišyti įrenginio registrų būsenas, taip paruošiant registrus tolimesniam darbui ir pašalinant registruose saugomus duomenis [20].

Naujesnė protokolo versija „Hummingbird-2“ gali sukurti autentifikacijos žymą kiekvienai žinutei, tačiau šifravimas toliau atliekamas 16 bitų ilgio blokais. Naujesnėje versijoje raktas pakeistas į 128 bitų ilgio ir vidinė būsena padidinta iki 128 bitų, kuri inicializuojama naudojant 64 bitų ilgio inicializavimo vektorių. Tam, kad autentifikuoti duomenis kurie keliauja kartu su šifruotais duomenims, „Hummingbird-2“ algoritmas naudoja autentifikuoto šifravimo su asocijuotais duomenimis metodą (Authenticaded Encryption with Associated Data). Duomenų asocijavimas atliekamas tik po duomenų užšifravimo. Taip pat, verta paminėti, kad naujesnė algoritmo versija naudoja mažiau energijos ir skaičiavimai yra atliekami greičiau nei su senesne versija [18].

Kadangi algoritmas naudoja hibridinę metodiką sudaryta iš blokinio šifro ir srautinio šifro, „Hummingbird“ algoritmas yra atsparus dažniausiai pasitaikančioms atakoms prieš blokinius šifrus ir srautinius šifrus. Taip pat algoritmas atsparus gimtadienio dienos (birthday) atakai, diferencialiniams ir tiesiniams kriptoanalizės metodams. Tačiau šis šifravimo algoritmas gali būti pažeistas kubo atakos jeigu naudojamas mažas vidinės būsenos keitimo ciklų skaičius [18].

### 1.7.5. „TEA“ šifravimo algoritmo analizė

TEA (Tiny Encryption Algorithm) sukurtas naudoti mažai resursų turinčiuose mažuose kompiuteriuose. Šis blokinis šifras paremtas didelio našumo, bet matematiškai paprastu šifravimo algoritmu, kuris yra išvestas iš „Feistel“ šifro. TEA algoritmas duomenis šifruoja 64 bitų ilgio blokais, kurie yra padalinti į 32 bitų ilgio blokus. Raktas naudojamas šifravimui yra 128 bitų ilgio. Pats algoritmas yra ciklais pagrįstas šifravimo metodas, ciklų kiekis gali būti kintamas skaičius, tačiau patariama naudoti 32 ciklus. TEA su 32 ciklais yra greitesnis nei DES su 16 ciklų ir šis algoritmas gali būti įgyvendinamas visuose programavimo kalbose. XTEA yra patobulinta TEA versija, kurioje naudojami 64 šifravimo ciklai. Patobulinta versija turi sudėtingesnį raktų valdymą ir pakeistas matematines operacijas. Tačiau didelio skirtumo efektyvumui nėra [18].

Kadangi raktų valdymas yra gana paprastas TEA algoritmas gali būti pažeidžiamas lygaus rakto atakos, kurios metu išnaudojama silpnybė, kad vienas raktas yra lygus trims kitiems raktams. Taip pat šis algoritmas gali būti atakuojamas naudojant panašaus rakto ataką ir slydimo ataką. Šios problemos yra išspręstos patobulintoje XTEA algoritmo versijoje [18].

### 1.7.6. „Curupira“ šifravimo algoritmo analizė

„Curupira“ algoritmas yra paremtas „Wide Trail“ strategija. Ši strategija skirta sukurti algoritmą, kuris sujungia resursų efektyvų išnaudojimą ir pasipriešinimą prieš diferencialinius ir tiesinius kriptoanalizės metodus. Šis algoritmas yra sukurtas naudoti sistemose, kuriuose ištekliai yra riboti. Tam, kad šis algoritmas tiktų matematiškai lengvų algoritmų grupei, „Curupin“ turi šias savybes [18]:

1. Duomenų bloko ilgis yra 96 bitai ir šis duomenų blokas yra pateikiamas kaip 3 eilučių 4 stulpelių baitų matrica.
2. Algoritme gali būti naudojami 96, 144 ir 192 bitų ilgio raktai.
3. Skaičiavimo ciklų skaičius yra nustatomas pagal rakto ilgį.
4. Algoritme yra naudojamas 8 eilučių 8 stulpelių bitų pakeitimo blokas, kuris yra įgyvendintas kaip du atskiri 4 eilučių 4 stulpelių bitų blokai. Šis sprendimas sutaupo atminties kiekį reikalingą saugoti pakeitimo blokams.

„Curupira-2“ yra patobulinta algoritmo versija, kuri skiriasi nuo pirmosios versijos tik tuo, kad naudoja kitokį raktų valdymo metodą, kuris padidina algoritmo efektyvumą, jeigu raktai yra sukuriami pačio algoritmo, vietoje jau išsaugoto rakto naudojimo [21].

### 1.7.7. „Twine“ šifravimo algoritmo analizė

„Twine“ algorimtas yra paremtas apibendrinta Feistel struktūra (Generalized Feistel Structure - GFS), kas leidžia įgyvendinti algoritmą ir aparatinėje įrangoje ir programiškai. Tačiau algoritmą įgyvendinus aparatinėje įrangoje, reiktų duomenis šifruoti įgyvendintu algoritmu du kartus tam, kad užtikrinti duomenų saugumą. Dėl šios priežasties „Twine“ naudoja patobulinta GFS struktūrą, kuri suteikia efektyvų ir greitą algoritmo veikimą išlaikant matematinių skaičiavimų lengvumą. Šis algoritmas matematinį lengvumą išlaiko dėl šių priežasčių [18]:

1. Šifruojami duomenys yra padalinami į 64 bitų ilgio blokus.
2. Duomenys yra šifruojami naudojant 36 ciklus.
3. Algoritmas turi du tipus: „Twine-80“, kuriame raktų ilgis yra 80 bitų ir „Twine-128“, kuriame raktų ilgis yra 128 bitai.

### 1.7.8. DTLS saugumo protokolas

DTLS (Datagram Transport Layer Security) protokolas yra transporto lygmens saugumo protokolas, kuris turi įgyvendintą raktų valdymą, parametrų derybos metodus ir saugų duomenų perdavimą. Kadangi DTLS protokolas yra bendrinis, skirtas plačiam taikymui, tai šis protokolas yra prasčiau optimizuotas realiu laiku veikiančiuose sistemose. Tam, kad sužinoti ar kitas komunikavimo įrenginys vis dar veikiantis, reikia paaukoti didelį kiekį skaičiavimo galios vykdant parametrų derinimą iš naujo. Taip pat, DTLS protokolui būtina suskaičiuoti kelio MTU reikšmę (PMTU – path maximus transision unit) iki kito IP adreso, tačiau neturi įgyvendinto žinučių valdymo šiai funkcijai atlikti nepažeidžiant tinkle siunčiamų žinučių efektyvumo.

### 1.7.9. „AES-CBC“ simetrinis blokinis šifravimo algoritmas

AES-CBC (Advanced Encrytion Standard – Cipher Block Chaining) algoritmas yra saugus šifras nuo standartinių atakų. Svarbu, kad pasikartojančios teksto iškarpos vienoje tekstogramoje yra paslepiamos, kadangi kiekvienam teksto blokui naudojama XOR Bulio operacija naudojant prieš tai užšifruotą teksto bloką. Dėl šios algoritmo funkcijos panaikinama galimybė analizuoti šifrogramą ieškant pasikartojančių iškarpų. Kadangi naudojami gana maži raktai ir teksto blokai (128, 192 ar 256 bitų ilgio), šifravimas ir iššifravimas šio algoritmų yra lengvas kompiuterinio skaičiavimo galios atžvilgiu. Taip pat, šiam algoritmui, šiandieninėje mikrovaldiklių architektūroje, plačiai taikoma įrangos lygio optimizacija. Ši optimizacija padidina AES-CBC algoritmo efektyvumą. Svarbiausia tai, kad AES 128 bitų šifras yra atsparus brutalios jėgos atakai dėl didėlio kiekio galimų rakto variacijų.

## 1.8. Transporto priemonių lokalaus belaidžio tinklo saugumo analizės išvados

1. Atlikus problematikos, atakų, komunikacijos technologijų ir kriptografinių algoritmų analizę, pastebėta, kad transporto priemonės lokalus tinklas gali būti pažeidžiamas daugybės skirtingų atakų. Dauguma atakų gali būti atliekamos iš išorės. Taip pat yra atakų, kurios gali būti atliekamos dėl naudotojų neatsargumo, pridedant užkrėstą įrenginį prie transporto priemonės lokalaus tinklo. Nemažai problemų lokaliam tinklui kyla dėl kriptografijos silpnumo arba neegzistavimo, todėl išoriniams objektas yra nesunku įgyvendinti ataką prieš transporto priemonės lokalų tinklą. Atsižvelgiant į tai, kad sistema dirba ribotų išteklių aplinkoje, kriptografiniai sprendimai yra itin ribojami, kadangi sudėtingi algoritmai reikalauja daug matematinių skaičiavimų, sistemos veikimas yra lėtinamas arba visai sustabdomas.

2. Dalis analizuotų problemų iškyla ir dėl autentifikacijos trūkumo. Išoriniam objektui įsilaužus į tinklą, šis objektas gali tinklui pateikti neteisingus duomenis, kurie gali pakenkti visos sistemos veikimui. Jeigu sistemoje nėra įgyvendinto arba įgyvendintas silpnas autentifikacijos metodas, atakos metu nėra kaip atpažinti įsilaužimą.

3. Taip pat pastebėta, kad dalį problemų galima išspręsti izoliuojant transporto priemonės lokalų tinklą nuo išorinio ir izoliuojant atskiras sistemos posistemes kontroliuojant posistemių komunikaciją naudojant kintančių kriptografinių raktų metodiką. Todėl nuspręstą tolimesniuose projekto vystymo etapuose projektuoti metodą užtikrinantį sistemos tinklo izoliaciją nuo išorinių objektų ir posistemių izoliaciją tarpusavyje komunikuojant periodiškai kintančiais kriptografiniais raktais, ir autentifikaciją užtikrinančiais algoritmais. Nuspręsta įgyvendinti algoritmą paremtą kriptografinių raktų periodinių keitimu, užtikrinti saugiai komunikacijai tarp skirtingų posistemių.

4. Siūlomo sprendimo tolimesniam vystymui pasirinkta naudoti „Bluetooth“ komunikacijos technologiją. Šis sprendimas priimtas, nes „Bluetooth“ technologija yra trumpo nuotolio, palaiko daugiau nei vieną komunikacijos sąsają ir nereikia sudaryti sąsajos su globaliu interneto tinklu.

5. Siūlomo sprendimo tolimesniam vystymui pasirinkta naudoti AES-CBC šifravimo algoritmą, kadangi šiam algoritmui realizuojamos įrangos lygio optimizacijos, šifruojama trumpais raktais ir teksto blokais. Taip pat kiekvienas užšifruotas blokas prisideda prie tolimesnių blokų šifravimo, taip padidinant šifravimo saugumą ir panaikinant atakų galimybes. Šis algoritmas tinka įterptinėms sistemoms.

## 1.9. Uždaviniai tolimesniam projekto vystymui

Projekto tikslas – suprojektuoti saugų komunikacijos protokolą, kuris būtu lengvas skaičiavimui atliekamų veiksmų skaičiumi, energijos suvartojimu ir būtu pritaikomas IOT technologijose transporto priemonėse.

Tolimesniam projekto vystymui iškelti šie uždaviniai:

1. Išvystyti metodo, paremto kriptografinių raktų kitimu, viziją.
2. Suprojektuoti metodo architektūrą. Kokie objektai komunikuoja, kurie objektai atsakingi už raktų keitimąsi ir atnaujinimą.
3. Suprojektuoti metodo programinę įrangą. Kokie naudojami kriptografiniai algoritmai, komunikacijos protokolai ir technologijos.

Kadangi DTLS turi trūkumų realaus laiko ribotų išteklių lokalaus tinklo taikymui, siūlomas naujas protokolas, kurio pagrindinis tikslas išspręsti DTLS trūkumus, transporto priemonių lokalaus tinklo taikymui. Taip pat panašios problemos egzistuoja ir su kitais rinkoje esančiais protokolais, tokiais kaip MQTT. Kadangi tai standartiniai protokolai skirti komunikacijai internetu ir yra naudojantys sertifikatus, ribotų išteklių sistemose kyla resursų trukumo problema. Todėl pasirinkta kurti naują protokolą. Detaliau apie sprendžiamas problemas aprašyta 2.1 poskyryje.

# 2. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo architektūra

Šiame skyriuje aprašoma siūlomo sprendimo sprendžiama problema, vizija, architektūra, kokie objektai komunikuoja, kurie objektai atsakingi už sesijos raktų generavimą, jų apsikeitimą. Aprašoma metodologija visų veiksmų reikalingų atlikti autentifikaciją, šifravimą ir duomenų siuntimą.

## 2.1. Problema išsprendžiama transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodu

Siūlomas metodas ir panaudojimo metodika skirta išspręsti šiandieną paplitusių protokolų silpnumus IOT sistemose. Šiandien plačiai naudojamas DTLS (Datagram Transport Layer Security) protokolas, kuris yra paremtas TLS protokolu, reikalauja „X.509“ sertifikatų tam, kad serveris ir klientas galėtų autentifikuotis saugiai ir patikimai. Įterptinėse IOT sistemose tai sukelia kelias problemas:

1. Atminties ištekliai – kiekvienam tinklo įrenginiui, kad sėkmingai sudaryti komunikaciją reikia sertifikato, todėl jutikliai ir jų posistemės turi palaikyti pakankamą atminties kiekį. Tai padidina gamybos kaštus ir įrenginio kompleksiškumą.

2. Sertifikatų valdymas – kadangi kiekvienas įrenginys tinkle turi turėti sertifikatą, sistemoje turi būti įgyvendintas sertifikatų valdymas, atšaukimas, naujų sertifikatų generavimas ir kita.

3. Sparta – kadangi sertifikatai naudoja viešo rakto kriptografiją, kiekviename sistemos tinklo įrenginyje naudojami vienas privatus ir vienas viešas raktas. Tai padidiną reikalaujamų veiksmų kiekį atliekant kriptografines funkcijas. Taip pat viešo rakto kriptografijoje raktai yra standartiškai ilgesni, kas prailgina šifravimą ir iššifravimą. Raktų valdymas ir papildomos kriptografinės funkcijos reikalauja daugiau skaičiavimo galios negu simetrinė kriptografija. Dėl šių priežasčių įterptinėse sistemose, ypač transporto priemonių įterptinėse sistemose, dėl išteklių ribojimo gali būti pastebimas tinklo stabdymas, paketų vėlavimas. Taip pat, dėl padidėjusio funkcijų kiekio, įterptinių sistemų įrenginiai gali pritrukti atminties resursų ir greičiau išnaudoti baterijos energijos šaltinį.

4. Energijos ištekliai – energijos ištekliai didėja kartu su atliekamų skaičiavimų kiekiu, todėl sertifikatų valdymui ir skaičiavimams atliekamiems naudojant sertifikatus reikia daugiau energijos.

Atsižvelgiant į kitus protokolus, tokius kaip MQTT, pastebima, kad saugumui užtikrinti taip pat naudojami „X.509“ sertifikatai, todėl išlieka tokios pat problemos. Kadangi siūlomas metodas sprendžia įterptinių sistemų belaidžio tinklo saugumo problemą transporto priemonių ribotų išteklių aplinkoje, dėl išvardintų resursų problemų šiandien plačiai naudojami protokolai yra neefektyvūs.

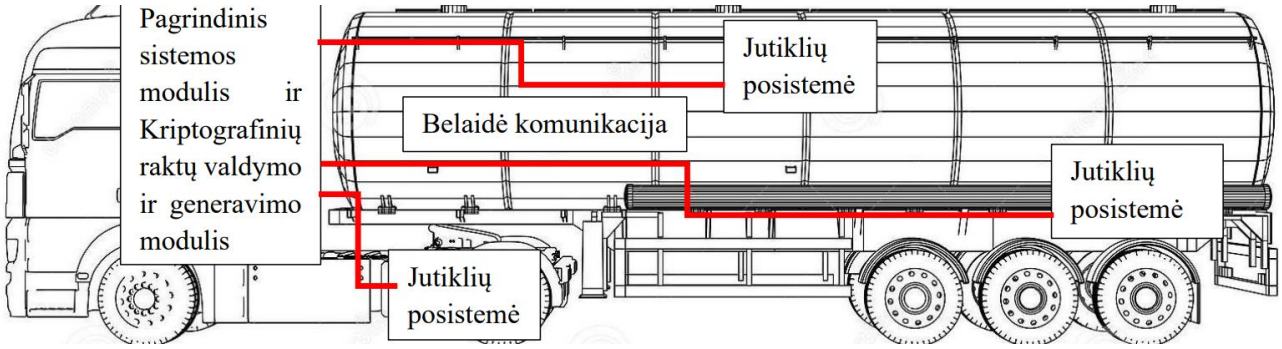
## 2.2. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo projektavimo kriterijai

Tam, kad sėkmingai įgyvendinti siūlomą sprendimą, projektavimo metu iškelti šie kriterijai:

1. Projektuojamas protokolas turi naudoti periodiškai keičiamus kriptografinius simetrinius sesijos raktus, kurie yra išvedami iš pagrindinio rakto ir naudojami sudaryti komunikacijai tik tarp dviejų tinklo grandinės dalių.
2. Projektuojamas protokolas turi būti lengvesnis už šiandien plačiai paplitusius TLS ir DTLS protokolus skaičiavimo galios, atminties sąnaudų ir energijos sąnaudų atžvilgiu.
3. Projektuojamas protokolas turi dirbti neviešinant kriptografinių raktų kūrimui naudojamų parametrų. Perduodant naują sesijos raktą, tinkle siunčiamas tik šifruotas kriptografinis raktas.
4. Projektuojamo protokolo žinutės pateikia siuntėjo autentifikacijos reikšmę MAC.

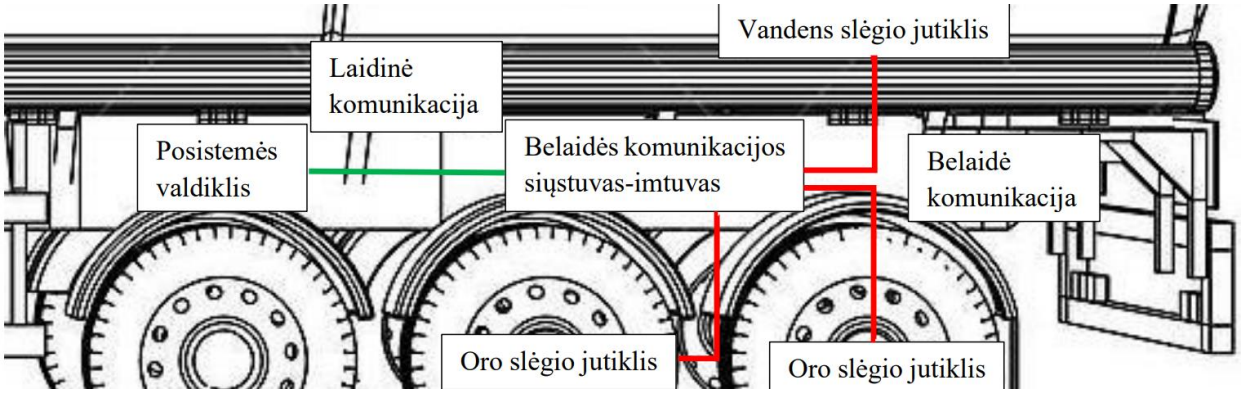
## 2.3. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo vizija

Sistema sudaryta iš vieno pagrindinio sistemos modulio, kuris yra atsakingas už sistemos veikimą, duomenų rinkimą iš jutiklių belaidžiu metodu, kelių skirtingų jutiklių posistemių ir vieno raktų valdymo modulio atsakingo už sesijos raktų generavimą ir atnaujinimą. Metodo pirminė vizija pateikta 1 pav.



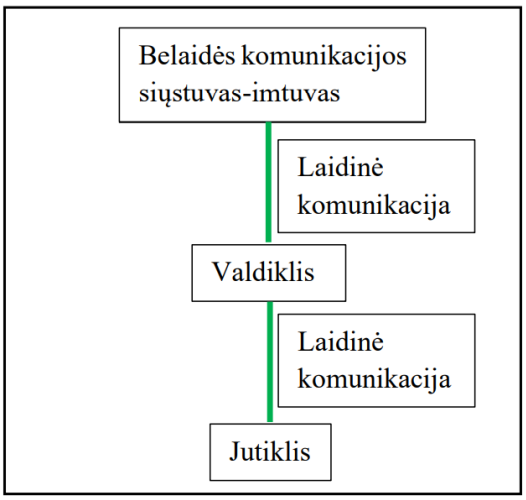

**1 pav.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo apibendrinta vizija

Jutiklių posistemė sudaryta iš posistemės valdiklio laidu sujungto su belaidės komunikacijos siųstuvu-imtuvu, kuris suteikia valdikliui galimybę komunikuoti su jutikliais ir pagrindiniu sistemos valdikliu. Jutiklių posistemės vizija pavaizduota 2 pav.



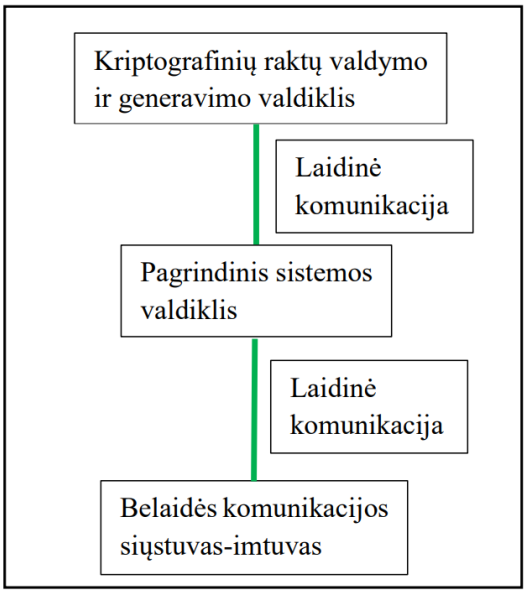

**2 pav.** Metodo jutiklių posistemės vizija

Jutikliui taip pat skiriamas valdiklis ir belaidės komunikacijos siųstuvas imtuvas. Valdiklis šiuo atveju valdo komunikacijos siųstuvą-imtuvą, jutiklio duomenis ir pačią komunikaciją su posistemės valdikliu. Jutiklio vizija pavaizduota 3 pav.




**3 pav.** Jutiklio vizija

Pagrindinis sistemos modulis sudarytas iš pagrindinio valdiklio laidine komunikacija sujungto su kriptografinių raktų valdymo valdikliu ir belaidės komunikacijos siųstuvo-imtuvo. Pagrindinio sistemos modulio vizija pavaizduota 4 pav.

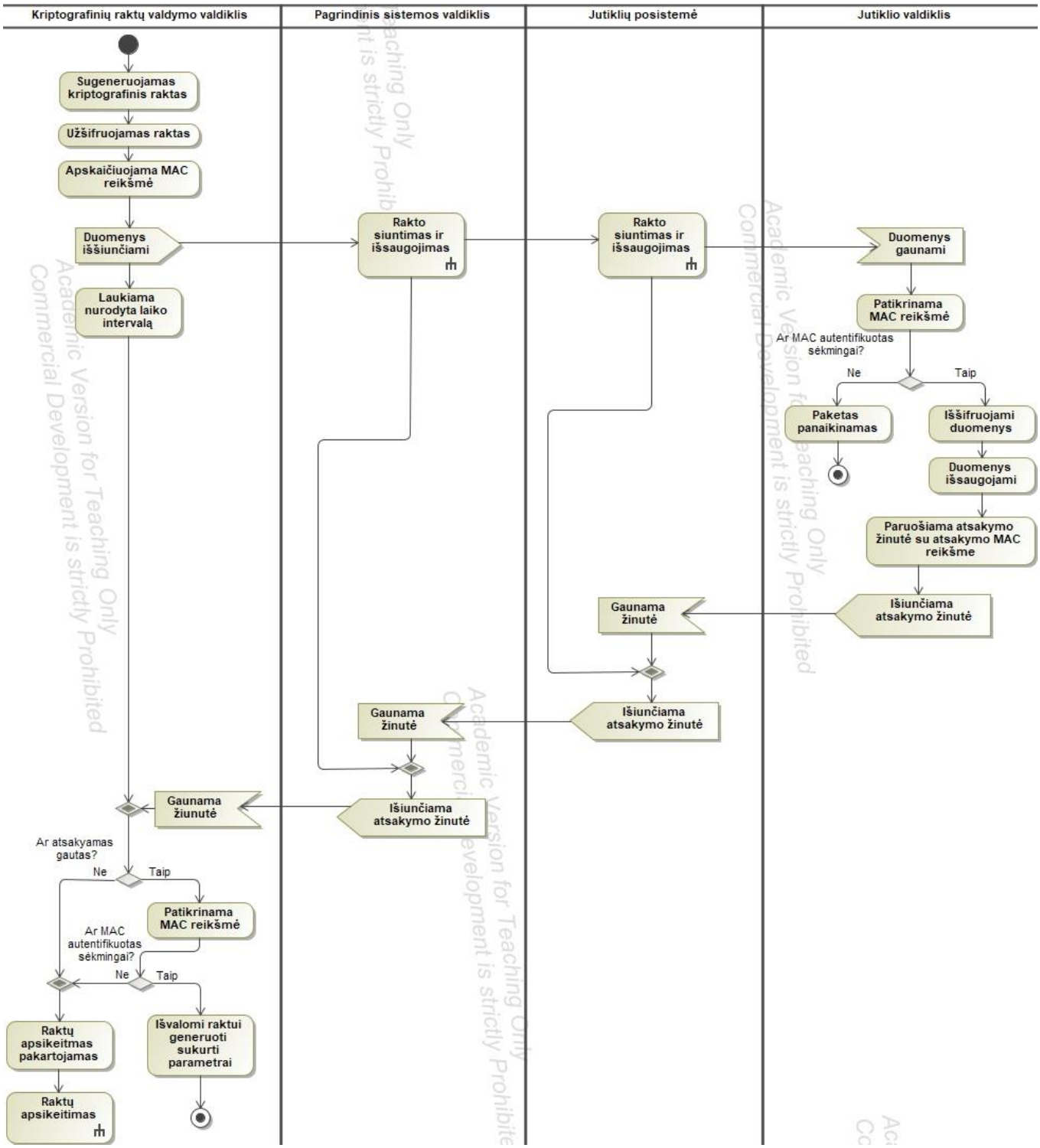



**4 pav.** Pagrindinio valdiklio ir kriptografinių raktų valdymo modulių vizija

## 2.4. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo kriptografinių raktų generavimo vizija

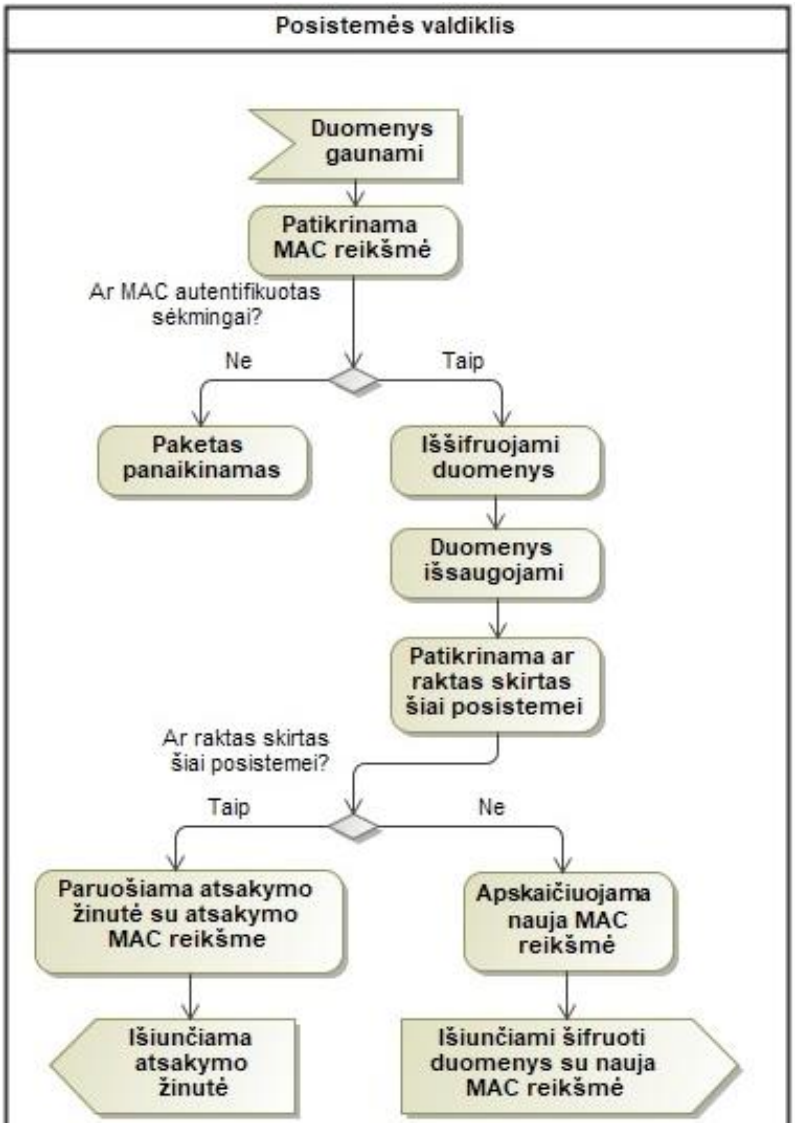
Sesijos raktai yra sugeneruojami kriptografinių raktų valdiklio. Sugeneruoti raktai yra užšifruojami pagrindiniu raktu, žinutei yra sugeneruojama MAC reikšmė ir viskas pateikiama pagrindiniam valdikliui nusiųsti atitinkamai sistemos posistemei. Posistemei gavus duomenis yra patikrinama MAC reikšmė, jeigu ji yra autentifikuojama, sesijos raktas iššifruojamas ir išsaugomas atmintyje iki kito rakto atnaujinimo. Tai atlikus apskaičiuojama atsakymo MAC reikšmė, kuri yra

užšifruojama ir išsiunčiama pagrindiniam valdikliui. Rakto išsaugojimas ir persiuntimas detaliau pavaizduotas 6 pav. Pagrindinis valdiklis šią reikšmę nusiunčia kriptografinių raktų valdikliui, kuris ją iššifruoja ir patikrina. Jeigu atsakymo MAC reikšmė autentifikuojama, valdikliui pateikiama, kad sesijos raktas sėkmingai gautas. Jeigu MAC reikšmėje yra klaidų, sugeneruojamas naujas sesijos raktas ir siunčiama dar kartą. Sesijos raktų apsikeitimo veiksmai pavaizduoti 5 pav. ir 6 pav.




**5 pav.** Sesijos raktų apsikeitimo vizija

Kiekviena posistemė patikrina gautų paketų MAC reikšmes ir pašalina paketą jeigu MAC reikšmė neautentifikuota sėkmingai. Taip pat kiekviena posistemė, sėkmingai autentifikavus MAC reikšmę, atlieka iššifravimą, atlieka šifravimą prieš siunčiant duomenis ir atlieka MAC reikšmių suskaičiavimą sėkmingai užšifravus duomenis. Šių veiksmų seka detaliau pavaizduota **6 pav.**




**6 pav.** Rakto išsaugojimas ir atsakymo arba rakto persiuntimas

Tam, kad periodinis kriptografinių sesijos raktų pakeitimas vyktų be vėlavimų, pasirinkta sugeneruoti raktus prieš inicializuojant raktų apsikeitimą. Priėmus tokį sprendimą sistema visuomet saugoja sugeneruotus naujus sesijos raktus, kurie gali būti išsiųsti iškarto. Kadangi tinklo apkrova gali būti didelė, priklausomai nuo sistemos naudojimo, šis sprendimas nevėlina duomenų perdavimo.

## 2.5. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo kriptografinių raktų valdymo architektūra

Siūlomame sprendime kriptografinių raktų valdymui skirtas valdiklis, kuris atsakingas už kriptografinių raktų valdymą. Valdiklis periodiškai privalo sugeneruoti naujus sesijos raktus kiekvienai sistemos posistemei ir juos pateikti pagrindiniam sistemos valdikliui, kuris šiuos raktus išsisaugo ir persiunčia atitinkamai posistemei. Sesijos raktai yra generuojami kiekvieną kartą iš naujo apskaičiavus atsitiktinius skaičius. Kai posistemei yra sukuriami sesijos raktai, jie yra užšifruojami

pagrindiniu sistemos raktu, kuris iš anksto yra įdiegtas į kiekvieną sistemos dalį. Taip pat sesijos raktams yra apskaičiuojama MAC reikšmė, kuriai kriptografinių raktų valdymo valdiklis laukia atsakymo. Atitinkamai sistemos posistemei gavus paketą su kriptografiniais sesijos raktais, posistemės valdiklis turi patikrinti gautą MAC reikšmę. Jeigu MAC reikšmė autentifikuota, valdiklis privalo suskaičiuoti naują MAC atsakymo reikšmę, patvirtinančią sėkmingą kriptografinių raktų gavimą. MAC reikšmė turi būti nusiųsta kriptografinių raktų valdymo valdikliui, kuris patikrina gautą MAC atsakymo reikšmę ir jeigu ji sėkmingai autentifikuojama, raktų apsikeitimas su atitinkama posisteme yra baigtas. Jeigu MAC atsakymo žinutė neautentifikuojama sėkmingai, kriptografinių raktų valdymo valdiklis sugeneruoja naujus sesijos raktus ir ciklas kartojasi tol, kol gauta MAC atsakymo žinutė yra autentifikuojama.

## 2.6. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo programinė architektūra

Šiame poskyryje aprašomi reikalingi parametrai ir funkcijos kriptografinių raktų generavimui. Pateikiama programinė architektūra siūlomam sprendimui.

### 2.6.1. Reikalingi parametrai ir funkcijos

Siūlomame sprendime, kriptografinių raktų valdymo valdiklis atsakingas už raktų generavimą ir perdavimą, tačiau tai atlikti saugiai reikalingi raktų generavimo parametrai, kurie nėra viešai prieinami arba perduodami tinklu. Tam apsibrėžiami tokie parametrai reikalingi užtikrinti saugų raktų generavimą, perdavimą ir saugią komunikaciją:

1. Identifikacija – kiekviena sistemos dalis ir posistemė, kuri komunikuoja sistemos tinkle, turi turėti savo identifikacinę reikšmę tam, kad sistemoje būtų galima atskirti atskiras posistemes.

2. Žinutės skaitiklis – skaitiklis saugantis išsiųstų žinučių kiekį, naudojamas kaip reikšmė sujungiama su sesijos raktais taip sinchronizuojanti komunikaciją tarp dviejų posistemių ir pakeičianti sesijos siunčiamų duomenų šifravimą. Keičiamas šifravimas todėl, kad siunčiant tuos pačius duomenis šifravimo algoritmas nusiųs tą pačia šifruotą bitų seką, kas gali būti panaudota atakuotojo, tačiau keičiant šifravimą pagal žinučių skaitiklį ta pati nešifruota teksto žinutė bus užšifruota skirtingai kiekvieną kartą.

3. Pagrindinis raktas – pagrindinis raktas skirtas išvesti visus sesijos raktus periodiškai. Šis raktas nekinta, nėra siunčiamas ir su juo nėra šifruojami duomenys. Pats sesijos raktas turi būti sugeneruotas ir išsaugotas kriptografinio raktų valdymo valdiklio įdiegimo į sistemą metu arba keitimo metu. Šis raktas turi būti saugomas visų sistemoje komunikuojančių valdiklių, šis raktas naudojamas sesijos raktų šifravimui ir iššifravimui.

4. Sesijos raktas – kiekviena posistemė komunikuojanti su kita sistemos dalimi turi turėti tos komunikacijos tunelio sesijos raktą. Sistemoje negali būti du vienodi sesijos raktai, kurie naudojami daugiau nei vienos posistemės. Viena posistemė gali naudoti tą patį sesijos raktą su visais posistemės jutikliais, tačiau komunikuojant su kita posisteme, sesijos raktas turi būti kitoks. Patys sesijos raktai turi būti keičiami periodiškai, taip apsisaugant nuo galimos brutalios jėgos atakų sesijos raktų atradimui.

5. Nešifruotas tekstas – kiekviena sistemos dalis duomenis siuntimui paruošia nešifruotu tekstu.

6. Šifruotas tekstas – kiekviena sistemos dalis prie išsiunčiant duomenis turi užšifruoti visus siunčiamus duomenis tuo metu saugomais sesijos raktais.

7. Rakto srautas – rakto srautas naudojamas užšifruoti duomenis naudojant būlio algebros funkcijas. Srautas sugeneruojamas iš sesijos rakto.

8. MAC reikšmė (Message Authentication Code) – skirta autentifikuoti komunikacijos tunelio pusėms. MAC reikšmė turi būti suskaičiuojama kiekvienam siunčiamam paketui.

9. Parametrai rakto generavimui – kiekvieną kartą rakto generavimui sukuriami atsitiktiniai parametrai, kurie naudojami rakto išvedimui iš pagrindinio rakto.

Raktų generavimui, šifravimui ir komunikacijos saugumui užtikrinti reikalingos šios funkcijos:

1. Rakto išvedimo funkcija (KDF) – tam, kad iš pagrindinio rakto išvesti sesijos raktą kiekvienai sistemos posistemei, reikia tai atliekančios funkcijos.

2. Santraukos funkcija suskaičiuoti ir patikrinti MAC reikšmei – kiekviena komunikuojanti sistemos posistemė prie kiekvieno paketo prideda MAC reikšmę, todėl reikia įgyvendinti santraukos funkciją, kuri kiekvieną kartą suskaičiuotų skirtingą MAC reikšmę. Taip pat reikia įgyvendinti funkciją MAC reikšmės patikrinimui ir autentifikavimui.

3. Simetrinio šifravimo algoritmo ir simetrinio iššifravimo algoritmo funkcijos – kadangi komunikacijoje naudojami sesijos raktai, komunikuojančios pusės gali naudoti vienodą raktą, taip sutaupant atminties ribotų išteklių sistemoje. Todėl užtenka funkcijos ar funkcijų šifruojančių ir iššifruojančių paketų duomenis naudojant tą patį sesijos raktą.

Apibrėžti parametrai ir funkcijos naudojami periodiškai generuoti kriptografinius raktus visiems sistemos tinklo įrenginiams. Kiekvienas posistemės įrenginys komunikuoja tik su vienu posistemės valdikliu, todėl šios posistemės įrenginiai gali naudoti vieną ir tą patį kriptografinį raktą to kriptografinio rakto naudojimo ciklu. Po nustatyto laiko periodo šis raktas pakeičiamas visuose posistemės įrenginiuose. Toks sprendimas sutaupo atminties resursus posistemės valdikliui, kadangi nereikia saugoti atskirų raktų kiekvienam posistemės įrenginiui. Posistemės valdiklis papildomai turi turėti dar vieną kriptografinį raktą sudaryti komunikacijai su sekančiu tinklo grandinės įrenginiu. Taip atskiriamas vienos posistemės įrenginių potinklis nuo kelių posistemių potinklio.

Kriptografiniai raktai keičiami periodiškai tam, kad išvengti galimos brutalios atakos prieš kriptografinį raktą. Kadangi tinkle naudojami ne slaptažodžiai, o kriptografiniai raktai, jie yra ilgesni, todėl brutalios jėgos ataką sėkmingai įgyvendinti reikia daugiau laiko. Todėl, periodiškai keičiant kriptografinius raktus, brutalios jėgos atakos sėkmė buvusiam raktui nekelia grėsmės sistemai. Taip pat šifravimo metu naudojama druskos (salt) pridėjimas, kas išsprendžia problemą siunčiant tokius pačius, nepakitusius duomenis. Tokiu atveju šifruotas tekstas yra nepakitęs, tačiau naudojant druskos reikšmę pagal paketų skaitliuką, šifruojant tą pačia duomenų reikšmę, rezultatas nelygus.

### 2.6.2. Raktų išvedimo funkcijos pasirinkimas

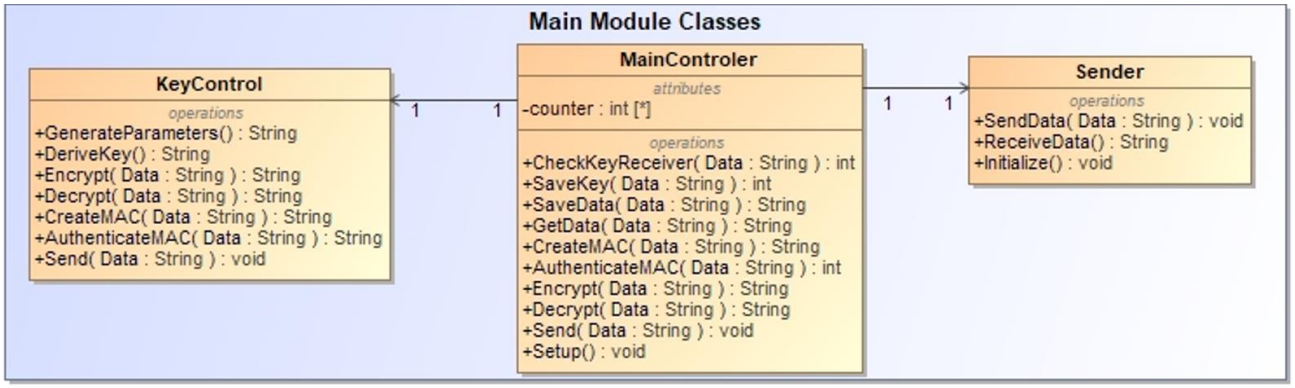
Siūlomame sprendime pasirinkta naudoti HKDF (simple key derivation functions based on HMAC message authentication code). Pagrindinė HKDF metodas yra „išskleisti, tada išplėsti“ (extract-then-expand) paradigma. Pirmuoju funkcijos vykdymo etapu, funkcija paima pateiktą įvesties raktų medžiagą ir „ištraukia“ iš jos fiksuoto ilgio pseudoatsitiktinį raktą. Antruoju etapu šis pseudoatsitiktinis raktas yra „išplečiamas“ į keletą kitų pseudoatsitiktinių raktų.

HKDF išgauna pseudo atsitiktinį raktą naudojant HMAC santraukos funkciją ir druskos (salt) reikšmę. Tuomet ši funkcija kriptografiškai stiprų raktą generuoja pagal pasirinktą rakto ilgį pakartotinai generuojant santraukos blokus ir juos sujungiant į vieną naują bloką, o pabaigoje apkarpant bloko ilgį į nustatytą naujo rakto ilgį. Tam, kad padidinti saugumą, prieš tai sugeneruotas santraukos blokas yra pridedamas prie naujai generuojamo bloko duomenų, taip padidinant generuojamo rakto išskirtinumą. Šio projekto metu pasirinkta naudoti „HMAC-SHA256“ funkciją HKDF įgyvendinimui.

MAC reikšmių skaičiavimui pasirinkta naudoti HMAC (Hash Message Authentication Code). HMAC skaičiuoja žinutės autentifikacijos reikšmę naudojant kriptografinių santraukų funkcijas, kurioms reik kriptografinio rakto. Naudojant HMAC galima patikrinti žinutės vientisumą ir autentiškumą. Tai panaikina sudėtingą viešos kriptografijos infrastruktūrą, paliekant HMAC funkcijai reikalingų kriptografinių raktų apsikeitimą komunikuojančiomis šalims, kurios yra atsakingos už saugų ir patikimą komunikacijos tunelį tam, kad galėtų apskaičiuoti santraukai reikalingą kriptografinį raktą.

### 2.6.3. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo pagrindinio valdiklio programinė architektūra

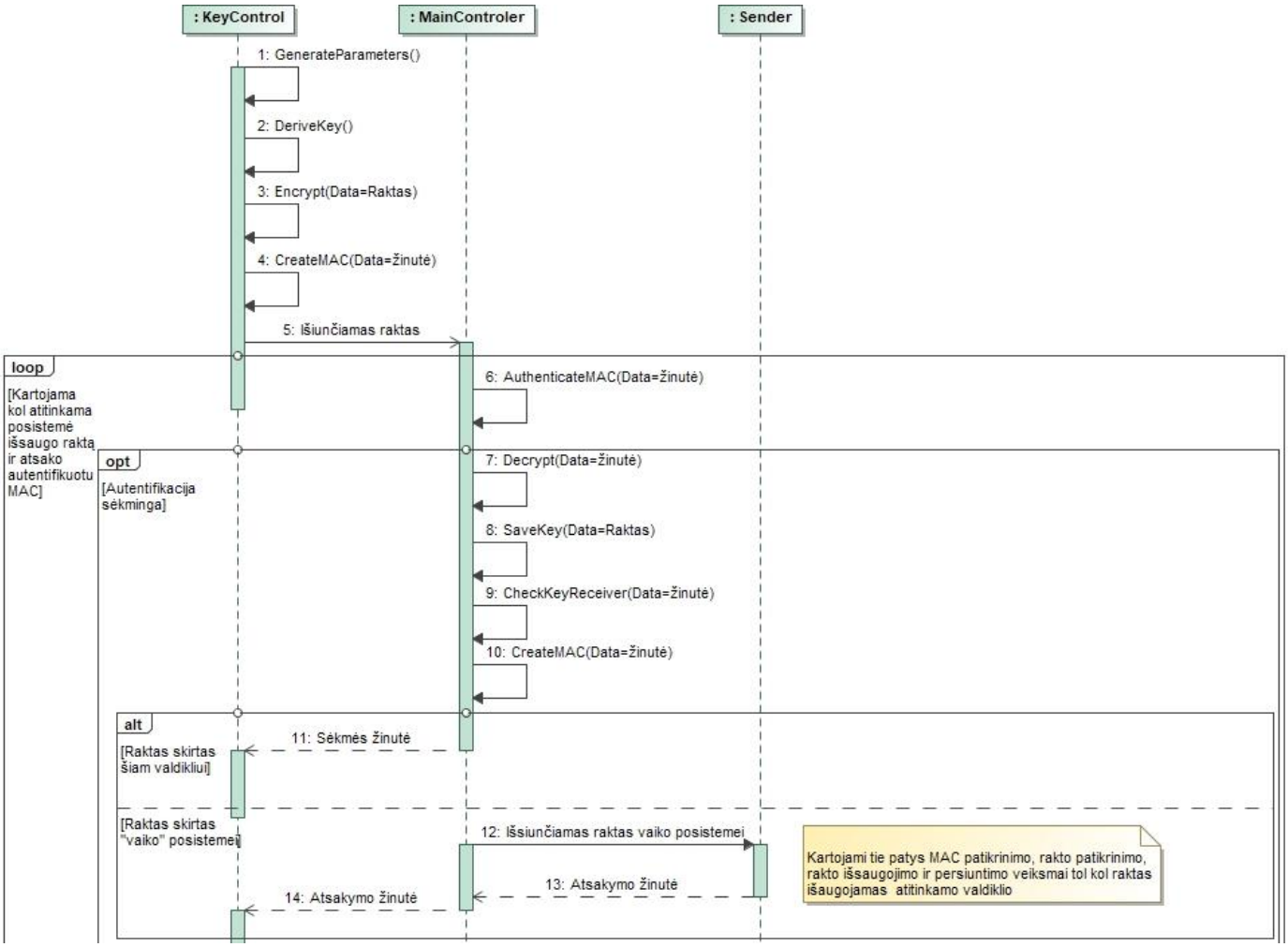
Kriptografinių raktų valdymo modulis yra pagrindinio modulio dalis, todėl kriptografinių raktų modulis komunikuoja su pagrindiniu valdikliu. Pagrindinio modulio klasių diagrama pavaizduota 7 pav. „KeyControl“ klasė skirta kriptografinių raktų valdikliui ir yra sudaryta iš funkcijų skirtų sesijos raktų generavimui, šifravimui ir siuntimui. „MainControler“ klasė skirta pagrindiniam valdikliui, kuris valdo duomenų saugojimą, sistemos veikimą, sesijos raktų perdavimą. „Sender“ klasė skirta belaidės komunikacijos siųstuvui-imtuvui valdyti, aprašomos pagrindinės funkcijos duomenų siuntimui ir gavimui, šifravimas ar MAC reikšmių skaičiavimas neatliekamas.




**7 pav.** Pagrindinio sistemos modulio klasių diagrama

### 2.6.4. Kriptografinių raktų generavimo ir išsaugojimo seka

Kriptografiniai raktai sukuriami pradžioje sugeneruojant atsitiktinius parametrus „KeyControl“ klasėje. Naudojant šiuos parametrus išvedamas sesijos raktas iš pagrindinio rakto. Sesijos raktas užšifruojamas pagrindiniu raktu ir paruoštai žinutei apskaičiuojama MAC reikšmė, kuri pridedama prie žinutės. Žinutė nusiunčiama pagrindiniam valdikliui, kuris patikrina MAC reikšmę, jeigu MAC reikšmė autentifikuojama sėkmingai, raktas iššifruojamas ir išsaugojamas. Toliau patikrinama ar raktas skirtas šiam valdikliui, jeigu tai ne galutinis valdiklis, apskaičiuojama nauja žinutė ir nusiunčiamas raktas tolimesniam šio valdiklio vaiko posistemes valdikliui, kuris atlieka tuos pačius veiksmus. Jeigu raktas išsaugotas, kriptografinių raktų moduliui turi sugrįžti sėkmės žinutė. Šio proceso sekų diagrama pavaizduota 8 pav.




**8 pav.** Kriptografinių raktų generavimo ir išsaugojimo sekų diagrama

## 2.7. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo projektavimo išvados

1. Siūlomas sprendimas pranašesnis už analizuotą DTLS protokolą atminties sąnaudų kiekiu todėl, kad naudojami simetriniai kriptografiniai raktai vietoje sertifikatų. Siūlomas protokolas pranašesnis energijos sąnaudų kiekiu ir skaičiavimo galios atžvilgiu, nes kriptografiniai raktai yra trupesni, dėl to reikia atlikti mažiau skaičiavimų atliekant kriptografines funkcijas.

2. Protokolo architektūra paprastesnė, nes nereikia įgyvendinti sertifikatų valdymo, atnaujinimo, panaikinimo funkcijų. Taip pat, DTLS protokolas yra skirtas standartiniam naudojimui, atvirame tinkle, užtikrinant komunikacijos saugumą internete. Transporto priemonės lokaliam tinklui prieigos prie interneto nereikia, todėl siūlomas sprendimas yra pranašesnis ribodamas komunikaciją. Tai leidžia protokolą pritaikyti silpnesniems, skaičiavimo galios atžvilgiu, įrenginiams, kurie taip pat turi mažiau atminties.

3. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodas netinka komunikacijai internetu ar sistemoms, kurios turi bent vieną posistemę ar įrenginį, kuris yra už lokalaus tinklo ribų. Taip pat, siūlomas sprendimas netinka dažnai kintančiai tinklo architektūrai, kurioje gali prisidėti naujų įrenginių nenumatytu laiku, kadangi visi tinklo įrenginiai turi turėti atmintyje išsaugotą nekintantį kriptografinį raktą, kuris yra naudojamas sesijos raktų apsikeitimui. Kadangi šis raktas yra neviešinamas, nauji įrenginiai negali būti pridėti prie sistemos, šio rakto nežinant, todėl nauji įrenginiai gali būti pridedami tik sistemos konfigūravimo metu. Be to, kiekvienai sistemos tinkle siunčiamai žinutei yra suskaičiuojama MAC reikšmė autentifikuojanti žinutės siuntėją. Šiai funkcijai naudojama efektyvi ir skaičiavimo galios atžvilgiu lengva MAC reikšmėms skaičiuoti skirta CMAC funkcija.

# 3. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo taikymas ir prototipas

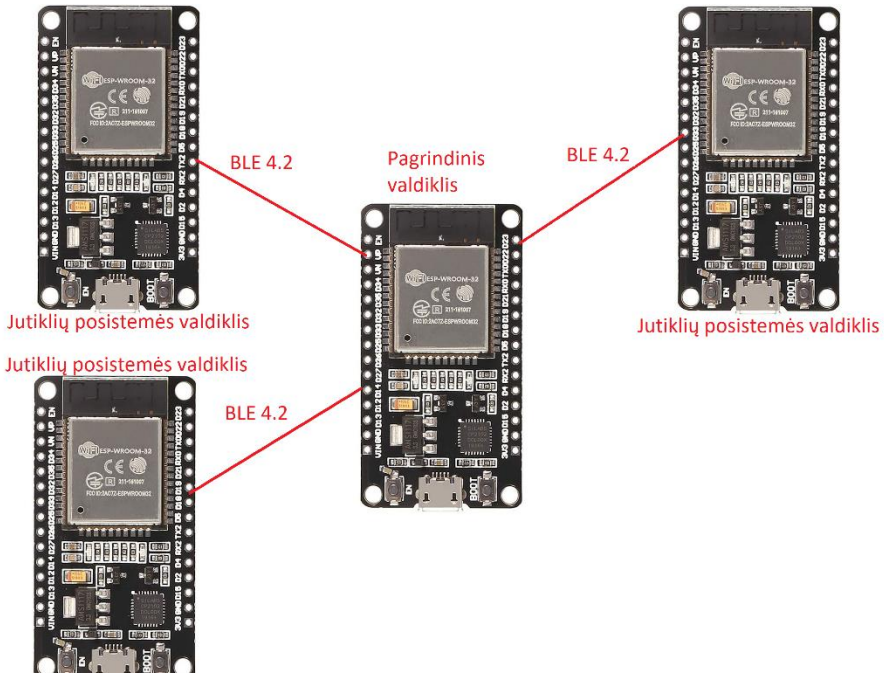
Šio projekto metu kuriamas metodas taikomas transporto lygmenyje ir yra skirtas naudoti duomenų perdavimo technologijos ar protokolo saugumui padidinti. Tarp šių technologijų gali būti „Bluetooth“, „Zigbee“, „WiFi“ ir kiti. Pateiktos diagramos jutiklių nuskaitymui pateikia metodo naudojimo viziją. Metodas gali būti taikomas bet kokioje IOT sistemoje reikalaujančioje komunikacijos su sesijos raktais. Privalomai turi būti modulis atsakingas už kriptografinių raktų valdymą, pagrindinis valdiklis sistemos valdymui ir bent viena posistemė su kuria pagrindinis valdiklis komunikuotų. Šis metodas gali būti pritaikomas laidinėse technologijose, tačiau dėmesys skirtas belaidėms technologijoms kadangi joms kyla daugiau saugumo grėsmių.

## 3.1. Prototipo realizacijos topologija

Šiame poskyryje aprašoma fizinė prototipo topologija, kuri įgyvendinama realizacijos metu ir toliau bus naudojama siūlomo sprendimo tyrimo metu.

### 3.1.1. Prototipo realizacijos vizija

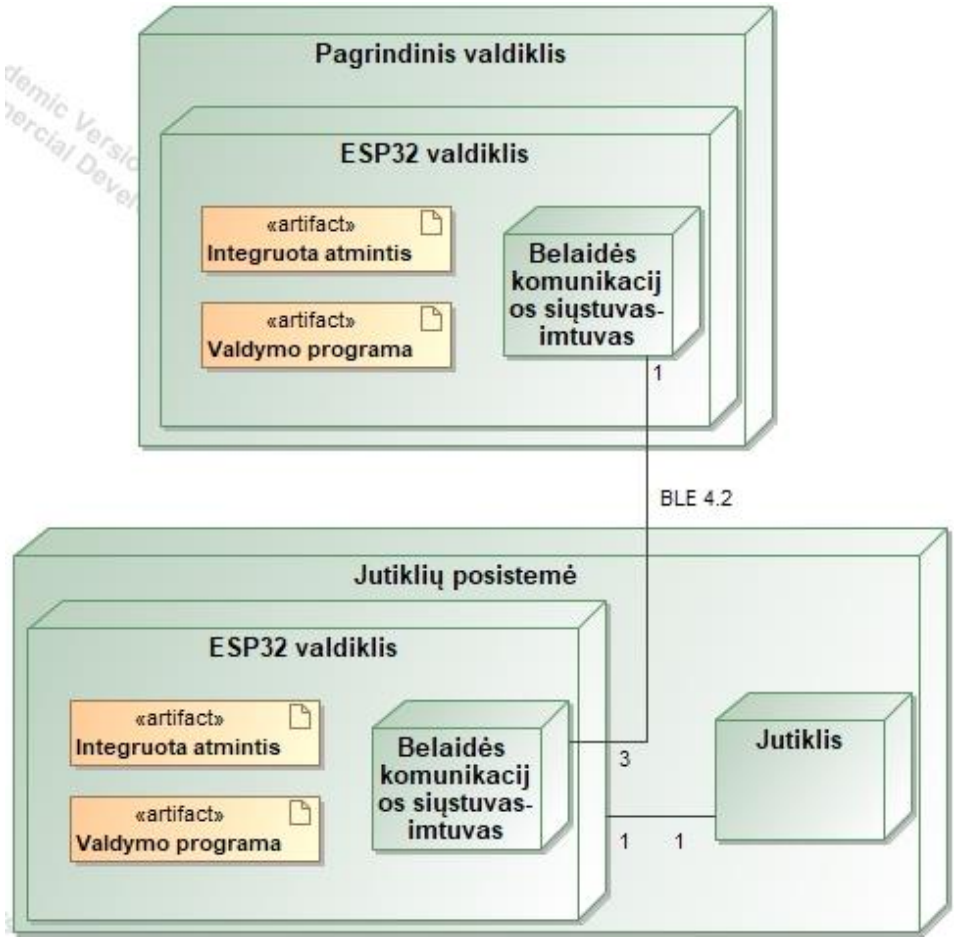
Prototipo realizacijai nuspręsta įgyvendinti keturis valdiklius: vienas pagrindinis valdiklis, kuris yra atsakingas už kriptografinių raktų generavimą ir paskirstymą kiekvienam sistemos valdikliui, trys posistemių valdikliai, kiekvienas turi savo atskirtus sesijos raktus su pagrindiniu valdikliu. Visa komunikacija šifruojama sesijos raktais. Komunikacija tarp valdiklių sudaroma „BLE v4.2“ (Bluetooth Low Energy) technologija taip, kaip pavaizduota 9 pav.




**9 pav.** Prototipo realizacijos topologijos vizija

### 3.1.2. Prototipo realizacijos tinklo įrenginių architektūra

Kaip pavaizduota prototipo realizacijos diegimo diagramoje pavaizduotoje 10 pav., prototipo sistema sudaryta iš dviejų tipų įrenginių: pagrindinio valdiklio ir jutiklių posistemės. Šie du įrenginių tipai tarpusavyje komunikuoja naudodami belaidės komunikacijos siųstuvą-imtuvą, naudojant „Bluetooth Low Energy v4.2“ technologiją. Taip pat, šioje diagramoje pavaizduotas ryšys 1 su 3, kadangi prototipo realizacijoje yra vienas pagrindinis valdiklis ir trys jutiklių posistemių valdikliai. Kiekvienas sistemos topologijos įrenginys sudarytas iš ESP32 valdiklio, kuriame yra integruotas siųstuvas-imtuvas, atmintis ir įrašoma valdymo programa. Jutiklių posistemė papildomai turi ir jutiklį duomenims generuoti.

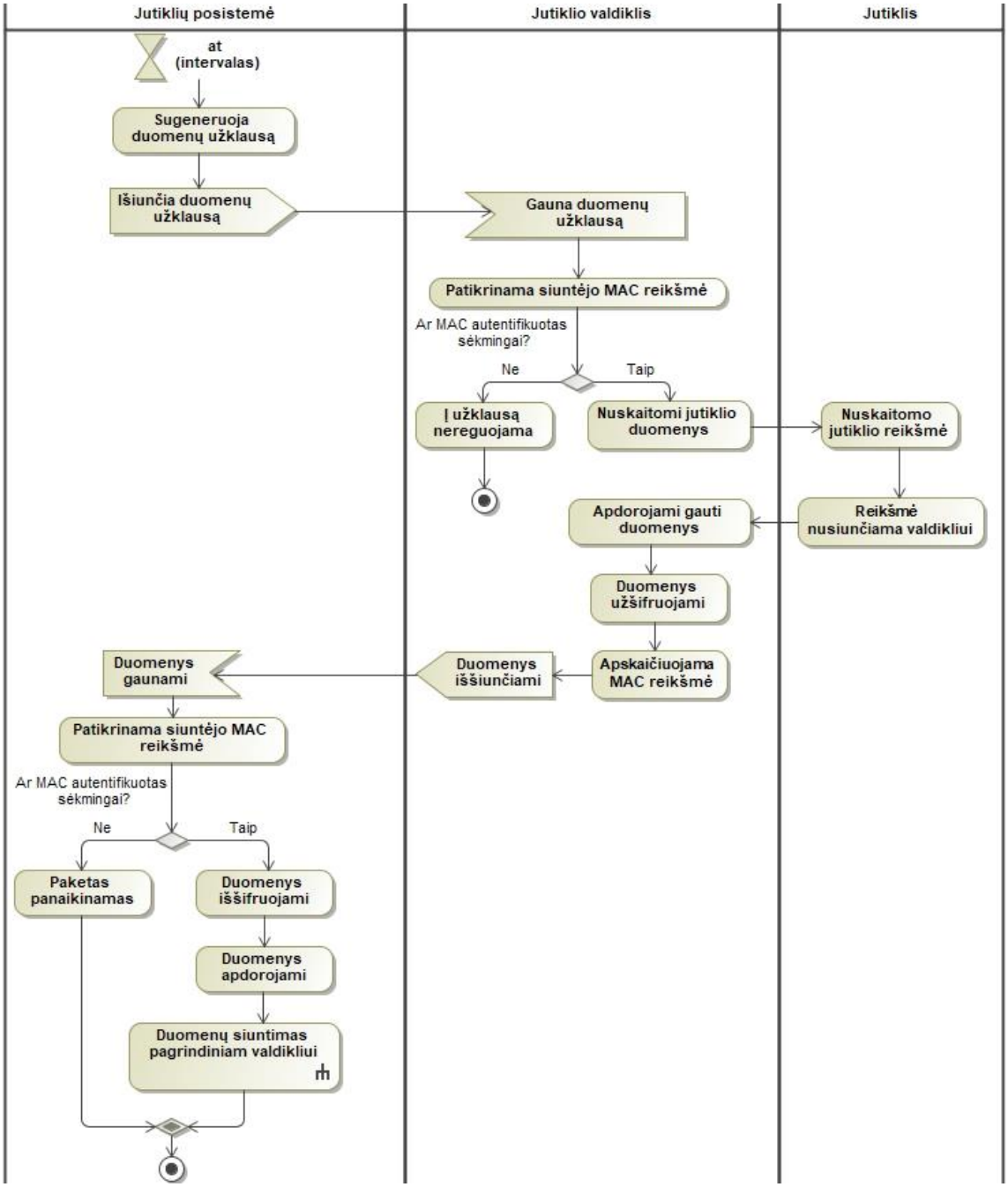



**10 pav.** Prototipo realizacijos diegimo diagrama

Siūlomo sprendimo prototipas realizuotas tik transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo saugumui analizuoti, visa sistema jutiklių stebėjimui ir jų duomenų saugojimui nerealizuota. Visa sistema pavaizduota 13 pav. kuris pateiktas 3.6 poskyryje.

### 3.3. Posistemių duomenų siuntimo transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodu vizija

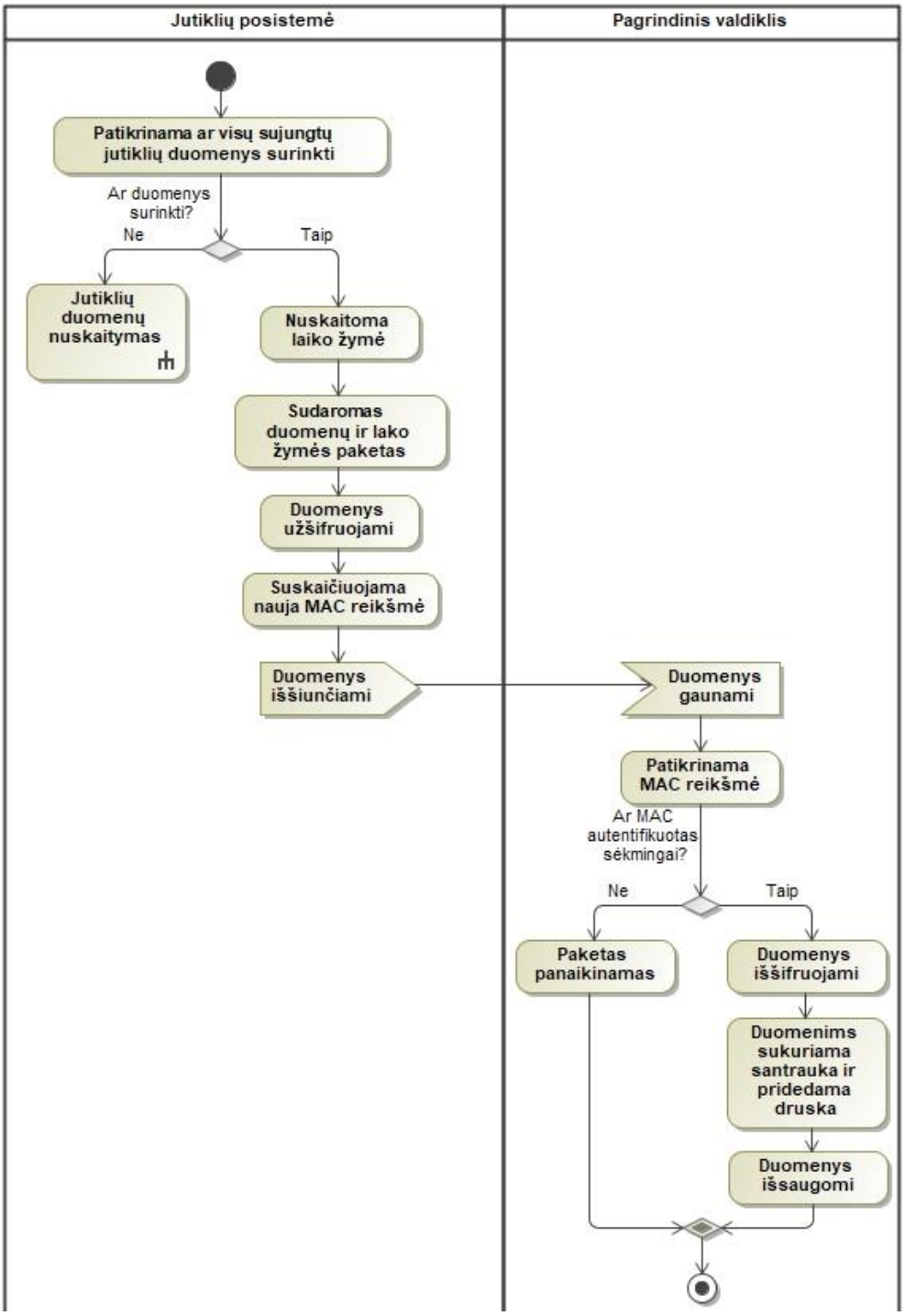
Jutiklio duomenys periodiškai nuskaitomi jutiklio valdiklio. Šie duomenys yra užšifruojami sesijos raktu ir pridedama MAC reikšmę, tai atlikus duomenys, naudojant belaidį komunikacijos siųstuvą-imtuvą, nusiunčiami jutiklių posistemei. Jutiklių posistemė gauna duomenis, naudojant belaidės komunikacijos siųstuvą-imtuvą, tuomet duomenys yra iššifruojami ir patikrinama MAC reikšmė taip, kaip pavaizduota 11 pav.




**11 pav.** Jutiklio duomenų nuskaitymas

Surinkus visų jutiklių duomenis, pridedama laiko žymė kada duomenys nuskaityti ir atliekamas šifravimas, ir MAC reikšmės generavimas. Užšifruoti duomenys ir MAC reikšmė nusiunčiami pagrindiniam sistemos valdikliui naudojant belaidės komunikacijos siųstuvą-imtuvą. Pagrindinis

valdiklis patikrina MAC reikšmę ir iššifruoja duomenis. Jeigu visais žingsniais MAC reikšmė autentifikuotą, duomenims sukuriama santrauka ir pridedama druskos dalis (hash and salt). Šie duomenys išsaugomi duomenų bazėje. Detalesni veiksmai pavaizduoti 12 pav.




**12 pav.** Jutiklių posistemės duomenų išsaugojimas

### 3.4. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo jutiklio architektūra

Siūlomame sprendime jutikliai yra belaidžiai. Tai palengvina jutiklių montavimą ir taisymą. Tačiau dėl to kiekvienas jutiklis tampa sudėtingesnis, kadangi kiekvienam jutikliui reikia valdiklio ir belaidės komunikacijos siųstuvo-imtuvo. Jutiklis yra atsakingas už rodmenų nuskaitymą, tačiau yra

valdomas valdiklio skirto šiam jutikliui. Valdiklis skirtas apdoroti jutiklio duomenis, juos užšifruoti ir pateikti siųstuvui-imtuvui išsiuntimui. Kadangi siūlomame sprendime bandoma užtikrinti belaidėmis technologijomis siunčiamų duomenų saugumą, šis valdiklis yra atsakingas už jutiklio rodmenų šifravimą sistemos valdomais sesijos raktais ir žinutės autentifikacijos kodo – MAC (Message Authentification Code) generavimą. Kiekviena jutiklio modulio siunčiama žinutė susideda iš šifruotų duomenų ir MAC reikšmės, taip suteikiant galimybė žinutės gavėjui patikrinti, autentifikuoti, žinutės siuntėją. Sesijos raktai valdikliui yra pateikiami iš kriptografinių raktų valdymo modulio ir yra išsaugomi trumpalaikėje atmintyje. Kriptografiniai raktai nėra saugomi atmintyje ir kiekvieną kartą paleidus sistemą turi būti gaunami iš sistemos kriptografinių raktų modulio. Jutiklio valdiklis turi laukti sistemos siunčiamų žinučių, kuriomis yra valdomas. Siūlomame sprendime jutiklių moduliai patys duomenų nesiunčia ir laukia duomenų nuskaitymo užklausos. Taip sumažinama galimybė išoriniam objektui pateikti papildomas žinutes į sistemą. Visiškai realizuoto siūlomo sprendimo jutiklio architektūra pavaizduota MAC reikšmę

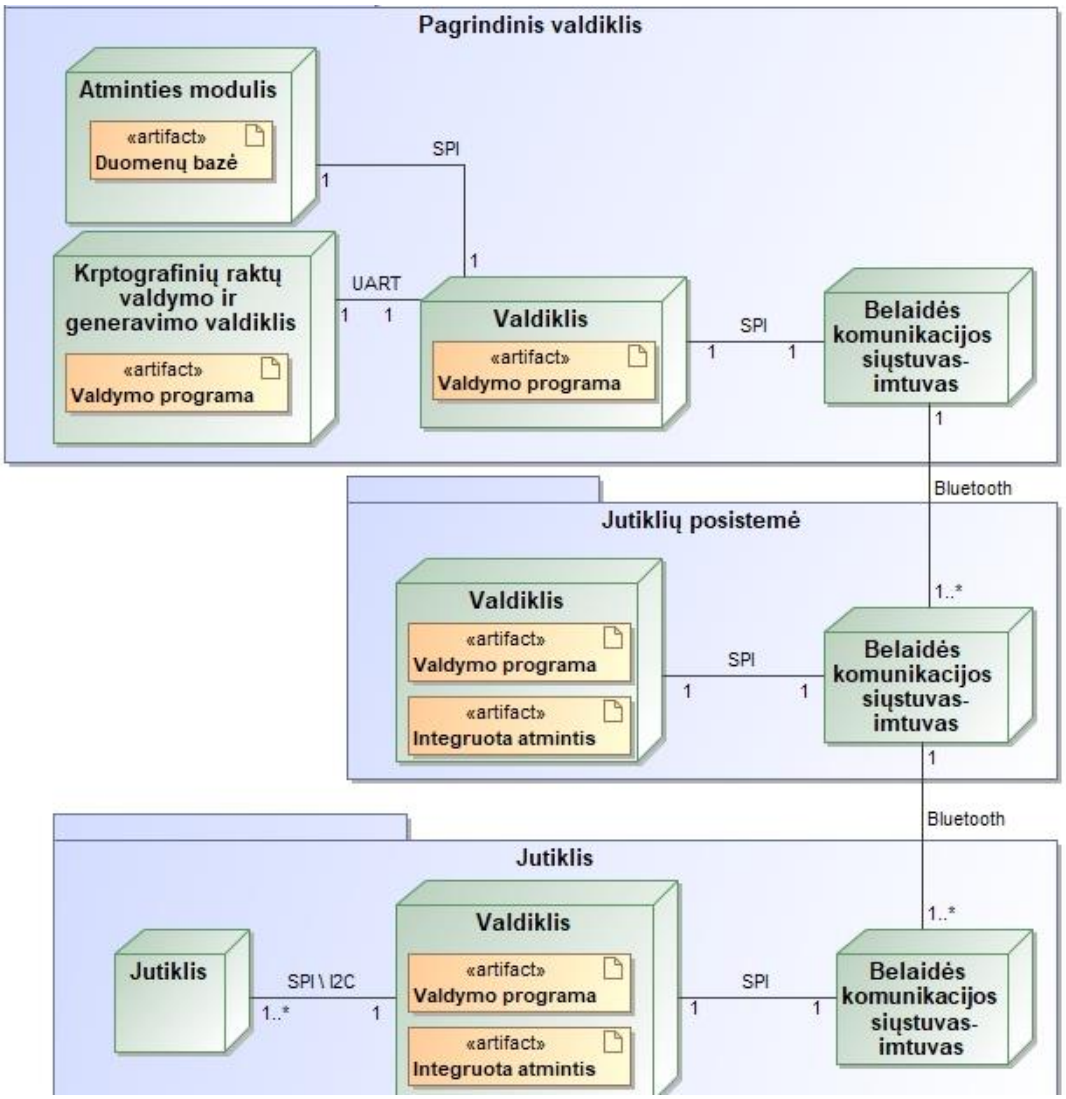
### 3.5. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo jutiklių posistemės architektūra

Siūlomame sprendime jutiklių posistemė skirta rinkti jai priskirtų jutiklių siunčiamus duomenis. Kadangi naudojami belaidžiai jutikliai, posistemei yra skirtas belaidės komunikacijos siųstuvas-imtuvas, kuriuo posistemės valdiklis gali siųsti duomenų užklausas kiekvienam jutikliui. Taip pat, belaidės komunikacijos siųstuvas-imtuvas yra naudojamas sudaryti komunikacijai su sistemos pagrindiniu moduliu, kuriam siunčia jutiklių duomenis ir iš kurio gauna kriptografinius sesijų raktus. Jutiklių posistemė naudoja tuos pačius sesijos raktus su visais posistemei skirtais jutikliais, tačiau kiekvienam jutikliui skirtas identifikacinis numeris. Pasirinkta naudoti tuos pačius sesijos raktus todėl, nes kriptografiniai raktai yra privalomai periodiškai keičiami ir toks sprendimas sumažina reikalaujamus atminties resursus, ypač kai kriptografiniai raktai yra ilgi. Patys sesijos raktai yra saugomi tik trumpalaikėje atmintyje ir kiekvieną kartą paleidus sistemą, raktai turi būti sugeneruojami iš naujo ir pateikiami jutiklių posistemei. Jutiklių posistemė periodiškai siunčia užklausas, kiekvienam jutikliui, gauti duomenims. Taip sumažinama galimybė išoriniam objektui pateikti papildomas žinutes į sistemą. Gavus jutiklio siunčiamus duomenis, posistemės valdiklis patikrina gautą MAC reikšmę tam, kad autentifikuoti siuntėją. Jeigu autentifikavimas sėkmingas, duomenys pridedami į paketą, kuriam skiriama laiko žymė. Šis paketas yra užšifruojamas ir jam suskaičiuojama MAC reikšmė. Tuomet paketas yra nusiunčiamas pagrindiniam sistemos moduliui. Visiškai realizuoto siūlomo sprendimo jutiklių posistemės architektūra pavaizduota 13 pav.

### 3.6. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo pagrindinio sistemos modulio architektūra

Siūlomame sprendime pagrindinis sistemos modulis yra atsakingas už gautų duomenų apdorojimą ir saugojimą iš visų posistemių. Į šį modulį įtrauktas kriptografinių raktų valdymo valdiklis, pagrindinis sistemos valdiklis, belaidės komunikacijos siųstuvas-imtuvas ir atminties modulis duomenų bazei. Duomenų bazė skirta saugoti visą reikalingą informaciją darbui sistemoje. Kadangi duomenų bazė yra pasiekiama tik pagrindinio valdiklio ir kriptografinių raktų valdymo valdiklio, ir nekomunikuoja su išoriniais objektais, gali saugoti visų posistemių kriptografinius raktus ir raktų generavimui reikalingus parametrus. Kriptografinių raktų valdymo valdiklis prie duomenų bazės gali prieiti tik siųsdamas komandas pagrindiniam valdikliui, kuris patikrina užklausas tam, kad neleisti įvykdyti neteisingų užklausų ir pagal jas atlieka veiksmus. Belaidės komunikacijos siųstuvas-imtuvas

skirtas komunikuoti su visomis sistemos posistemis. Pagrindinis valdiklis privalo patikrinti MAC reikšmes kiekvienam gautam paketui tam, kad nustatyti ar siuntėjas yra autentifikuojamas sistemoje. Jeigu klaidų nerasta, valdiklis iššifruoja gautus duomenis ir juos įrašo į duomenų bazę. Kaip ir kitose sistemos dalyse, kriptografiniai raktai turi būti sugeneruojami iš naujo kiekvieną kartą paleidus sistemą. Visiškai realizuoto transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo pagrindinio valdiklio architektūra pavaizduota 13 pav.



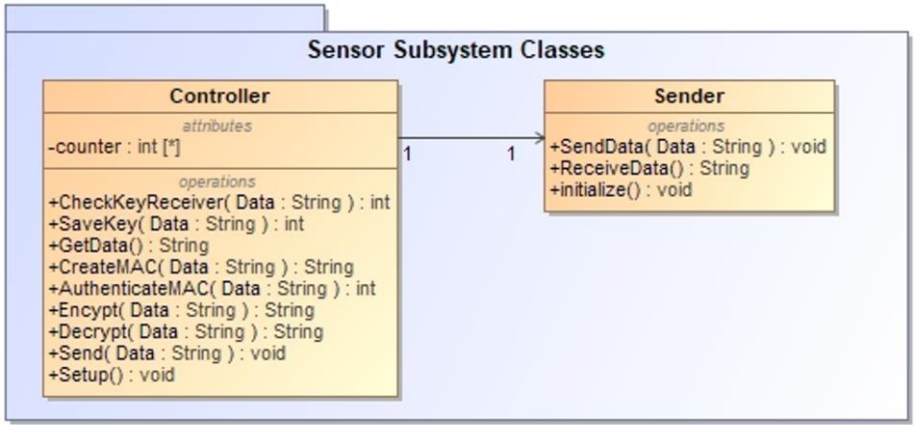

**13 pav.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo sistemos architektūra

## 3.7. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo panaudojimo programinė architektūra

Šiame poskyryje pateikiama programinė architektūra siūlomam sprendimui ir jo panaudojimui.

### 3.7.1. Jutiklių posistemės programinė architektūra

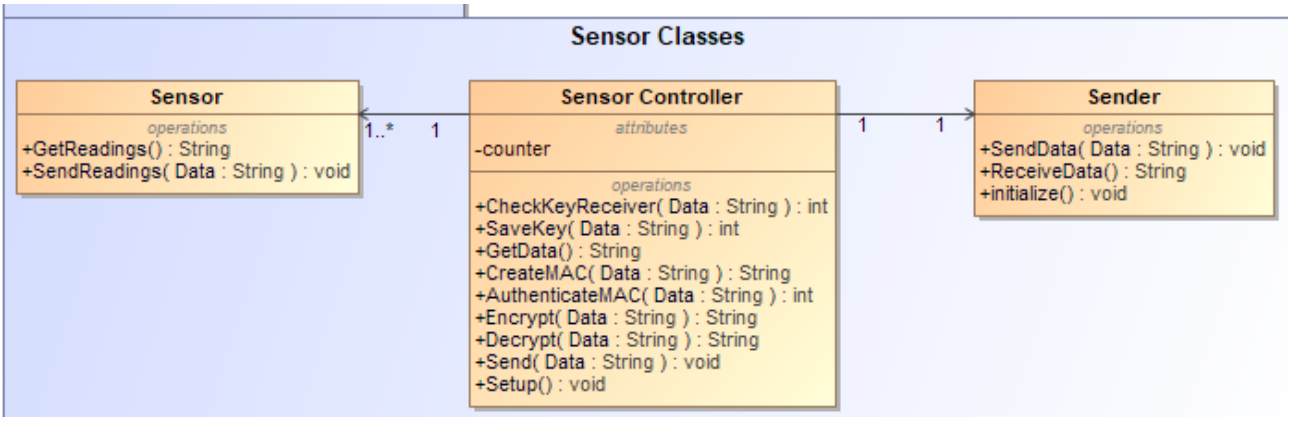
Jutiklių posistemės klasių diagrama pavaizduota 14 pav. „Controller“ klasė skirta posistemės valdikliui ir yra sudaryta iš pagrindinių funkcijų šifravimui, MAC reikšmių skaičiavimui, duomenų iš jutiklių valdiklių rinkimui ir apdorojimui. „Sender“ klasė skirta belaidės komunikacijos siųstuvui-imtuvui valdyti, sudaryta tik iš duomenų siuntimo ir gavimo funkcijų, šifravimas ar MAC reikšmių skaičiavimas šioje klasėje neatliekamas.




**14 pav.** Jutiklių posistemės klasių diagrama

### 3.7.2. Jutiklio programinė architektūra

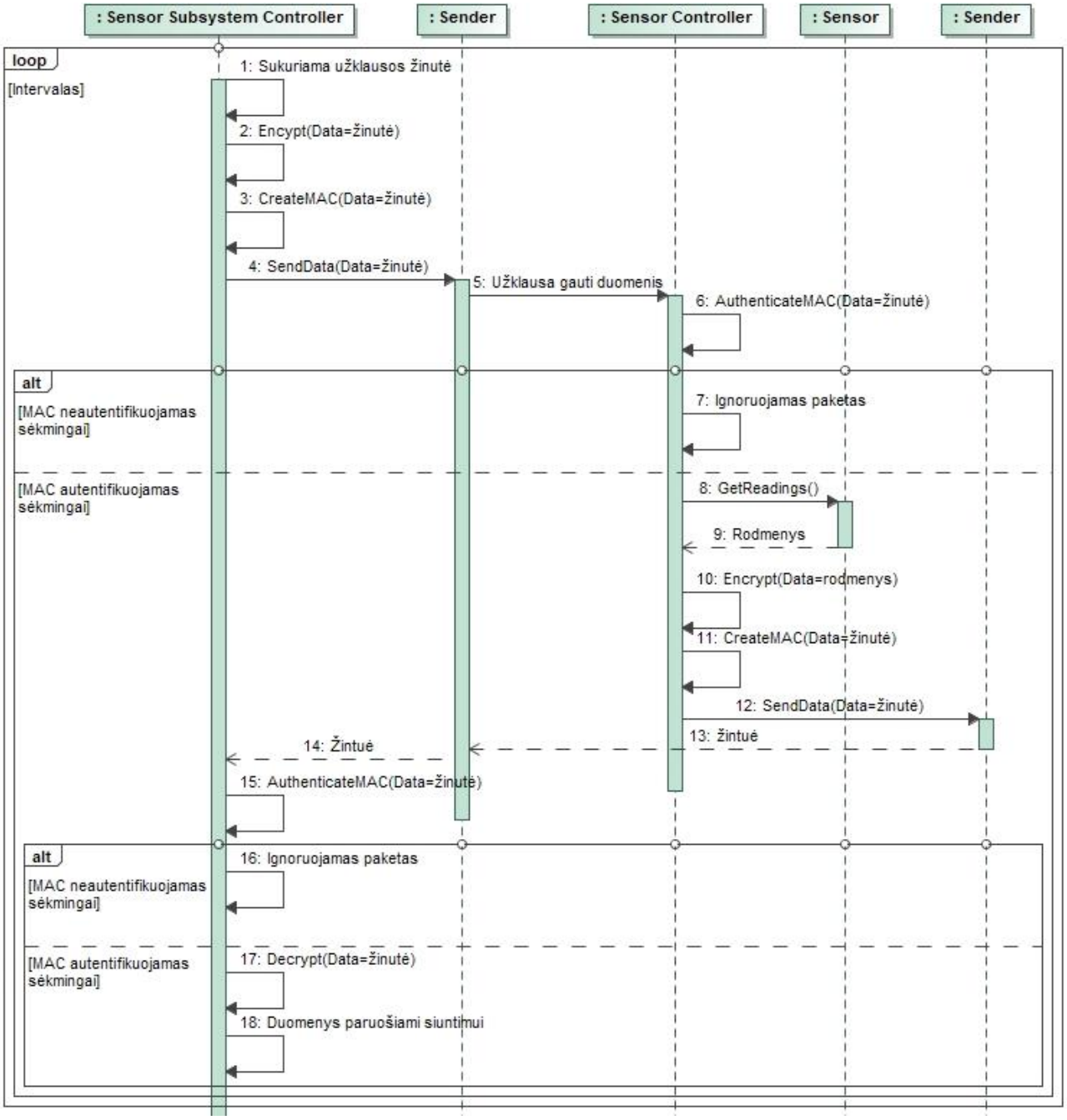
Jutiklių klasių diagrama pavaizduota **15 pav.** „Controller“ klasė skirta jutiklio valdikliui ir yra sudaryta iš pagrindinių funkcijų šifravimui, MAC reikšmių skaičiavimui, duomenų iš jutiklių rinkimui ir apdorojimui siųsti posistemės valdikliui. „Sender“ klasė skirta belaidės komunikacijos siųstuvui-imtuvui valdyti, sudaryta tik iš duomenų siuntimo ir gavimo funkcijų, šifravimas ar MAC reikšmių skaičiavimas šioje klasėje neatliekamas. „Sensor“ klasė yra skirta išgauti duomenis gautus iš jutiklių priskirtų jutiklio valdikliui. Ryšys daug su vienu nurodytas todėl, nes gali būti skirti keli tokie patys jutikliai matuoti vienai reikšmei skaičiuojant vidurkį.




**15 pav.** Jutiklio klasių diagrama

### 3.7.3. Jutiklio duomenų nuskaitymas

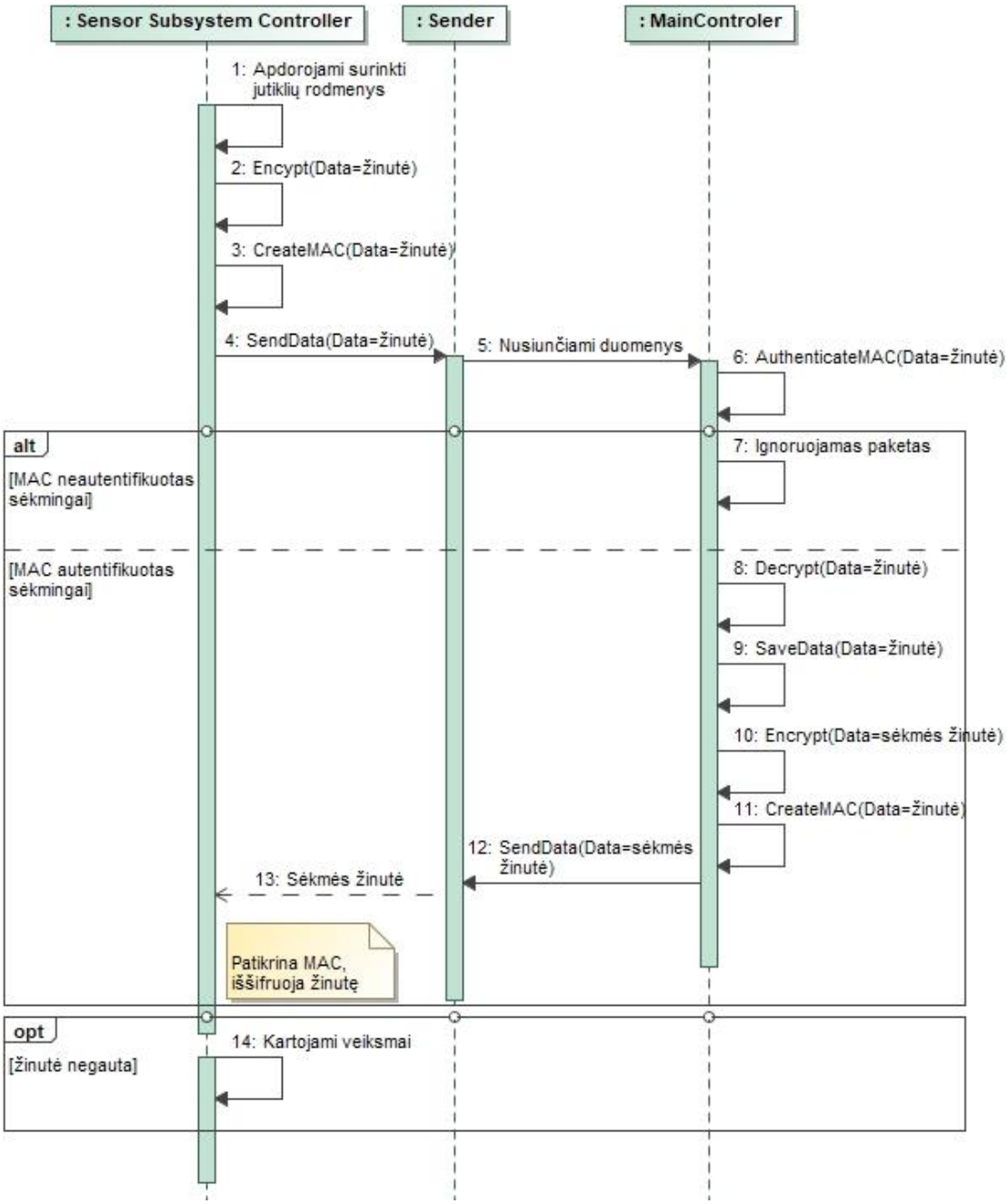
Jutiklių rodmenys yra nuskaitomi ciklu. Duomenų nuskaitymas yra aktyvuojamas jutiklių posistemės valdiklio. Valdiklis sukuria rodmenų užklausą, ją užšifruoja ir apskaičiuoja žinutei MAC reikšmę. Ši žinutė su MAC reikšme nusiunčiama jutiklio valdikliui, kuris patikrina MAC reikšmę. Jeigu MAC reikšmė nustatyta kaip neautentiška, žinutės paketas ignoruojamas ir atsakymas nesiunčiamas. Kitu atveju, žinutė iššifruojama ir nuskaitomi jutiklio duomenys. Šie duomenys užšifruojami ir žinutei apskaičiuojama MAC reikšmė. Ši žinutė ir MAC reikšmė nusiunčiama jutiklių posistemės valdikliui. Posistemės valdiklis turi patikrinti MAC reikšmę, jeigu ji nustatyta kaip neautentiška, paketas ignoruojamas. Kitu atveju, žinutė iššifruojama ir rodmenys paruošiami siuntimui pagrindiniam valdikliui. Veiksmų seka pavaizduota 16 pav. Apie duomenų siuntimą pagrindiniam valdikliui plačiau aprašyta 3.7.4 skyrelyje.




**16 pav.** Jutiklių duomenų nuskaitymas

### 3.7.4. Duomenų siuntimas pagrindiniam valdikliui

Duomenų siuntimas pagrindiniam valdikliui atliekamas surinkus posistemės jutiklių rodmenis. Šie duomenys apdorojami ir sukuriama nauja žinutė. Žinutė užšifruojama ir jai sukuriama MAC reikšmė. Ši žinutė nusiunčiama pagrindiniam valdikliui, kuris patikrina MAC reikšmę. Jeigu MAC reikšmė neautentifikuojama sėkmingai, paketas ignoruojamas. Kitu atveju, duomenys iššifruojami ir duomenys išsaugojami. Duomenis išsaugojus jutiklių posistemei nusiunčiama sėkmės žinutė. Jeigu posistemė žinutės negauna, skaitoma, kad įvyko klaida ir veiksmai kartojami iš naujo. Veiksmų seka pavaizduota 17 pav.




**17 pav.** Jutiklių posistemės duomenų siuntimas pagrindiniam valdikliui

### 3.8. Prototipo realizacijai naudojama techninė įranga

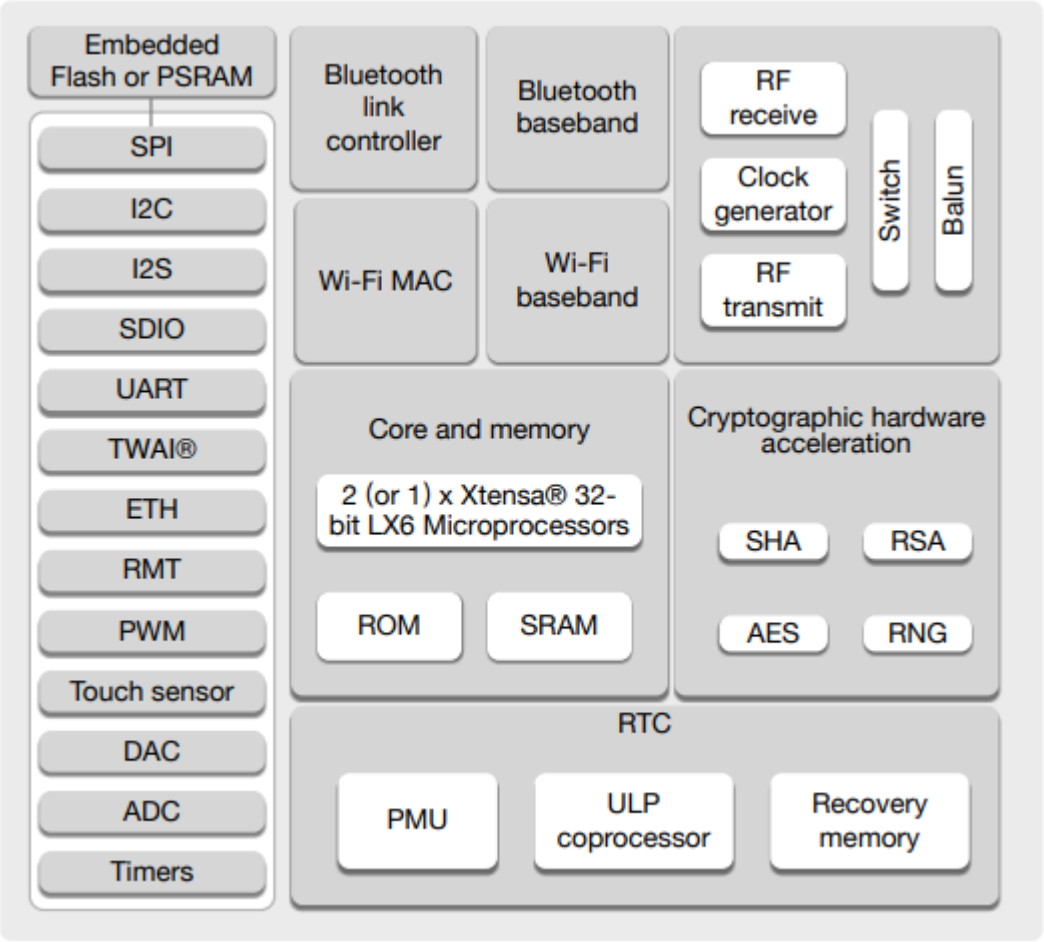
Šiame poskyryje aprašoma pasirinkta siūlomo sprendimo prototipo realizacijai techninė įranga ir jos specifikacijos.

#### 3.8.1. Prototipo realizacijai pasirinktas mikrovaldiklis

Kiekvienai sistemos posistemei pasirinkta naudoti „ESP32“ mikrovaldiklį. Šis valdiklis pasirinktas dėl plataus naudojimo, „ESP32“ valdiklių įvairumo ir techninių specifikacijų, kurių funkcinių blokų diagrama pavaizduota 18 pav. [22].

„ESP32“ pasirinktas prototipo realizacijai dėl šių techninės įrangos specifikacijų:

1. Kriptografinės aparatinės įrangos pagreitinimas AES, santraukų funkcijoms ir kitoms kriptografinėms funkcijoms.
2. „Bluetooth Classic“, „Low Energy“ valdiklis ir bazinė juosta – palaiko „Bluetooth v4.2“ versiją ir naujesnę, mažai energijos naudojančią, „BLE“ versiją. Taip pat, palaikoma daugiau negu vienas kliento prisijungimas ar serveris.
3. Dviejų branduolių 32-bit LX6 mikroprocesorius palaikantis 240 MHz spartą.
4. 520 KB SRAM, 448 KB ROM ir 16 KB RTC SRAM atminties.
5. 34 programuojami GPIO.
6. Palaikomi SPI, I2C, I2S, UART serijinės komunikacijos protokolai.




**18 pav.** „ESP32“ mikrovaldiklio funkcinių blokų diagrama [23]

Prototipo realizacijai naudojami tik „ESP32“ mikrovaldikliai kiekvienai posistemei įgyvendinti.

### 3.9. Prototipo realizacijai pasirinkta programinė įranga

Šiame poskyryje pateikiama prototipo realizacijai naudojama programinė įranga, programavimo aplinka, įrankiai ir bibliotekos.

#### 3.9.1. Prototipo realizacijai naudojama programavimo aplinka ir įrankiai

Programavimo aplinkai pasirinka sukonfigūruoti „Microsoft Visual Studio Code“ teksto redaktorių, nes tai yra atviro kodo teksto redaktorius su gausiu pasirinkimu įskiepių. Iš kurių pagrindinis prototipo realizacijai naudojamų įskiepių yra „Espressif IDF“.

Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodui įgyvendinti pasirinkta naudoti „C“ programavimo kalbą. Ši kalba pasirinkta, nes tai yra žemo lygio programavimo kalba suteikianti aukštą resursų valdymą, programos parašytos „C“ užima mažiau atminties ir naudoja mažiau išteklių.

„Espressif IDF“ įskiepis suteikia ir instaliuoja visas reikalingas bibliotekas ir visus įrankius į „Visual Studio Code“ teksto redaktorių: serijinio prievado programinę įrangą, palaikomų įrenginių konfigūracijas, „Buildroot“ įrankį skirtą programuoti, modifikuoti, išjungti ar įjungti įrenginio technines specifikacijas. Suteikia kompiliavimo, programos diegimo, stebėjimo ir kitus įrankius.

Kadangi projekte naudojama daug „Bluetooth“ sąsajų, kriptografinių bibliotekų ir kitų modulių, tai reikia įgalinti ir sukonfigūruoti valdiklio techninėje įrangoje ir „Espressif IDF“ aplinkoje, kas atliekama „Buildroot“ pagalba. Šis įrankis, teisingai sukonfigūravus, kompiliuoja ir naudoja tik specialiai įgalintus modulius, panaikinant nenaudojamų modulių ir bibliotekų užimamą atmintį ir skaičiavimo galią ribotų išteklių sistemose. Taip pat, įprastinėje įrenginių konfigūracijoje dauguma reikalingų modulių yra išjungti, dėl ko prototipo realizacijai „Buildroot“ įrankis yra būtinas.

Naudojant „Buildroot“ įgalinti šie moduliai ir sukonfigūruoti šie nustatymai:

1. Serijinės komunikacijos sparta nustatyta į reikšmę lygią „115200“ – naudojama duomenų spausdinimui kompiuteryje ir programinės įrangos diegimui įrenginyje.
2. Flash atmintis sukonfigūruota 2 MB dydžio.
3. Įgalinta tik „Bluetooth BLE“ versija, naudojanti mažiau energijos ir palaikanti kelias klientų sąsajas.
4. „Bluetooth“ jungčių skaičius nustatytas į 3. Kiekviena galima komunikacija padidina naudojamos atminties kiekį. Kadangi prototipe nuspręsta naudoti ne daugiau 3 komunikacijų valdiklyje, nuspręsta naudoti mažiau išteklių.
5. Įgalinta „Bluetooth BlueDroid“ implementacija, kuri yra paremta „Android Bluetooth“ implementacija suteikiančia galimybę lengvai skenuoti tinklą ieškant įrenginių, lengvesnį duomenų perdavimą tinkle ir kitas funkcijas. Svarbiausia „Bluedroid“ funkcija prototipo realizacijai yra GATT klientų ir GATT serverių (dar vadinama „Bluetooth Smart“) palaikymas. Kiekvienas GATT kliento ar serverio profilis suteikia galimybę sudaryti naują komunikaciją.
6. „ESP32“ valdiklio procesoriaus sparta nustatyta 240 MHz, kas padidina skaičiavimo galią.
7. Įgalinti „MBEDTLS“ bibliotekos moduliai: „CRYPTO“, „CMAC“, „AES“, „SHA“, „SHA512“, „MD5“, „HKDF“. Suteikiantys reikalingas bibliotekas realizuoti siūlomo kriptografinio sprendimo programinį kodą.
8. Įgalintas „ESP LOG“ modulis, suteikianti galimybę atspausdinti duomenis tvarkingu teksto formatavimu ir šešioliktainiu formatu. Kas suteikia patogumo testavimo ir rezultatų analizavimo

metu, nes analizuojami kriptografinių raktų ir duomenų šifravimų ir iššifravimų rezultatai yra baitų seka. Naudojant šešioliktainį formatą duomenys lengviau skaitomi.

### 3.9.2. Prototipo realizacijai naudojamos kodo bibliotekos

Pagrindinė biblioteka reikalingos siūlomo sprendimo realizacijai yra „MBEDTLS“, kuri suteikia visas reikalingas funkcijas:

1. „AES-CBC“ šifravimui naudojama „mbedtls/aes“ biblioteka.
2. „HKDF“ rakto išvedimui naudojama „mbedtls/hkdf“ ir „mbedtls/md“ bibliotekos.
3. „HMAC“ autentifikacijos kodui suskaičiuoti naudojama „mbedtls/md“ biblioteka.

„Bluetooth“ komunikacijai sudaryti naudojamos skirtingos „esp“ ir „freertos“ bibliotekos:

1. „esp_bt“ ir „esp_bt_main“ aprašytos pagrindinės funkcijos „Bluetooth“ valdiklio komandoms sudaryti.
2. „esp_gap_ble_api“ skirta tinklo skenavimui, įrenginių radimui ir topologijos valdymui.
3. "esp_gatts_api", "esp_gattc_api", „esp_gatt_common_api“ skirtos kliento ir serverio komunikacijai sudaryti, valdyti ir duomenų perdavimo valdymui.
4. „freertos/FreeRTOS“, „freertos/task“, „freertos/event_group“ skirtos įgalinti „Bluetooth“ žinučių įvykiams ir užduotims sudaryti, nestabdant viso įrenginio veiklos.

## 3.10. Prototipo realizacijos kodo struktūra

Prototipo realizacijos kodą sudaro 7 antraštės (header) failai ir 6 kodo failai.

Antraščių failų medis sudarytas iš tokių failų:

1. AES_CBC.h – antraštė skirta aprašyti globalias funkcijas šifravimui, iššifravimui ir globalioms apibrėžtoms vertėms, kurias gali naudoti kiti programos moduliai naudojantys šifravimą.
2. BLE_CLIENT.h – antraštė skirta aprašyti globalias funkcijas „Bluetooth“ komunikacijos klientui sudaryti ir valdyti.
3. BLE_SERVER.h – antraštė skirta aprašyti globalias funkcijas „Bluetooth“ komunikacijos serveriui sudaryti ir valdyti.
4. DEFINES.h – antraštė skirta aprašyti globalioms apibrėžtoms vertėms, kurios yra naudojamos visoje programoje.
5. HKDF.h – antraštė skirta aprašyti globalias funkcijas ir apibrėžtas vertes, kurios yra skirtos sesijos raktų išvedimui.
6. HMAC.h - antraštė skirta aprašyti globalias funkcijas ir apibrėžtas vertes, kurios yra skirtos žinučių autentifikavimo vertėms apskaičiuoti.
7. Main.h – antraštė skirta pagrindiniam programos paleidimui reikalingų verčių apibrėžimui.
8. TIMER.h – antraštė skirta aprašyti globalias funkcijas ir apibrėžtas vertes, kurios yra skirtos laiko įvykiams apibrėžti ir valdyti.

Kodo failų medis sudaryti iš tokių failų:

1. AES_CBC.c – aprašomos funkcijos šifravimui, iššifravimui ir šifruojamų žinučių blokų patikrinimui, ilgio skaičiavimams ir kamšalo (padding) užpildymo blokuose realizavimui.
2. BLE_CLIENT.c – aprašomos funkcijos sudaryti ir valdyti „Bluetooth“ kliento komunikaciją, ieškoti ir prisijungti prie nurodytų „Bluetooth“ serverių, siųsti užklausas serveriams ir gauti atsakymus iš serverių.

3. BLE_SERVER.c – aprašomos funkcijos sudaryti ir valdyti „Bluetooth“ serverio komunikaciją, leisti „Bluetooth“ klientui prisijungi, gauti ir apdoroti užklausas, išsiųsti atsakymus.
4. HKDF.c – aprašomos funkcijos reikalingos sesijos raktų išvedimui ir apdirbimui.
5. HMAC.c – aprašomos funkcijos žinučių autentifikavimo vertėms apskaičiuoti.
6. Main.c – skirta paleisti visus reikalingus modulius, „Bluetooth“ komunikaciją ir laikmatį.
7. TIMER.c – aprašomos funkcijos sukonfigūruoti laikmatį, sudaryti įvykius, juos užregistruoti ir paleisti laikmatyje. Šios funkcijos periodiškai paleidžia jutiklių duomenų rinkimo užklausas ir sesijos raktų generavimą ir siuntimą.
8. Main.c – paleidžiamas programos veikimas.

Taip pat, failų medyje priklauso du atskiri failai:
1. CMakeList.txt – failas skirtas aprašyti kompiliavimo taisykles failams nurodant, kurie failai įeiną į programos kompiliavimą.
2. Sdkconfig – „Buildroot“ įrankio naudojamas failas, kuris skirtas įrenginio techninei įrangai ir bibliotekų sąrašui sukonfigūruoti. Prototipo realizacijos metu naudojamas nustatyti procesoriaus spartai, atminties kiekiui, „BLE v4.2“ įgalinimui, kelių klientų ir serverių įgalinimui.

### 3.11. Prototipo veikimas

Šiame poskyryje aprašomas pagrindinio valdiklio ir jutiklių posistemių veikimas realizuotame prototipe.

#### 3.11.1. Pagrindinio valdiklio prototipo realizacijos veikimas

Įjungus įrenginį pasileidžiama „main.c“ sukompiliuota programa, kuri inicializuoja „Bluetooth“ kliento nustatymus, paleidžiant „BLE_CLIENT.c“ funkcijas. Įjungiamas „Bluetooth“ kontroleris, inicializuojamas „Bluedroid“ funkcionalumas palaikantis GATT profilių komunikaciją, užregistruojamos funkcijos apdorojančios komunikacijos metu siunčiamas ir gaunamas žinutes. Tai atlikus yra paleidžiamas „Bluetooth“ tinklo skenavimas, ieškant nurodytų įrenginių. Prototipo realizacijoje ieškoma įrenginių su pavadinimais: „BLUE“, „BLACK“, „ALONE“. Šie įrenginiai turi būti „Bluetooth“ serveriai. Radus reikiamą įrenginį pradedama prisijungti prie rasto serverio. Tai atlikus kliento įrenginys serveriui nusiunčia informaciją, nurodančią, kokio tipo užklausos bus siunčiamos serveriui. Šie veiksmai atliekami kol pasiekiamas prisijungusių įrenginių skaičius, kuris realizacijoje nustatytas į 3 įrenginius. Kai pagrindinis valdiklis paleidžia „Bluetooth“ komunikaciją, ji pradeda dirbti lygiagrečiai programos, todėl „main.c“ sėkmingai tęsia darbą ir paleidžia laikmačio programą, aprašytą „TIMER.c“ faile. Laikmatis nepradeda darbo tol, kol pagrindinis valdiklis prisijungia prie visų trijų „Bluetooth“ serverių įrenginių. Kadangi „Bluetooth“ komunikacija veikia lygiagrečiai, likusios programos, įrenginiams praradus ryšį ir bandant jį atstatyti, ar siunčiant didesnes žinutes, nėra stabdomos.

Prisijungus prie visų „Bluetooth“ serverių įrenginių, laikmačio programa paleidžią raktų generavimą ir išsiuntimą prisijungusiems serverių įrenginiams. Raktų generavimas yra atliekamas kiekvienam prijungtam serverio įrenginiui atskirai. Pradžioje paleidžiamos „HKDF.c“ generavimo funkcijos, kurios sugeneruoja kriptografinį sesijos raktą. Šis raktas yra išsaugomas ir perduodamas „AES_CBC.c“ šifravimo funkcijoms, kurios suskaičiuoja sesijos rakto šifrogramą naudojant nekintantį, įrenginyje išsaugotą pagrindinį raktą, kuris tinkle yra nežinomas, bet saugomas kiekvieno įrenginio. Šiai šifrogramai yra paleidžiamos „HMAC.c“ funkcijos, suskaičiuojančios užšifruotam kriptografiniam raktui skirtą vientisumą ir autentiškumą patvirtinančią reikšmę. Kai tai atliekama,

sudaroma nauja žinutė, kurioje yra pateikiama apskaičiuota HMAC reikšmė - 32 baitai, žinutės tipas 1 – baitais, įrenginio ID tinkle – 1 baitas, kriptografinio sesijos rakto šifrograma – 16 baitų. Taip sudaromas 50 baitų paketas, kuris nusiunčiamas „Bluetooth“ serverio įrenginiui.

Kai žinutė yra nusiunčiama „Bluetooth“ serveriui, laukiama atsakymo žinutės. Atsakymo žinutė yra patikrinama. Iš gautos žinutės yra paimama HMAC ilgio dalis ir šifrogramos ilgio dalis. Šifrogramai pagrindinis valdiklis suskaičiuoja naują HMAC reikšmę su prieš tai sugeneruotu ir saugomu komunikacijos tunelio kriptografiniu sesijos raktu. Naujai sugeneruota HMAC reikšmė yra palyginama su atsakymo žinutėje pateikta HMAC reikšme. Jeigu reikšmės yra lygios, pagrindinis valdiklis tęsia su likusiais įrenginiais tol, kol kiekvienam prijungtam „Bluetooth“ serveriui yra sėkmingai sugeneruojamas ir nusiunčiamas kriptografinis sesijos raktas. Programos darbo rezultatai pavaizduoti 19 pav.

Atėjus laikui pakeisti kriptografinius sesijos raktus, atliekami tokie patys veiksmai, tačiau kol dirbama su vienu įrenginiu, kiti įrenginiai tęsia komunikaciją su senais kriptografiniais raktais tol, kol pagrindinis valdiklis nepradeda darbo su jais.

Kai pagrindinis valdiklis paskirsto pirmuosius kriptografinius sesijos raktus, yra sukuriamas vienas laikmatis su dvejomis grupėmis:

1. „TIMER_GROUP_0“ – laikmačio grupė skirta sekti laiką kada paleisti periodinį kriptografinių sesijos raktų generavimą.
2. „TIMER_GROUP_1“ – laikmačio grupė skirta sekti laiką kada reikia sugeneruoti užklausą jutiklių duomenų rinkimui.

Kai laikmatis pasiekia „TIMER_GROUP_1“ nurodyta laiko periodą, pradedama generuoti užklausas „Bluetooth“ serverio įrenginiams atsiųsti duomenis pagrindiniams valdikliui. Pradžioje patikrinama ar šiuo metu su įrenginiu nėra keičiamasi kriptografiniais sesijos raktais. Jeigu įrenginys yra laisvas, suskaičiuojama teksto „data“ šifrograma „AES_CBC“ šifru naudojant komunikacijai su „Bluetooth“ serverio įrenginiu saugomą kriptografinį sesijos raktą. Šiai šifrogramai yra apskaičiuojamas HMAC reikšmė ir sudaroma nauja žinutė taip pat kaip siunčiant kriptografinį raktą. Tuomet yra laukiama atsakymo. Kai atsakymo žinutė yra gaunama iš „Bluetooth“ serverio įrenginio, taip pat, kaip raktų apsikeitimo atsakymo žinutėje, yra patikrinama HMAC reikšmė. Jeigu HMAC reikšmės sutampa, atsakymo žinutėje pateikti šifruoti duomenys yra iššifruojami. Programos rezultatai pavaizduoti 20 pav. ir 21 pav. Kaip pavaizduota paveikslėlyje, jeigu šifruojamas tekstas pilnai neužpildo viso „AES_CBC“ bloko ilgio, yra užpildomas kamšalu „00“ kuris yra matomas iššifravus tekstogramą pažymėtą geltonai 21 pav.

„TIMER_GROUP_0“ laikmačio įvykis skirtas periodiniam kriptografinių sesijos raktų generavimui pavaizduotas 22 pav.

Pagrindinio valdiklio laikmatis pabaigos neturi ir veikia tol kol pagrindinis valdiklis yra išjungiamas.

### 3.11.2. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo prototipo eksperimento scenarijus

Tam, kad tinkamai atlikti eksperimentą reikia įjungti keturis įrenginius: vieną pagrindinį valdiklį, kuris yra „Bluetooth“ klientas ir 3 tinklo įrenginius, kurie yra „Bluetooth“ serveriai. Kai kliento įrenginys yra įjungiamas, yra laukiama kol bus susijungta su kitais trimis serverio įrenginiais, darbas tęsiamas tik kai sėkmingai susijungiama prie visų įrenginių. Kai susijungimas su visai įrenginiais atliekamas sėkmingai, kliento įrenginys atlieka kriptografinių sesijos raktų generavimą ir apsikeitimą:

1. Suskaičiuojamas naujas kriptografinis raktas iš pagrindinio rakto naudojant HKDF rakto išvedimo funkciją.
2. Kriptografinis raktas yra užšifruojamas „AES-CBC“ algoritmu.
3. Užšifruotam kriptografiniam raktui suskaičiuojama žinutės autentifikacijos reikšmė (MAC) naudojant HMAC funkciją.
4. Sudaroma nauja žinutė, kurios pradžioje įrašyta žinutės autentifikacijos reikšmė (MAC), toliau žinutės kodas, įrenginio, kuriam žinutė siunčiama kodas ir užšifruotas kriptografinis raktas.
5. Žinutė nusiunčiama tinklo įrenginiui, kuriam yra skirta.
6. Laukiama atsakymo iš tinklo įrenginio, kurį gavus patikrinama žinutėje esanti autentifikacijos reikšmė. Jeigu ši reikšmė sėkmingai patikrinama, programa užskaito, kad kriptografinis raktas nusiųstas sėkmingai.
7. Veiksmai 1-5 kartojami kol kriptografiniai raktai yra nusiunčiami visiems tinklo įrenginiams, prie kurių yra prisijungta. Programa nelaukia kol bus gaunama žinutė pažymėta veiksmo numeriu 6, o iškarto pradeda darbus iš naujo generuojant kriptografinį raktą kitam tinklo įrenginiui. Veiksmas numeriu 6 dirba asinchroniškai.
8. Kai visi kriptografiniai raktai yra paskirstyti, sukuriama duomenų užklausos žinutė, kuri yra užšifruojama „AES-CBC“ algoritmu.
9. Šiai žinutei suskaičiuojama autentifikacijos reikšmė (MAC).
10. Sudaroma nauja žinutė, kurios pradžioje įrašyta žinutės autentifikacijos reikšmė (MAC), toliau žinutės kodas, įrenginio, kuriam žinutė siunčiama kodas ir užšifruotas duomenų užklausos tekstas.
11. Žinutė nusiunčiama tinklo įrenginiui, kuriam yra skirta.
12. Laukiama atsakymo iš tinklo įrenginio, kurį gavus patikrinama žinutėje esanti autentifikacijos reikšmė. Jeigu ši reikšmė sėkmingai patikrinama, programa užskaito, kad duomenys iš serverio gauti sėkmingai.
13. Veiksmai 8-11 kartojami kol duomenų užklausos žinutės nusiunčiamos visiems tinklo įrenginiams, prie kurių yra prisijungta. Programa nelaukia kol bus gaunama žinutė pažymėta veiksmo numeriu 12, o iškarto pradeda darbus iš naujo kuriant duomenų užklausą kitam tinklo įrenginiui. Veiksmas numeriu 12 dirba asinchroniškai.

14. Po vienos minutės kartojami veiksmai nuo numeriu 1 pažymėto veiksmo.

Visi „Bluetooth“ tinklo serverių įrenginiai dirba viena programa. Pati programa neinicializuoja jokių veiksmų ir tik reaguoja į žinutes gautas iš kliento įrenginio:

1. Gavus žinutę yra patikrinamas žinutės kodas nurodantis kokio tipo žinutė gauta.
2. Patikrinama žinutės autentifikacijos reikšmė (MAC), jeigu ši reikšmė neautentifikuota sėkmingai, žinutė ištrinama. Jeigu žinutė autentifikuota, darbai atliekami pagal žinutės kodą.
3. Jeigu žinutė yra kriptografinio rakto apsikeitimo žinutė:
    a. Iššifruojamas kriptografinis raktas naudojant „AES-CBC“ algoritmą ir pagrindinį sistemos kriptografinį raktą.
    b. Kriptografinis raktas išsaugojamas.
    c. Sugeneruojamas tekstas su sėkmės reikšme.
4. Jeigu žinutė yra duomenų užklausa:
    a. Iššifruojama duomenų užklausa naudojant „AES-CBC“ algoritmą ir sesijos kriptografinį raktą.
    b. Sugeneruojamas tekstas su atsitiktiniu skaičiumi ir matavimo vienetu.
5. Tekstas užšifruojamas „AES-CBC“ algoritmu naudojant sesijos raktą.
6. Užšifruotam tekstui suskaičiuojama autentifikacijos reikšmė (MAC).
7. Sudaroma nauja žinutė su užšifruotu tekstu, kurios pradžioje yra žinutės autentifikacijos reikšmė, toliau žinutės atsakymo kodas, įrenginio, kuris siunčia atsakymą kodas ir užšifruotas tekstas.

### 3.11.3. Jutiklių posistemės valdiklio prototipo realizacijos veikimas

Įjungus įrenginį pasileidžia „main.c“ sukompiliuota programa, kuri inicializuoja „Bluetooth“ serverio nustatymus, paleidžiant „BLE_SERVER.c“ funkcijas. Įjungiamas „Bluetooth“ kontroleris, inicializuojamas „Bluedroid“ funkcionalumas palaikantis GATT profilių komunikaciją, užregistruojamos funkcijos apdorojančios komunikacijos metu siunčiamas ir gaunamas žinutes. „Bluetooth“ serverio įrenginyje „Bluetooth“ tinklo skenavimas nėra paleidžiamas, tačiau įrenginiui yra sukonfigūruojama informaciją, kuri turi būti matomą skenuojančių įrenginių. Taip paleidžiamas „Bluetooth“ skelbimas (advertising), kurio metu įrenginys tampa prieinamas „Bluetooth“ tinkle. Tuomet įrenginys laukia kol bus atsiunčiamas kriptografinis sesijos raktas. Kai raktas atsiunčiamas, programa gali tęsti darbą.

Jutiklio posistemės valdikliui gavus kriptografinių sesijos raktų apsikeitimui skirtą žinutę, pirma yra patikrinama žinutės HMAC reikšmė, suskaičiavus naują HMAC reikšmę atsiųstai šifrogramai naudojant pagrindinį kriptografinį raktą įdiegtą įrenginyje. Jeigu HMAC reikšmės atitinka, šifrograma yra iššifruojama „AES_CBC“ šifru naudojant pagrindinį kriptografinį raktą ir naujas

gautas kriptografinis sesijos raktas yra išsaugojamas. Tuomet suskaičiuojama nauja šifrograma tekstui „success“, kuris yra užšifruojamas naudojantis „AES_CBC“ šifrą su naujai gautu kriptografiniu sesijos raktu. Šiai šifrogramai apskaičiuojama HMAC reikšmė ir sudaromas naujas paketas. Šis paketas nusiunčiamas pagrindiniams valdikliui kaip atsakymo žinutė.

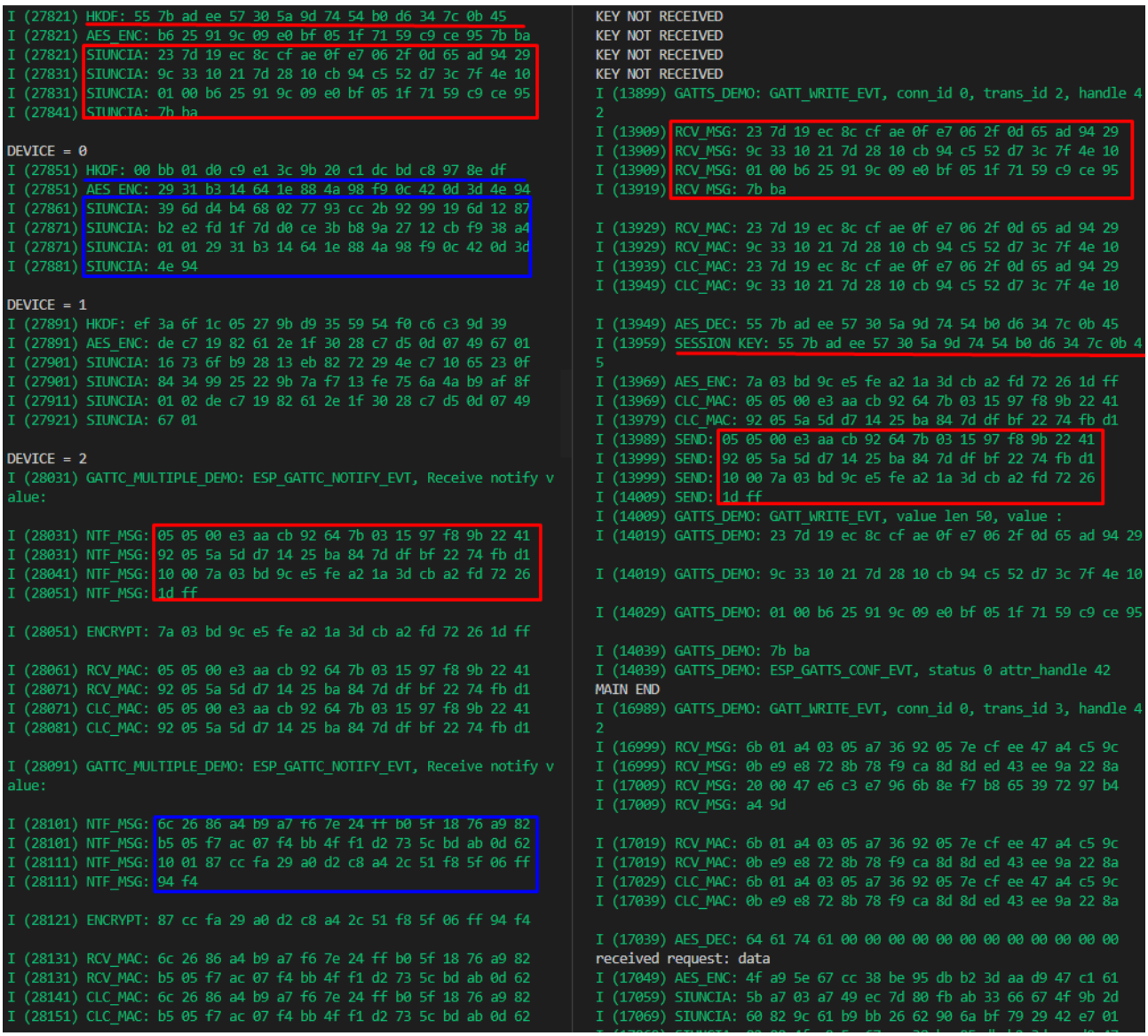
Jutiklio posistemės valdikliui gavus duomenų užklausos žinutę iš pagrindinio valdiklio, taip pat, kaip raktų žinutėje yra patikrinama HMAC reikšmė. Jeigu HMAC reikšmė atitinka, sugeneruojamas atsitiktinis skaičius kaip jutiklio imitacinė reikšmė. Ši reikšmė užšifruojama su „AES_CBC“ šifru naudojant kriptografinį sesijos raktą. Šifrogramai apskaičiuojama nauja HMAC reikšmė ir sudaromas naujas paketas taip pat, kaip raktų apsikeitimo žinutės atsakyme. Atsakymo žinutė nusiunčiama atgal pagrindiniam valdikliui.

Jutiklių posistemės valdiklis prototipo realizacijoje neatlieką kitų veiksmų savaime. Visi darbai yra inicializuojami pagrindinio valdiklio per „Bluetooth“ žinutes.

### 3.11.4. Realizuotos sistemos duomenys tyrimui

Kaip pavaizduota 19 pav. pirmajame stulpelyje, HKDF žyma yra atspausdintas suskaičiuotas kriptografinis sesijos raktas, kuris toliau yra užšifruojamas. Šifravimo rezultatas yra atspausdinamas „AES_ENC“ žyma. Užšifruotas kriptografinis sesijos raktas turi būti nusiųstas, todėl sudaromas paketas, kuris atspausdinamas „SIUNCIA“ žyma. Pirmosios dvi eilutės susideda iš suskaičiuotos autentifikavimo reikšmės, trečioje eilutėje pirmieji du skaičiai yra žinutės tipo kodas ir įrenginio, kuriam žinutė priklauso kodas. Po šio kodo reikšmių, duomenų sraute pateikiamas užšifruotas kriptografinis sesijos raktas. Įrenginio kodu „00“ kriptografinio sesijos rakto skaičiavimo ir siuntimo duomenys pažymėti raudona spalva. Toliau prieduose pateiktame 51 pav., mėlyna spalva pažymėti tokie patys veiksmai įrenginiui kodu „01“.

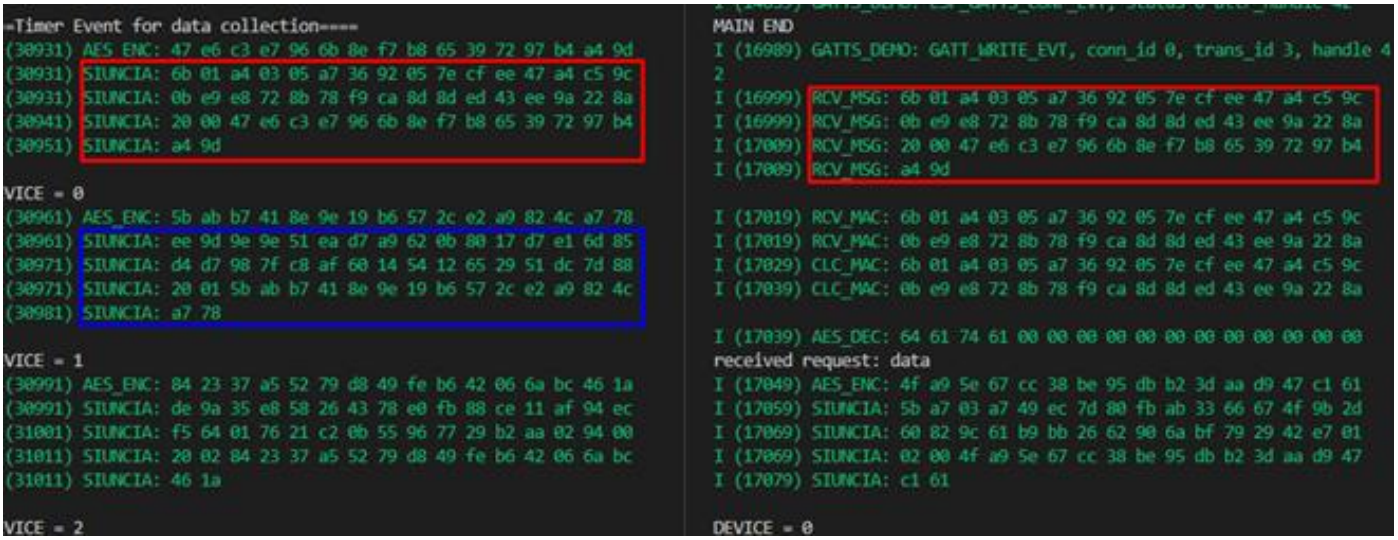
Antrajame stulpelyje 19 pav. pirmojo įrenginio kodu „00“ gauta žinutė su užšifruotu kriptografiniu sesijos raktu pažymėta žyma „RCV_MSG“ ir pažymėta raudona spalva. Palyginus duomenis su pirmajame stulpelyje atvaizduotais siųstais duomenimis matyti, kad žinutė sutampa. Toliau, antrame stulpelyje yra palyginama autentifikacijos reikšmė. Pirmuoju žingsniu autentifikacijos žinutė yra paimama iš gautos žinutės ir terminale atspausdinta žyme „RCV_MAC“. Programai suskaičiavus palyginamąją autentifikacijos reikšmę, kuri suskaičiuota naudojant užšifruotą kriptografinį raktą iš gautos žinutės, terminale ši reikšmė yra atspausdinama žyme „CLC_MAC“. Jeigu šios reikšmės sutampa, programa iššifruoja kriptografinį sesijos raktą ir rezultatą atspausdina žyme „AES_DEC“ ir „SESSION_KEY“, kai kriptografinis sesijos raktas yra sėkmingai išsaugomas. Toliau, yra sukuriama sėkmės žinutė, ji užšifruojama ir rezultatas yra atspausdinamas žyme „AES_ENC“. Šiai užšifruotai žinutei yra suskaičiuojama autentifikacijos reikšmė atspausdinama žyme „CLC_MAC“. Programai sudarius duomenų žinutę ji yra atspausdinama žyme „SEND“. Žinutėje pirmos dvi eilutės susideda iš autentifikacijos reikšmės, tuomet įrašomas žinutės kodas ir įrenginio, kuris žinutę siunčia, kodas „00“. Ši žinutė yra gaunama ir atspausdinama 19 pav. pirmajame stulpelyje naudojant žymą „NTF_MSG“. Toliau, programa atlieka tokius pačius veiksmus patikrinti žinutės autentiškumui.




**19 pav.** Pagrindinio valdiklio raktų generavimas ir paskirstymas pirmam jutiklių posistemės valdikliui

Prieduose pateikto paveikslėlio 51 pav. antrajame stulpelyje yra atliekami tokie patys veiksmai kaip 19 pav. antrajame stulpelyje, tik kito įrenginio, su kodu „01“. Šio įrenginio rezultatai pažymėti mėlyna spalva.

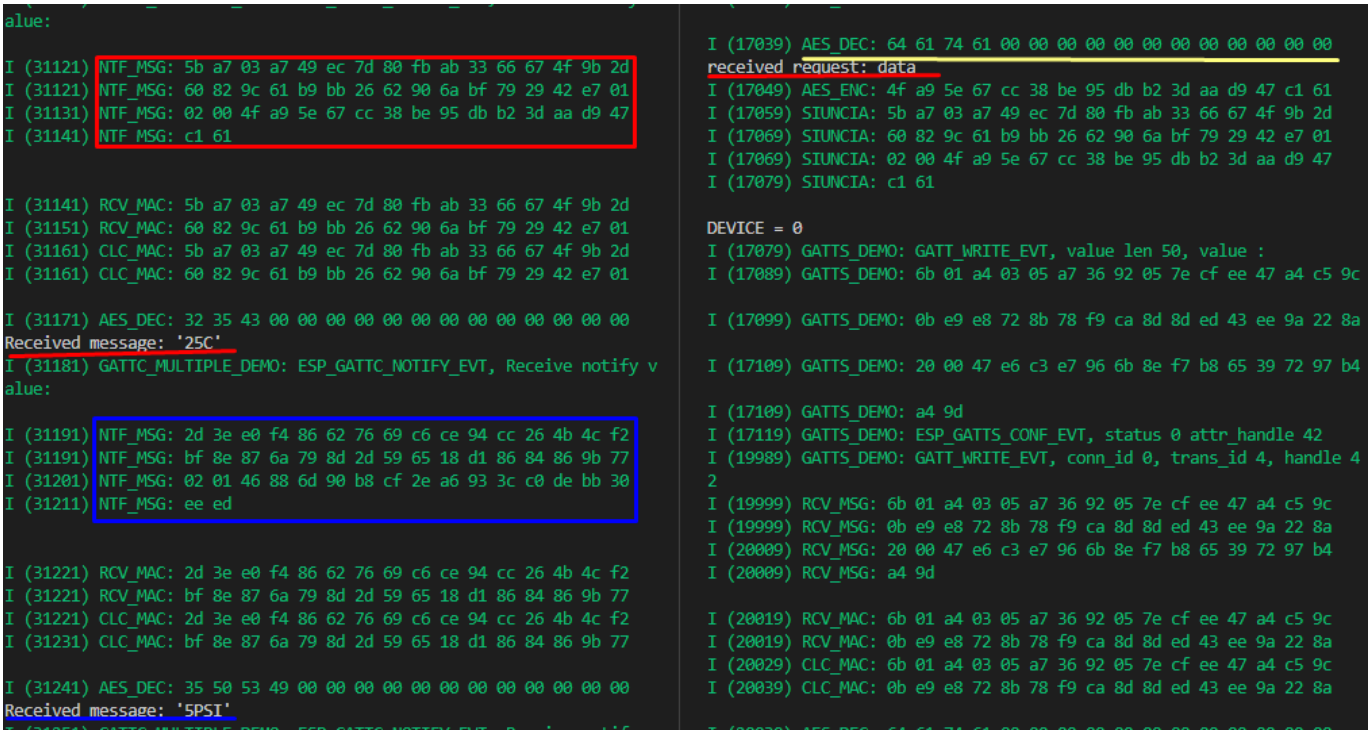
Duomenų užklausa yra pavaizduota 20 pav. Pirmajame stulpelyje pavaizduotas duomenų užklausų siuntimas visiems įrenginiams tinkle. Raudona spalva pažymėta žinutė įrenginiui kodu „00“. Eilutėje pažymėtoje žyma „AES_ENC“ yra atspausdintas teksto „data“ šifravimo rezultatas, toliau žyma „SIUNCIA“ atspausdina visą žinutę su suskaičiuota autentifikacijos reikšme, žinutės kodu, įrenginio, kuriam žinutė skirta kodu ir užšifruotu tekstu. Antrajame paveikslėlio stulpelyje žyma „RCV_MSG“ atspausdinama įrenginio kodu „00“ gauta žinutė. Šioje žinutėje pateikta autentifikacijos reikšmė „RCV_MAC“ yra palyginama su naujai programos suskaičiuota šifruoto teksto autentifikacijos reikšme „CLC_MAC“. Jeigu šios reikšmės sutampa, šifruotas žinutės tekstas yra iššifruojamas ir rezultatas atspausdinamas žyme „AES_DEC“, terminale pateikiamas rezultatas ASCII kodu po žymės „received request:“. Tuomet yra užšifruojami pavyzdiniai jutiklio duomenys, suskaičiuojama autentifikacijos reikšmė ir sudaroma žinutė atspausdinta žyme „SIUNCIA“.




**20 pav.** Pagrindinio valdiklio duomenų užklausos pirmam jutiklių posistemės valdikliui

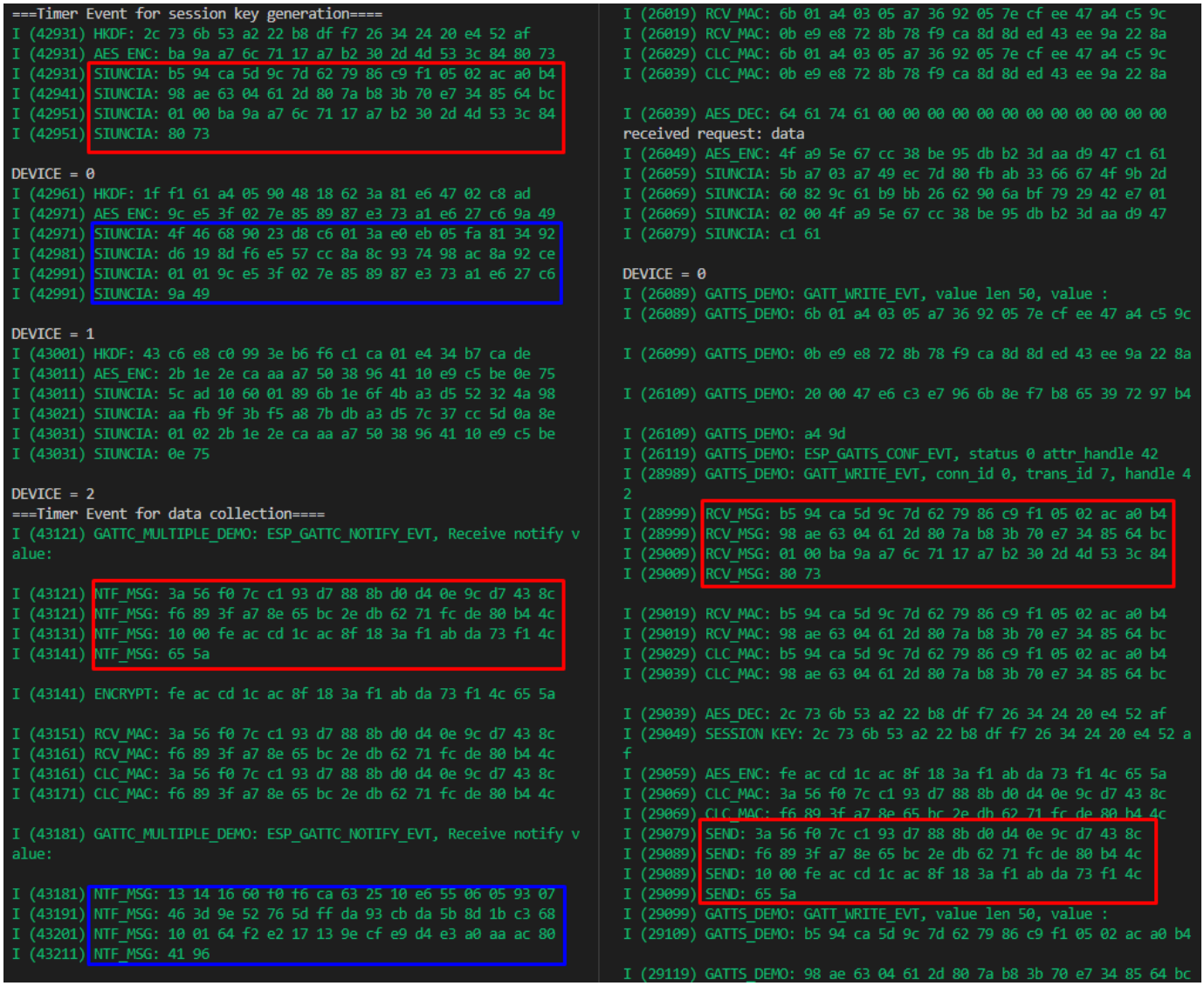
Antrajame 52 pav. stulpelyje atliekami tokie patys veiksmai įrenginio kodu „01“.

Šių įrenginių siunčiamų duomenų gavimas pavaizduotas 21 pav. pirmajame stulpelyje. Raudona spalva pažymėti duomenys gauti iš įrenginio kodu „00“, 53 pav. mėlyna spalva pažymėti duomenys gauti iš įrenginio kodu „01“. Gautos žinutės atspausdintos žyme „NTF_MSG“. Šioms žinutėms patikrinama autentifikacijos reikšmės pažymėtos „RCV_MAC“ ir „CLC_MAC“. Toliau, duomenys iššifruojami ir atspausdinami žymėmis „AES_DEC“ ir „received message:“.




**21 pav.** Pagrindinio valdiklio duomenų užklausų atsakymo žinutė iš pirmojo posistemių valdiklio

Praėjus nustatytam laiko periodui yra atliekamas naujas kriptografinių sesijos raktų skaičiavimas ir paskirstymas, kuris pavaizduotas 22 pav. ir 54 pav. Visi atliekami veiksmai ir atspausdinami rezultatai atitinka 19 pav. ir 51 pav. pavaizduotus veiksmus ir žymes, tik su naujais duomenimis.




**22 pav.** Pagrindinio valdiklio periodinis kriptografinių sesijos raktų generavimo laiko įvykis – pirmo įrenginio duomenys

Visi duomenys terminale turi ir duomenų atspausdinimo laiką nuo įrenginio įjungimo. Šie duomenys yra matuojami milli-sekundėmis ir atspausdinami eilutės pradžioje skliausteliuose.

### 3.12. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo prototipo realizacijos išvados

1. Projekto realizacijos metu pastebėta, kad sėkmingai sukurti „Bluetooth“ kelių įrenginių komunikaciją reikalavo daugiausiai laiko ir programos kodo eilučių. Taip pat, „Bluetooth“ komunikacijai sudaryti tarp kelių įrenginių reikėjo įgyvendinti GATT profilius dėl ko programos kompleksiškumas padidėjo. Realizacijos įgyvendinimo metu detaliai analizuotas „Bluetooth“ GATT profilių veikimas, konfigūracija, žinučių siuntimas, gavimas ir atsakymo žinučių siuntimas ir gavimas. Rezultate, realizuota įvykiais pagrįsta „Bluetooth“ komunikacija nestabdanti sistemos ir programos veikimo.

2. Dėl detalios „ESP“ mikrovaldiklių sistemos ir jiems skirtos programavimo aplinkos dokumentacijos, projekto prototipo realizacijos saugumo programos sukurtos naudojant „espressif“ įrankio vidines bibliotekas, kadangi didžioji dalis kriptografinių algoritmų yra įdiegta į „mbedtls“ biblioteką. Pastebėta, kad sudėtingiausia dalis iš saugaus metodo realizacijos yra saugus rakto apsikeitimas tarp įrenginių ir saugus žinučių apsikeitimas tarp įrenginių. Tam, kad tinkamai realizuoti, projekto realizacijoje, kiekvienai žinutei sukuriamas naujas paketas su visa saugumo informacija ir užšifruotais duomenimis. Šis paketas inkapsuliavus „Bluetooth“ komunikacijos paketu, sėkmingai nusiunčiamas kitam tinklo įrenginiui.

3. Kadangi pasirinkta techninė įranga yra daugiau nei pakankamai stipri kompiuterinio skaičiavimo galios atžvilgiu, visos kriptografinės funkcijos vykdomos itin sparčiai ir sistemos vėlavimų ar stabdymų nepastebėta. Taip pat, pastebėta, kad kiekvienoje žinutėje nusiunčiama HMAC reikšmė visada gaunama neiškraipyta komunikacijos tunelyje ir visada yra teisinga patikrinus įrenginio, gavusio kito tinklo įrenginio žinutę. Dėl šios priežasties visos žinutės yra sėkmingai apdorojamos. Todėl nuspręsta, kad projekto prototipo realizacija yra sėkminga ir paruošta siūlomo sprendimo tyrimui.

# 4. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo tyrimas

Šiame skyriuje aprašomi tyrimo metu analizuojami siūlomo sprendimo parametrai ir tyrimo metodai. Tyrimo tikslas – išanalizuoti projekto metu siūlomą sprendimą ir nuspręsti ar transporto priemonių lokalios belaidžio tinklo saugos metodas yra pranašesnis už rinkoje esančius metodus tyrimo metu analizuojamais parametrais.

Kadangi projekto metu pasiūlytas sprendimas yra skirtas transporto priemonių lokaliam belaidžiam tinklui, tyrimo metu bus tiriamas metodų veikimas įterptinėje sistemoje ir realizuojant metodus mikrovaldikliuose.

## 4.1. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo tyrimo parametrai

Ištirti ar siūlomas sprendimas yra pranašesnis už kitus rinkoje esančius metodus yra tiriami šie parametrai:

1. Energijos efektyvumas – tiriama kiek kiekvienas metodas naudoja energijos matuojant mili-amperais (mA) ir konvertuojant sąnaudos į vatus (W). Matavimai atliekami stebint kiek metodas naudoja energijos atliekant nurodytus veiksmus ir viso įrenginio veikimo metu esančias energijos sąnaudas. Todėl, pradžioje patikrinama kiek įrenginys naudoja energijos prieš pradedant dirbti. Matavimai atliekami šiais atvejais:

    1.1. Įrenginiui siunčiant duomenis – kai įrenginys atlieka visus reikalingus veiksmus gauti užklausai, sudaryti naujai žinutei ir ją išsiųsti.

    1.2. Įrenginiui generuojant kriptografinius sesijos raktus vienam tinklo įrenginiui – matuojama kiek energijos išnaudojama kriptografinių sesijos raktų generavimui, nusiuntimui ir atsakymo žinutei gauti komunikuojant tik su vienu įrenginiu.

    1.3. Įrenginiui generuojant kriptografinius raktus trims tinklo įrenginiams – atliekamas toks pats matavimas kaip generuojant kriptografinius sesijos raktus vienam įrenginiui, tačiau atsižvelgiama ar energijos sąnaudos nedidėja kai generuojami kriptografiniai sesijos raktai daugiau nei vienam tinklo įrenginiui.

2. Metodo sparta laiko atžvilgiu – tiriama kiek laiko užtrunka atlikti reikalingus veiksmus nusiųsti ir gauti žinutei tarp dviejų tinklo įrenginių matuojant mili-sekundėmis (ms). Matavimai atliekami šiais atvejais:

    2.1. Įrenginiui siunčiant duomenis – kai įrenginys atlieka visus reikalingus veiksmus gauti užklausai, sudaryti naujai žinutei ir ją išsiųsti.

    2.2. Įrenginiui generuojant kriptografinius sesijos raktus vienam tinklo įrenginiui – matuojama kiek laiko užtrunkama kriptografinių sesijos raktų generavimui, nusiuntimui ir atsakymo žinutei gauti komunikuojant su tik vienu įrenginiu.

    2.3. Įrenginiui generuojant kriptografinius raktus trims tinklo įrenginiams – atliekamas toks pats matavimas kaip generuojant kriptografinius sesijos raktus vienam įrenginiui, tačiau atsižvelgiama ar užtunkama daugiau laiko nusiųsti žinutę kiekvienam įrenginiui kai generuojami kriptografiniai sesijos raktai daugiau nei vienam tinklo įrenginiui.

3. Atminties resursų naudojimas – tiriama kiek atminties užima metodo programa, naudojami kriptografiniai parametrai ar sertifikatai matuojant kilobaitais (Kb).

4. Kriptografijos saugumas – tiriama ar kriptografiniai raktai, naudojami siūlomame sprendime, yra saugūs nuo brutalios jėgos atakų ir kaip dažnai juos reikia keisti.

## 4.2. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo tyrimo metodika

Šiame poskyryje tiksliai aprašomi apibrėžtų parametrų matavimo žingsniai ir metodika. Aprašoma energijos sąnaudų, spartos laiko atžvilgiu, atminties išteklių sąnaudų ir kriptografijos saugumo tyrimo metodika.

### 4.2.1. Energijos efektyvumo matavimo metodika

Energijos efektyvumas yra svarbus parametras, nes siūlomo sprendimo taikymas yra skirtas ribotų išteklių įterptinėms sistemoms, kurios gali būti taikomos transporto priemonėse. Todėl, per didelės sąnaudos gali sukelti sistemos ar duomenų perdavimo trikdžius.

Tam, kad tinkamai pamatuoti siūlomo sprendimo energijos sąnaudas atliekami tokie veiksmai:

1. Įrenginys, kuriame įdiegtas siūlomas sprendimas, yra prijungiamas prie tyrimo elektros grandinės. Ši grandinė susideda iš pastovaus 5 voltų energijos šaltinio, kurio įžeminimas yra prijungtas prie įrenginio įžeminimo ir multimetro.

2. Pastovaus energijos šaltinio 5 voltų jungtis prijungiama prie įrenginio 5 voltų jungties naudojant multimetrą, kuris tampa elektros grandinės dalis. Pagal šaltinį nr. [24], suprojektuota energijos sąnaudų matavimo elektros grandinės diagrama pavaizduota 23 pav. Šioje diagramoje pavaizduota, kad įrenginys, kuris yra tiriamas, yra izoliuotas nuo išorinių poveikių. Šie poveikiai gali būti jutikliai ar papildomai prijungta įranga prie tiriamojo objekto. Tiriamasis įrenginys prijungtas tik prie energijos šaltinio ir ampermetro (multimetro sukonfigūruoto matuoti elektros srovę amperais).

3. Multimetras sukonfigūruojamas elektros grandinėje matuoti elektros srovę mili-amperų tikslumu.

4. Pamatuojamos įrenginio energijos sąnaudas prieš paleidžiant tiriamąją programą.

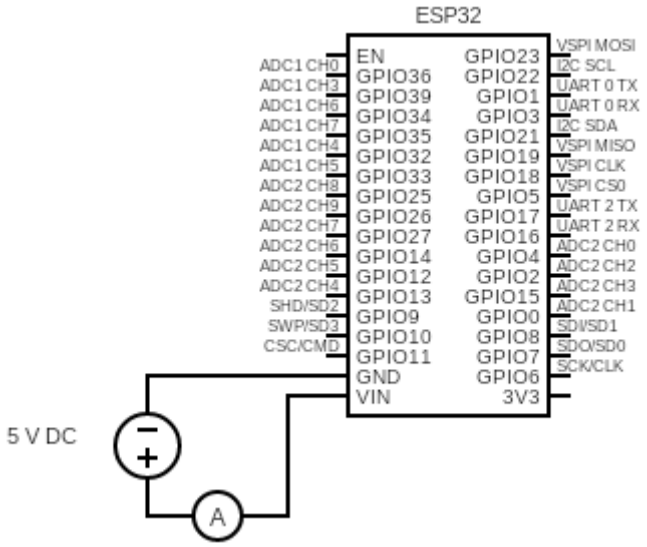
5. Atliekami 4.1 poskyryje „Energijos efektyvumas“ punktu pažymėtų parametrų tyrimai.

6. Matavimai atliekami filmuojant vaizdo kamera tam, kad tiksliai matytųsi atliekami žingsniai ir rezultatai.

7. Multimetras yra prijungtas prie kompiuterio rinkti visiems tarpiniams duomenims, iš kurių sudaromi energijos sąnaudų grafikai.

8. Matavimo rezultatai surašomi lentelėje, kuri toliau naudojama tyrimo metu.

Energijos sąnaudų matavimo elektros grandinė pavaizduota 23 pav. Pavaizduota diagrama atitinka realią elektros srovės matavimo konfigūraciją tyrimo metu. Šio tyrimo rezultatai aprašyti 4.3 poskyryje skyreliuose 4.3.1, 4.3.4 ir 4.3.8.




**23 pav.** Elektros sąnaudų matavimo elektros grandinės diagrama

### 4.2.2. Spartos laiko atžvilgiu matavimo metodika

Metodo sparta laiko atžvilgiu yra svarbus parametras, nes siūlomo sprendimo taikymas yra skirtas realiu laiku veikiančioje sistemose, kuriose duomenų vėlavimas gali kelti grėsmę žmonių sveikatai ir turtui.

Tam, kad tinkamai pamatuoti siūlomo sprendimo spartą laiko atžvilgiu atliekamai tokie veiksmai:

1. Siūlomo sprendimo prototipo metu realizuotoje programoje įjungiama informacijos spausdinimo į kompiuterio terminalą funkcija. Šioje informacijoje yra pateikia laiko žymė, kuri nurodo kada veiksmas atliktas atsižvelgiant į įrenginio įjungimą.
2. Terminalo rezultatai išsaugojami programai baigus darbą.
3. Lentelėje surašomos laiko žymės, kada buvo atlikti 4.1 poskyryje „Metodo sparta laiko atžvilgiu“ punkte pažymėti veiksmai. Lentelės duomenys toliau naudojami tyrimo metu.

### 4.2.3. Atminties resursų tyrimo metodika

Atminties resursų sąnaudos yra svarbus parametras, nes siūlomo sprendimo taikymas yra skirtas ribotų išteklių įterptinėms sistemoms. Todėl, didesnės atminties sąnaudos didina eksploatacijos sąnaudas, ribojant metodo efektyvumą ir panaudojimo galimybės.

Tam, kad tinkamai pamatuoti siūlomo sprendimo atminties resursus atliekami tokie veiksmai:

1. Programos kompiliavimo metu nustatoma kiek programa reikalauja atminties resursų.
2. Kiekvieno siūlomo sprendimo prototipo realizacijos metu įgyvendinto įrenginio sukompiliuotos programos atminties sąnaudos surašomos į lentelę, kuri toliau naudojama tyrimo metu.

### 4.2.4. Kriptografijos saugumo tyrimo metodika

Kriptografijos saugumas yra svarbus parametras, nes siūlomo sprendimo taikymas yra skirtas apsaugoti naudojamą komunikacijos technologiją, kuri siūlomo sprendimo tyrime yra „Bluetooth Low Energy“.

Tam, kad tinkamai ištirti siūlomo sprendimo kriptografijos saugumą atliekami tokie veiksmai:

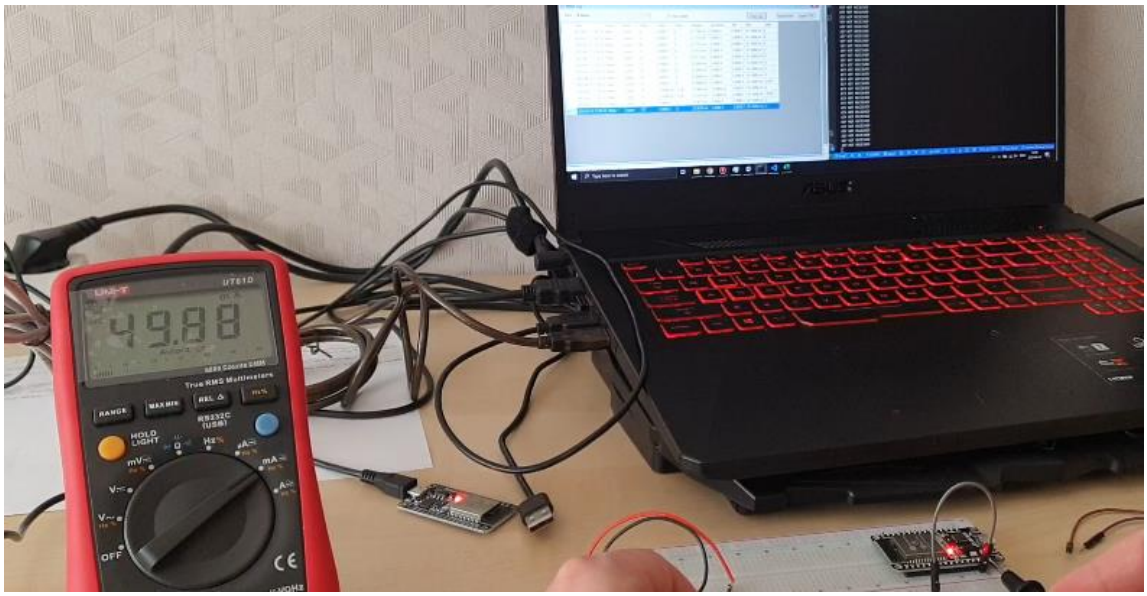
1. Tiriama ar galima suskaičiuoti naudojamą kriptografinį sesijos raktą brutalios jėgos ataka (brute-force attack) ir kitomis atakomis.
2. Išanalizuojamos galimos atakos naudojamoms kriptografinėms technologijoms ir metodams:
    2.1. Tiriamos galimos atakos ir silpnybės „AES-CBC 128“ šifravimo algoritmui.
    2.2. Tiriamos galimos atakos ir silpnybės „HMAC“ autentifikacijos reikšmės skaičiavimo algoritmui.
    2.3. Tiriamos galimos atakos ir silpnybės „HKDF“ kriptografinių raktų išvedimo funkcijai.

## 4.3. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo tyrimo rezultatai

Šiame poskyryje pateikiami tyrimo rezultatai kiekvienam tiriamam parametrui atskirai. Aprašomi atlikti energijos sąnaudų, spartos laiko atžvilgiu, atminties išteklių sąnaudų ir kriptografijos saugumo tyrimo rezultatai.

### 4.3.1. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo energijos efektyvumo tyrimo rezultatai

Pirmuoju žingsniu pamatuota kiek įrenginys naudoja energijos prieš siūlomo sprendimo prototipo programos paleidimą. Pastebėta, kad įrenginys naudoja apytiksliai 50 mA srovės. Matavimas atliktas taip, kaip pavaizduota 24 pav. kairėje matomame multimetre.




**24 pav.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodui realizuoti naudojamo įrenginio energijos sąnaudos prieš metodo paleidimą

Antruoju žingsniu atliktas matavimas rakto generavimui ir išsiuntimui vienam tinklo įrenginiui. Pastebėta, kad įrenginys daugiausiai naudojo 77 mA srovės ir tai truko 266 ms.

Trečiuoju žingsniu, tinklo topologija sukonfigūruota su trimis įrenginiais, prie kurių siūlomo sprendimo prototipo kliento įrenginys prisijungia ir atlieka kriptografinių raktų apsikeitimą. Pastebėta, kad įrenginys daugiausiai naudojo 79 mA srovės ir tai truko 600 ms.

Ketvirtuoju žingsniu, tinklo topologija sukonfigūruota tik su vienu įrenginiu, prie kurio prototipo kliento įrenginys prisijungia. Atlikus energijos sąnaudų matavimą siunčiant ir gaunant žinutę, pastebėta, kad įrenginio energijos sąnaudos neviršijo 68 mA srovės ir truko 33 ms laiko.

Penktuoju žingsniu, tinklo topologija sukonfigūruota su trimis tinklo įrenginiais prie kurių kliento įrenginys prisijungia. Atlikus energijos sąnaudų matavimą siunčiant ir gaunant žinutės tarp šių įrenginių, pastebėta, kad įrenginiai daugiausiai naudoja 74 mA srovės ir tai truko 300 ms.

Šeštuoju žingsniu atliktas serverio, prie kurio klientas prisijungia, įrenginio kriptografinių raktų gavimo energijos sąnaudų matavimas. Pastebėta, kad įrenginys daugiausiai naudojo 57 mA srovės ir tai truko 600 ms.

Septintuoju žingsniu atliktas serverio įrenginio žinučių gavimo ir siuntimo energijos sąnaudų matavimas. Pastebėta, kad įrenginys daugiausiai naudojo 56 mA ir tai truko 300 ms.

Išviso atlikti 5 matavimai ir visų žingsnių rezultatai pateikti 1 lentelėje.

**1 lentelė.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo energijos sąnaudos

| **Žingsnis / Matavimas** | **Pirmas** | **Antras** | **Trečias** | **Ketvirtas** | **Penktas** | **Šeštas** | **Septintas** |
|---|---|---|---|---|---|---|---|
| 1. | 50 mA | 77 mA | 79 mA | 68 mA | 74 mA | 57 mA | 56 mA |
| 2. | 51 mA | 78 mA | 79 mA | 69 mA | 75 mA | 58 mA | 56 mA |
| 3. | 50 mA | 77 mA | 78 mA | 68 mA | 74 mA | 57 mA | 57 mA |
| 4. | 53 mA | 76 mA | 80 mA | 70 mA | 77 mA | 58 mA | 57 mA |
| 5. | 52 mA | 77 mA | 79 mA | 70 mA | 75 mA | 58 mA | 58 mA |
| **Vidurkis:** | **51 mA** | **77 mA** | **79 mA** | **69 mA** | **75 mA** | **58 mA** | **57 mA** |
| **Bendras vidurkis:** | **66.57 mA** | | | | | | |

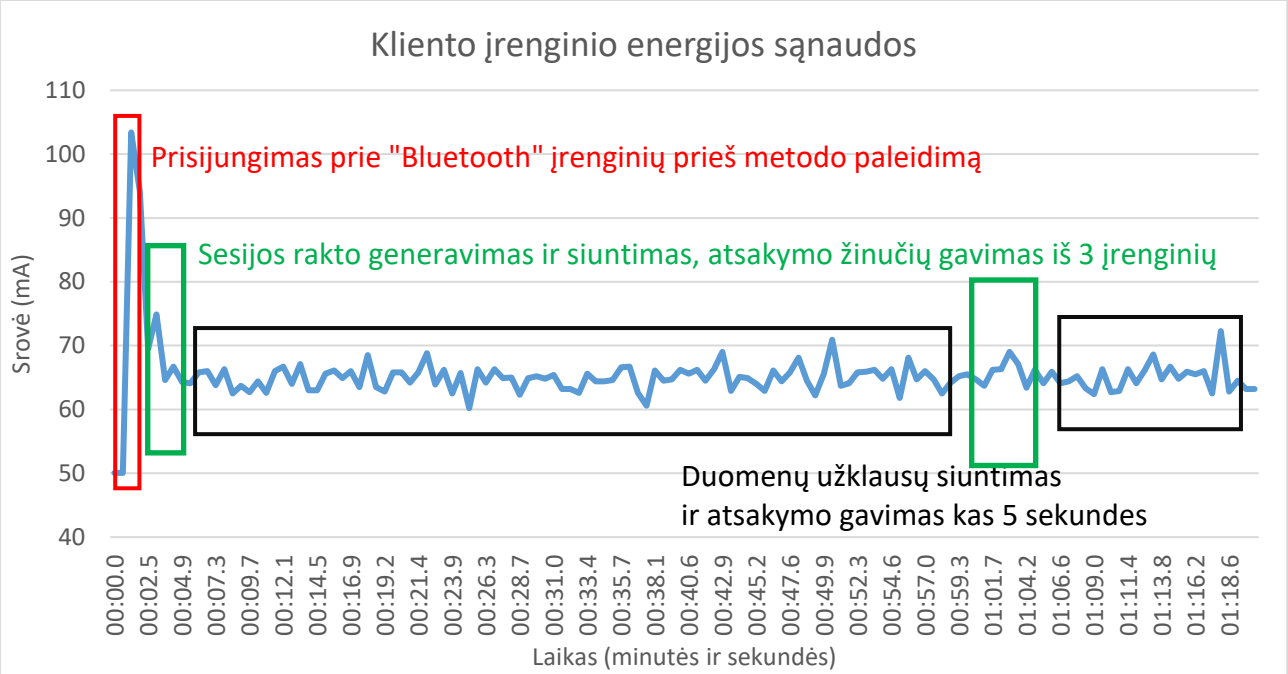
Prijungus multimetrą prie kompiuterio ir įrašant kiekvieną srovės pokytį sudaryti energijos sąnaudų grafikai pavaizduoti 25 pav. ir 26 pav., ir suskaičiuotos galios sąnaudos naudojant galios lygtį (1).

$$P = I \times U; \quad (1)$$

čia P – galia, I – srovė, U – įtampa.

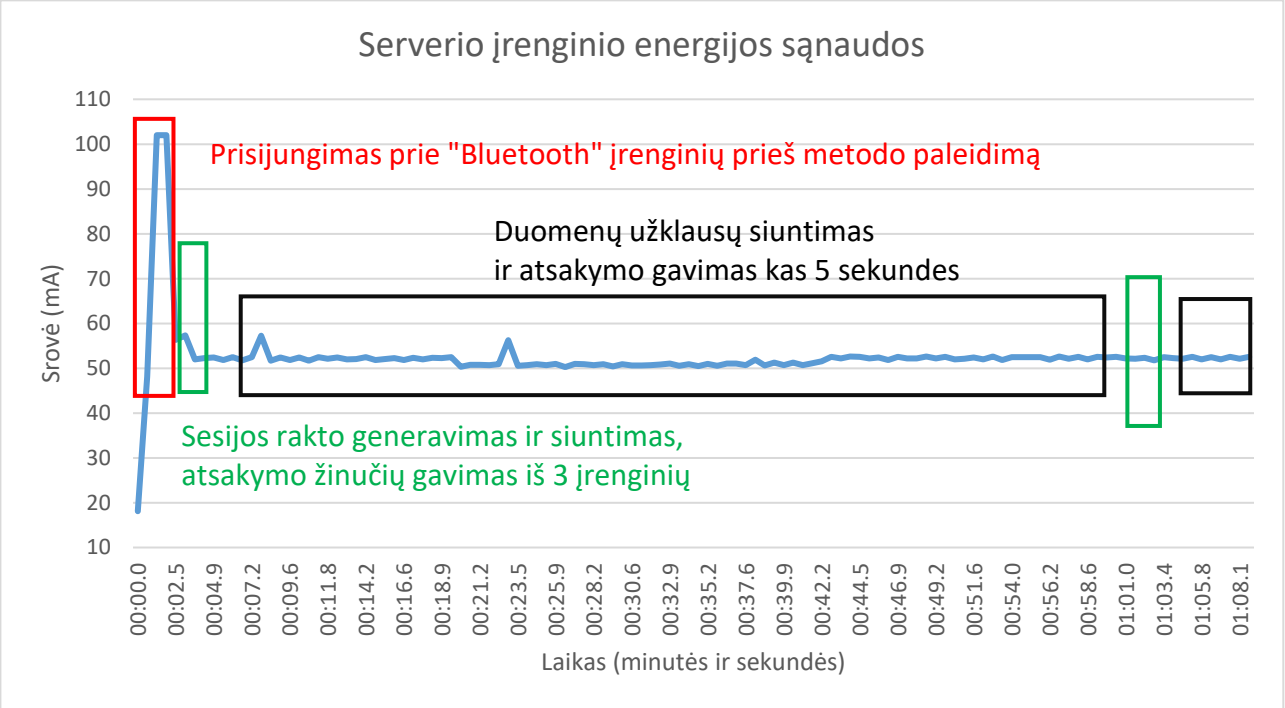
Duomenys, pagal kuriuos sudaryti grafikai, patekti prieduose 13 lentelėje ir 14 lentelėje. Šiuose grafikuose matoma, kad didžiausias energijos suvartojimas buvo įrenginio paleidimo metu. Šis energijos suvartojimas yra skirtas prisijungti prie kitų „Bluetooth“ tinklo įrenginių, tai atliekama prieš siūlomo sprendimo metodo paleidimą. Skaičiuojant kiek tyrimo metu įrenginys sunaudojo energijos

galios vatais, šie duomenys įtraukiami tam, kad palyginti su kitų, rinkoje esančių, metodų energijos sąnaudomis. Svarbu pabrėžti, kad pirmojo energijos sąnaudų testo metu, kriptografiniai sesijos raktai keičiami kas 60 sekundžių.




**25 pav.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo kliento įrenginio energijos sąnaudų grafikas

Įrenginys buvo įjungtas 80 sekundžių ir srovės vidurkis su prisijungimu prie „Bluetooth“ įrenginių buvo 65.354 mA. Kadangi įrenginys prijungtas prie 5 voltų įtampos, pritaikius galios lygtį (1), gauname, kad siūlomo sprendimo kliento įrenginio energijos sąnaudos yra 0.327 Vato (W).




**26 pav.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo serverio įrenginio energijos sąnaudų grafikas

Įrenginys buvo įjungtas 69 sekundes ir srovės vidurkis su prisijungimu prie „Bluetooth“ įrenginių buvo 52.4564 mA. Kadangi įrenginys prijungtas prie 5 voltų įtampos, pritaikius galios lygtį (1) gauname, kad siūlomo sprendimo serverio įrenginio energijos sąnaudos yra 0.2623 Vato (W).

### 4.3.2. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo spartos laiko atžvilgiu tyrimo rezultatai

Spartos laiko atžvilgiu tyrimas vykdomas išsaugojus terminale atspausdintus duomenis nurodančius kada ir kokie veiksmai buvo atlikti. Taip pat, pastebėta, kad atspausdinant daugiau nei vieną eilutę, spausdinimo laikas užtrunka 10 mili-sekundžių. Šis laikas atskaičiuojamas iš galutinio rezultato. Kaip pavaizduota 27 pav. kairėje, apskliaustas skaičius yra terminalo teksto eilutės atspausdinimo laikas nuo įrenginio paleidimo skaičiuojant mili-sekundėmis. Pirmoji eilutė, kuri pažymėta „TIMER“ žyma, nurodo laiką, kada pradėtas kriptografinio rakto generavimas. Žyma „HKDF“ nurodo, kad kriptografinio rakto generavimas yra baigtas, taip pat, šioje eilutėje pateikiamas sugeneruotas kriptografinis raktas. Abi eilutės žymą tą patį laiką nuo įrenginio paleidimo – 5821 ms, Taip pat, ta pati laiko žymė atspausdinama ir „AES_ENC“ pažymėtoje eilutėje, kuri žymi kriptografinio rakto šifravimo pabaigą ir atspausdina kriptografinio rakto šifrogramą. Iš šių rezultatų daroma išvada, kad kriptografinio rakto generavimas ir užšifravimas užtrunka trumpiau nei 10 ms.

```
I (5821) TIMER: FLAG for primary key generation
I (5821) HKDF: e3 f6 6e e2 84 eb 1d 23 ac 6c 90 48 ec 95 35 0a
I (5821) AES_ENC: 07 63 a7 4d 2f 7c ac 0d d1 8d b4 02 87 41 26 e9
I (5821) SIUNCIA: 00 5f 69 51 bd 6a f3 24 5e 6e 4f 57 92 bf b1 ba
I (5831) SIUNCIA: 67 2b e8 96 da df 2b 10 7d 89 a1 b2 e6 16 8a 3a
I (5841) SIUNCIA: 01 00 07 63 a7 4d 2f 7c ac 0d d1 8d b4 02 87 41
I (5841) SIUNCIA: 26 e9
```

**27 pav.** Siūlomo sprendimo kliento vieno kriptografinio rakto generavimo laiko sąnaudos

Tam, kad kliento įrenginys gautų kriptografinio sesijos rakto apsikeitimo atsakymą iš serverio įrenginio prireikė apie 220 ms. Atsakymo žinutės informacija pavaizduota 28 pav. Tačiau, antrą kartą siunčiant generuojant ir siunčiant kriptografinį raktą prireikė 200 ms. Pastebėta, kad kartojant veiksmus, siūlomas sprendimas, šių veiksmų įvykdymui užtrunka trumpiau.

```
I (6041) NTF_MSG: f9 29 75 03 a5 e8 25 46 d0 9f 98 d6 6f 29 f6 a0
I (6041) NTF_MSG: 65 fc 8d cd 3e c5 5e 9e 7d e9 13 dd ee 52 ce 2c
I (6051) NTF_MSG: 10 00 37 a9 a2 34 df 44 96 15 2b 83 73 0e 07 d0
I (6061) NTF_MSG: b6 a3
```

**28 pav.** Siūlomo sprendimo kliento vieno kriptografinio rakto atsakymo iš serverio įrenginio laiko sąnaudos

Tuos pačius rezultatus galima pastebėti duomenų rinkimo žinutės generavime ir siuntime pavaizduotame 29 pav. Kairėje atspausdinta laiko žymė rodo, kad žinutės užšifravimas netrunka ilgiau 10 ms.

```
I (10931) TIMER: FLAG
I (10931) AES_ENC: 38 5e bf d1 89 0b 7c 94 e2 70 d1 8e 3b eb 66 c0
I (10931) SIUNCIA: a9 93 ea c1 f3 06 a5 8e 5b 36 7b 95 0a 1b c7 12
I (10941) SIUNCIA: 28 c7 7f cd a1 9a fa 27 67 fd e2 69 c6 e5 0d 28
I (10941) SIUNCIA: 20 00 38 5e bf d1 89 0b 7c 94 e2 70 d1 8e 3b eb
I (10951) SIUNCIA: 66 c0
```

**29 pav.** Siūlomo sprendimo kliento vienos duomenų užklausos žinutės generavimo laiko sąnaudos

Vienam įrenginiui siunčiant žinutę ir gaunant atsakymą užtrukta 200 ms. Gauto atsakymo informacija pavaizduota 30 pav.

```
I (11151) GATTC_MULTIPLE_DEMO: ESP_GATTC_NOTIFY_EVT, Receive notify value:

I (11151) NTF_MSG: 12 27 a2 4c af 84 61 88 a3 6f 3b 68 9f 76 37 e1
I (11161) NTF_MSG: 09 b3 89 49 9b 26 58 49 d9 f6 8f d2 7d d6 e4 f7
I (11161) NTF_MSG: 02 01 34 09 7e 4a 37 85 6d 02 57 b2 81 52 15 3a
I (11171) NTF_MSG: c3 1f
```

**30 pav.** Siūlomo sprendimo kliento vienos duomenų užklausos žinutės atsakymo gavimo laiko sąnaudos

Surinkus ir išanalizavus surinktus duomenis atspausdintus terminale, gauti rezultatai pateikti 2 lentelėje. Pastebėta, kad visi atliekami siūlomo sprendimo veiksmai užtrunka trumpiau nei 10 ms. Taip pat, siunčiant duomenų užklausas trims skirtingiems įrenginiams ir atspausdinant visus rezultatus terminale (spausdinimas terminale užtrunka 10 ms dviem eilutėms teksto) neužtruko ilgiau 380 ms duomenų užklausų generavime, šifravime, siuntime ir atsakymo gavime iš visų trijų įrenginių. Tiek pat laiko užtrukta ir skaičiuojant naujus sesijos raktus, juos šifruojant, siunčiant, ir gaunant atsakymo žinutes iš visų trijų įrenginių. Todėl, daroma išvada, kad daugiausiai laiko išnaudojama komunikacijos technologijos, siunčiant duomenis.

**2 lentelė.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo laiko sąnaudos

| **Atliekami veiksmai** | **Laiko sąnaudos (ms)** |
|---|---|
| Rakto generavimas ir užšifravimas | < 10 |
| Rakto gavimo atsakymo žinutės autentifikavimas ir iššifravimas | < 10 |
| Gauto naujo sesijos rakto žinutės autentifikavimas ir rakto iššifravimas | < 10 |
| Duomenų užklausos generavimas ir užšifravimas | < 10 |
| Duomenų užklausos atsakymo žinutės autentifikavimas ir iššifravimas | < 10 |
| Gautos duomenų užklausos žinutės autentifikavimas ir užklausos iššifravimas | < 10 |
| Duomenų užklausų siuntimas, apdorojimas ir atsakymo gavimas komunikuojant su trimis tinklo įrenginiais | 380 |
| Kriptografinio rakto apsikeitimas su vienų įrenginiu | 220 |
| Duomenų žinutės apsikeitimas su vienu įrenginiu | 200 |

### 4.3.3. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo atminties išteklių sąnaudų tyrimo rezultatai

Kliento įrenginio programos atminties išteklių sąnaudos pavaizduotos 31 pav. Visos programos atminties sąnaudos pažymėtos žyma „Total image size“ ir nurodo 622053 baitų (622.05 Kb) dydį. Į programos dydį įeina ir analizei skirtų duomenų spausdinimas terminalo lange.

```
Total sizes:
Used static DRAM:   32216 bytes (  92364 remain, 25.9% used)
      .data size:   16304 bytes
      .bss  size:   15912 bytes
Used static IRAM:   90546 bytes (  40526 remain, 69.1% used)
      .text size:   89519 bytes
   .vectors size:    1027 bytes
Used Flash size :  515203 bytes
      .text     :  418967 bytes
      .rodata   :   95980 bytes
Total image size:  622053 bytes (.bin may be padded larger)
```

**31 pav.** Siūlomo sprendimo kliento įrenginio programos atminties išteklių sąnaudos

Serverio įrenginio programos atminties išteklių sąnaudos pavaizduotos 32 pav. Visos programos atminties sąnaudos pažymėtos žyma „Total image size“ ir nurodo 616113 baitų (616.11 Kb) dydį. Į programos dydį įeina ir analizei skirtų duomenų spausdinimas terminalo lange. Pastebėta, kad serverio programa užima 5940 baitų mažiau už kliento programą.

```
Total sizes:
Used static DRAM:   32200 bytes (  92380 remain, 25.8% used)
      .data size:   16352 bytes
      .bss  size:   15848 bytes
Used static IRAM:   90026 bytes (  41046 remain, 68.7% used)
      .text size:   88999 bytes
   .vectors size:    1027 bytes
Used Flash size :  509735 bytes
      .text     :  412955 bytes
      .rodata   :   96524 bytes
Total image size:  616113 bytes (.bin may be padded larger)
```

**32 pav.** Siūlomo sprendimo serverio įrenginio programos atminties išteklių sąnaudos

Abiejų siūlomo sprendimo programų atminties išteklių sąnaudos pateiktos 3 lentelėje.

**3 lentelė.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo atminties išteklių sąnaudos

| **Programa** | **Atminties išteklių sąnaudos baitais** |
|---|---|
| Kliento įrenginio programa | 622053 |
| Serverio įrenginio programa | 616113 |
| | |
| **Skirtumas** | **5940** |

### 4.3.4. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodas kriptografijos saugumo tyrimo rezultatai

„AES-CBC“ šifras priklauso AES šifrų šeimai, visi AES šifrai yra atsparūs brutalios jėgos atakoms, nes suskaičiuoti teisingą naudotą kriptografinį raktą užtrunka per ilgai. Taip pat, siūlomame sprendime, kiekvienoje žinutėje yra pateikiama žinutės autentifikacijos reikšmė (MAC), kuri suteikia papildomo saugumo nuo tinklui nepriklausančių įrenginių ir įsilaužėlių ar duomenų žinutės įterpimo tinkle. Kadangi gavus kiekvieną žinutę yra patikrinama autentifikacijos reikšmė prieš pradedant iššifruoti duomenis, pirma reikia apeiti sistemos autentifikaciją. Verta paminėti, kad sistemoje siunčiamos žinutės yra visada šifruotos, todėl panaikinamos visos galimybės panaudoti atakas, kurioms būtina turėti tekstogramą (duomenys prieš šifravimą), kad galėtų patikrinti gautą rezultatą ar ieškoti pasikartojančių blokų šifrogramoje. Taip pat, kiekvieno šifravimo metu naudojama kintanti druskos (salt) reikšmė, kuri pakeičia šifrogramą net siunčiant dvi tos pačios tekstogramos žinutes, kurios šifruojamos tuo pačiu kriptografiniu sesijos raktu. Kadangi naudojama kintančios druskos reikšmė, kuri yra skaitliukas, tai padeda ir sinchronizuoti tinklo įrenginių veikimą. Todėl sumažinamas sėkmės šansas įterpiant seniau nuskaitytą sistemos siųstą duomenų žinutę. Ši funkcija padidina atsparumą prieš pakartojimo ataką, kadangi duomenų žinutė tinkle neatitiks skaitliuko reikšmės tarp komunikuojančių įrenginių [25].

Sistemos saugumas papildomai yra padidinamas kriptografinių sesijos raktų keitimu. Šis kriptografinių raktų keitimas gali būti konfigūruojamas skirtingai, tačiau tyrimo metu, raktų keitimas atliekamas kas vieną minutę, o duomenų žinučių siuntimas atliekamas kas penkias sekundes. Atsižvelgiant į anksčiau aptartą sistemos kriptografijos saugumą, galima pastebėti, kad sesijos raktai gali būti keičiami ir didesniais laiko periodais. Kadangi energijos sąnaudų tyrimo metu pastebėta, kad energijos sąnaudos nekyla generuojant ir keičiantis naujais kriptografiniais sesijos raktais, daroma išvada, kad raktų keitimas gali būti atliekamas taip dažnai kaip testuota tyrimo metu. Ši funkcija padeda papildomai sumažinti rakto atspėjimo tikimybę. Jeigu sistemoje pagaunamas duomenų žinutės paketas, kuriam pavyksta apskaičiuoti kriptografinį raktą, priklausomai kaip sistemoje kriptografinių raktų keitimas yra sukonfigūruotas, suskaičiuotas raktas gali būti nebepanaudojamas, nes sistemoje kriptografiniai raktai jau bus pakeisti.

Žinučių autentifikacijos reikšmei (MAC) apskaičiuoti naudojama HMAC (Hashed Message Authentication Code) santraukos funkcija yra laikoma saugia ir patikima, nes ši funkcija yra atspari žodyno atakai ir dėl to, kad suskaičiuota autentifikacijos reikšmę itin sunku padirbti nežinant slaptojo rakto. Tačiau yra svarbu, kad būtų naudojami skirtingi slaptieji raktai tam, kad užtikrinti saugumą. Siūlomame sprendime kiekvienas komunikacijos tunelis komunikuoja naudojant atskirus kriptografinius sesijos raktus, taip padidinant HMAC suteikiamą saugumą. Taip pat, siūlomame sprendime naudojama „HMAC-SHA256“ variacija, kurios saugumas padidinamas ir dėl to, kad ši funkcija reikalauja pakankamai ilgo rakto, kas panaikina brutalios jėgos atakos sėkmę. „HMAC-SHA256“ santrauka yra negrįžtama (not reversible), todėl analizuojant sistemos siunčiamuose žinutėse pridėtą autentifikacijos reikšmę nėra prasmės. Be to, HMAC santraukos funkcija yra atspari ilgio pratęsimo atakai (length extension attack), kurios metu galima pridėti papildomus duomenis prie žinutės, kuriai skaičiuojama autentifikacijos reikšmė, ir pratęsti santraukos funkciją [26].

Kriptografinių sesijos raktų išvedimo funkcija HKDF (HMAC-based Key derivation Function) yra paremta HMAC santraukos funkcija, kuri analizuota anksčiau. HKDF gali veikti su ir be druskos (salt) reikšmės, tačiau druskos reikšmės naudojimas sustiprina HKDF funkcijos rezultatą. Todėl

siūlomame sprendime druskos reikšmė, sprendimo realizacijoje tai yra skaitliukas, yra pridedama funkcijos vykdyme. Taip pat prie druskos reikšmės papildomai pridedama ir laiko žymą. Ši žyma yra skaičiuojama nuo įrenginio paleidimo, todėl vienos mili-sekundės skirtumas, atliekant skaičiavimus, pakeis HKDF funkcijos apskaičiuojamą kriptografinį sesijos raktą. Verta paminėti, kad HKDF turi du parametrus, kurie keičią apskaičiuojamą raktą: druskos (salt) ir informacijos (info) parametrai. Skaitliukas paduodamas į informacijos parametrą, laiko žymė į druskos parametrą. Dėl šių įgyvendintų parametrų, raktų išvedimo funkcija HKDF visuomet sugeneruoja išskirtinį kriptografinį sesijos raktą [26].

### 4.3.5. DTLS protokolo energijos efektyvumo tyrimo rezultatai

Pirmuoju žingsniu pamatuota kiek įrenginys naudoja energijos prieš DTLS protokolo programos paleidimą. Pastebėta, kad įrenginys naudoja apytiksliai 72 mA srovės. Matavimas atliktas taip pat, kaip transporto priemonių lokalaus belaidžio tinklo siūlomo metodo tyrimo metu pavaizduoto 24 pav.

Antruoju žingsniu pamatuota kiek DTLS kliento įrenginys naudoja energijos prisijungiant prie DTLS serverio įrenginio komunikacijai sudaryti. Pastebėta, kad įrenginys naudojo apytiksliai 176 mA srovės ir tai truko 367 ms.

Trečiuoju žingsniu pamatuota kiek DTLS kliento įrenginys naudoja energijos sudarant naują komunikacijos sesiją su DTLS serverio įrenginiu. Pastebėta, kad įrenginys naudojo apytiksliai 130 mA srovės ir tai truko 970 ms.

Ketvirtuoju žingsniu pamatuota kiek DTLS kliento įrenginys naudoja energijos gaunant naują duomenų žinutę iš DTLS serverio įrenginio. Pastebėta, kad įrenginys naudojo apytiksliai 94 mA srovės ir tai truko 30 ms.

Penktuoju žingsniu pamatuota kiek DTLS serverio įrenginys naudoja energijos DTLS kliento įrenginiui prisijungiant prie serverio įrenginio. Pastebėta, kad įrenginys naudojo apytiksliai 140 mA srovės ir tai truko 1400 ms.

Šeštuoju žingsniu pamatuota kiek DTLS serverio įrenginys naudoja energijos siunčiant reikalingą informaciją sudaryti komunikacijos sesijai su DTLS kliento įrenginiu. Pastebėta, kad įrenginys naudoja apytiksliai 137 mA srovės ir tai truko 400 ms.

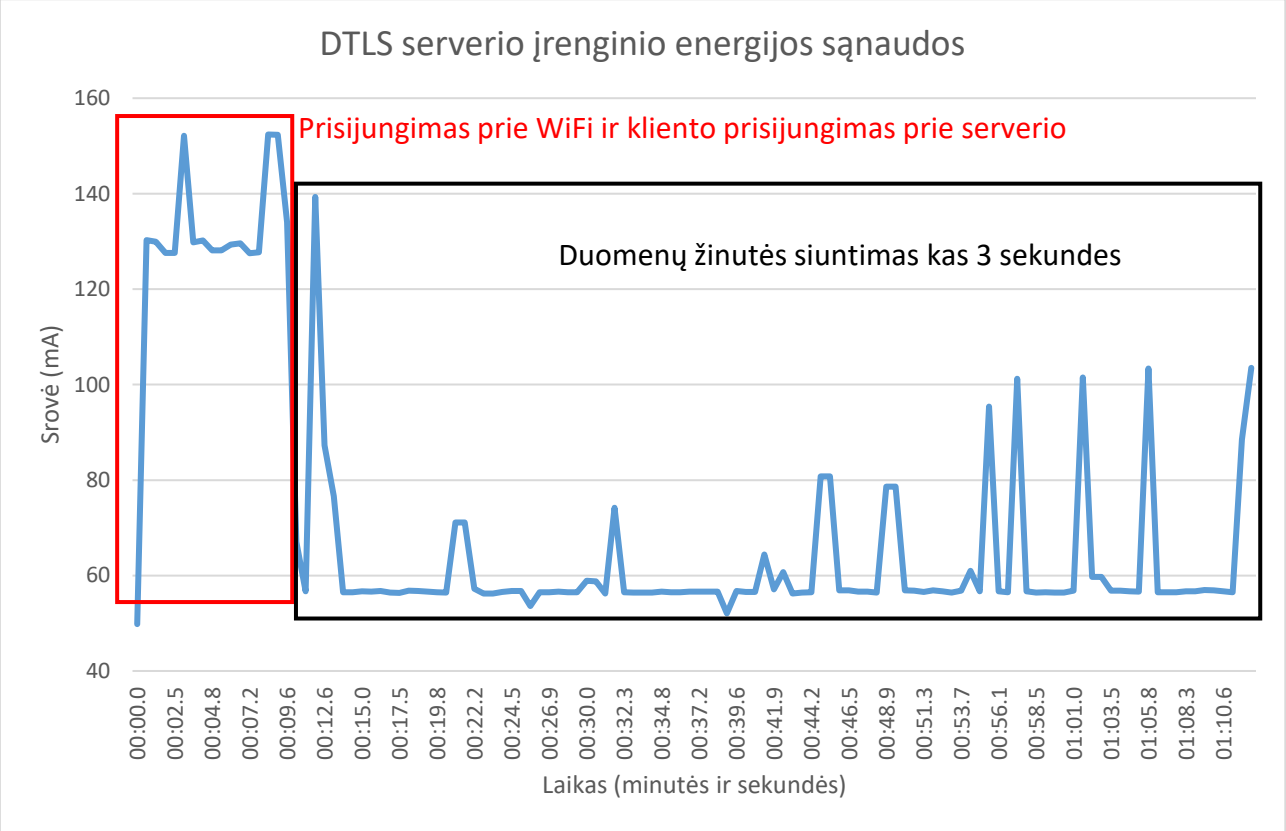
Septintuoju žingsniu pamatuota kiek DTLS serverio įrenginys naudoja energijos siunčiant duomenų žinutę DTLS kliento įrenginiui. Pastebėta, kad įrenginys naudoja apytiksliai 91 mA srovės ir tai truko 30 ms.

Visi matavimai pakartoti 5 kartus. Šių matavimų rezultatai pateikti 4 lentelėje.

**4 lentelė.** DTLS protokolo energijos sąnaudos

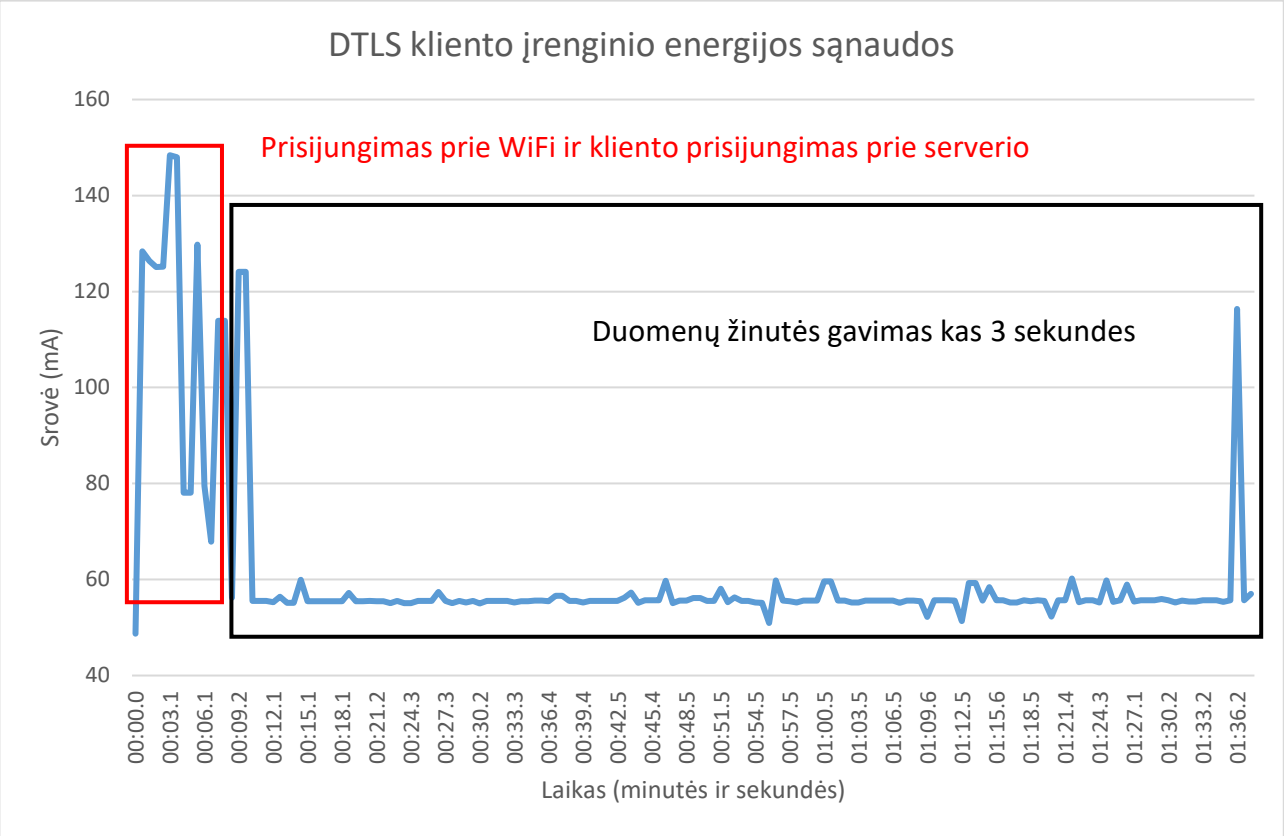
| Žingsnis / Matavimas | Pirmas | Antras | Trečias | Ketvirtas | Penktas | Šeštas | Septintas |
|---|---|---|---|---|---|---|---|
| 1. | 72 mA | 176 mA | 130 mA | 94 mA | 140 mA | 137 mA | 91 mA |
| 2. | 71 mA | 166 mA | 124 mA | 92 mA | 147 mA | 136 mA | 92 mA |
| 3. | 72 mA | 174 mA | 127 mA | 86 mA | 143 mA | 138 mA | 91 mA |
| 4. | 73 mA | 170 mA | 129 mA | 87 mA | 140 mA | 137 mA | 93 mA |
| 5. | 72 mA | 175 mA | 131 mA | 93 mA | 141 mA | 137 mA | 91 mA |
| **Vidurkis:** | **72 mA** | **172 mA** | **128 mA** | **90 mA** | **142 mA** | **137 mA** | **91 mA** |
| **Bendras vidurkis:** | **118.86 mA** | | | | | | |

Prijungus multimetrą prie kompiuterio ir įrašant kiekvieną srovės pokytį, DTLS kliento ir serverio įrenginiams sudaryti energijos sąnaudų grafikai pavaizduoti 33 pav. ir 34 pav. Duomenys, pagal kuriuos sudaryti grafikai, patekti prieduose 13 lentelėje ir 14 lentelėje. Šiuose grafikuose matoma, kad didžiausias energijos suvartojimas buvo įrenginio paleidimo metu. Šis energijos suvartojimas yra skirtas prisijungti prie „WiFi“ tinklo ir DTLS kliento įrenginiui prisijungti prie DTLS serverio įrenginio. Šis energijos suvartojimas yra įtraukiamas į energijos galios vatais skaičiavimą tam, kad palyginti su siūlomo sprendimo energijos sąnaudomis.




**33 pav.** DTLS serverio įrenginio energijos sąnaudų grafikas

Įrenginys buvo įjungtas 72 sekundes ir srovės vidurkis su prisijungimu prie „WiFi“ tinklo ir kliento įrenginio buvo 71.4395 mA. Kadangi įrenginys prijungtas prie 5 voltų įtampos, pritaikius galios lygtį (1), gauname, kad DTLS serverio įrenginio energijos sąnaudos yra 0.3572 Vato (W). Tačiau reikia atkreipti dėmesį į tai, kad DTLS serverio įrenginys pradėjo naudoti daugiau energijos žinučių siuntimui. Šį energijos sąnaudų pakilimą galima matyti grafike nuo 44 sekundės darbo. Šis energijos sąnaudų padidėjimas įvyko dėl įrenginio sesijos prie „WiFi“ tinklo atnaujinimo. Dėl to galima daryti išvadą, kad energijos galios sąnaudos gali padidėti priklausomai nuo „WiFi“ tinklo ir jo sesijų konfigūracijos.

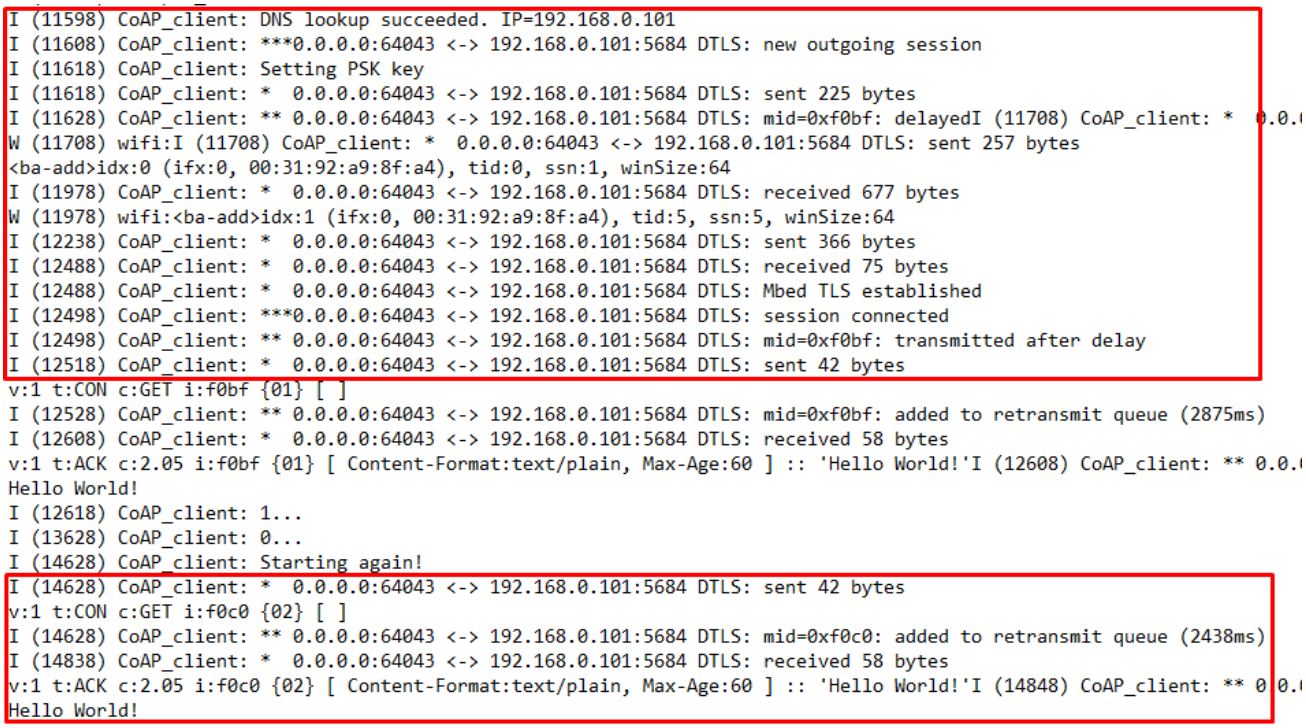



**34 pav.** DTLS kliento įrenginio energijos sąnaudų grafikas

Įrenginys buvo įjungtas 97 sekundes ir srovės vidurkis su prisijungimu prie „WiFi“ tinklo ir serverio įrenginio buvo 61.4912 mA. Kadangi įrenginys prijungtas prie 5 voltų įtampos, pritaikius galios lygtį (1), gauname, kad DTLS kliento įrenginio energijos sąnaudos yra 0.3075 Vato (W). Tačiau reikia atkreipti dėmesį į tai, kad po minutės ir 35 sekundžių pastebėtas energijos sąnaudų padidėjimas, kuris iš karto nukrito. Energijos sąnaudų didėjimas, kaip DTLS serverio įrenginyje, nepastebėtas.

### 4.3.6. DTLS spartos laiko atžvilgiu tyrimo rezultatai

Spartos laiko atžvilgiu tyrimas vykdomas išsaugojus terminale spausdinamus duomenis nurodančius kada ir kokie veiksmai buvo atlikti. Taip pat, kaip ir siūlomo sprendimo tyrime, pastebėta, kad atspausdinant daugiau nei vieną eilutę, spausdinimo laikas užtrunka 10 mili-sekundes. Šis laikas atskaičiuojamas iš galutinio rezultato. Kaip pavaizduota 35 pav. kairėje, apskliaustas skaičius yra terminalo teksto eilutės atspausdinimo laikas nuo įrenginio paleidimo skaičiuojant mili-sekundėmis. Pirmasis raudonas rėmelis žymi naujos sesijos sudarymą, kuris prasidėjo apytiksliai 11.6 įrenginio veikimo sekundę, o nauja sesija sėkmingai sudaryta 12.5 įrenginio veikimo sekundę. Sudaryti naujai sesijas tarp dviejų įrenginių DTLS protokolas užtruko 910 ms. Galima daryti išvada, kad prie tinklo topologijos prijungus daugiau įrenginių, kiekvienam įrenginiui sudaryti sesiją reikės tiek pat laiko. Antrasis raudonai pažymėtas rėmelis žymi vienos užklausos žinutės siuntimą ir atsakymo gavimą. Užklausa išsiųsta 14.6 įrenginio veikimo sekundę, o atsakymas gautas 14.8 įrenginio veikimo sekundę. Įrenginiui gauti atsakymą užtruko 210 ms.




**35 pav.** DTLS protokolo komunikacijos sesijos sudarymo ir žinutės gavimo laiko sąnaudos

Surinkus ir išanalizavus surinktus duomenis atspausdintus terminale, gauti rezultatai pateikti 5 lentelėje. Pastebėta, kad komunikacijos sesijos sudarymas užtrunka ilgiausiai, todėl kiekvieną kartą siunčiant žinutę, sudaryti naują sesiją sukeltų per didelį sistemos vėlinimą, ypač komunikuojant daugiau nei dviem tinklo įrenginiams. Todėl, visiems tyrimams, DTLS programos sukonfigūruotos palaikyti sesiją gyvą, ir siunčiant kitas žinutes naudojant tą pačia sesiją. Pastebėta, kad žinučių siuntimui ir atsakymo gavimui, nekuriant naujos komunikacijos sesijos, vidutiniškai reikia 200 ms.

**5 lentelė.** DTLS protokolo laiko sąnaudos

| Atliekami veiksmai | Laiko sąnaudos (ms) |
|---|---|
| Naujos komunikacijos sesijos sudarymas | 910 |
| Vienos žinutės išsiuntimas ir atsakymo gavimas | 200 |

### 4.3.7. DTLS protokolo atminties sąnaudų tyrimo rezultatai

DTLS kliento įrenginio programos atminties išteklių sąnaudos pavaizduotos 36 pav. Visos programos atminties sąnaudos pažymėtos žyma „Total image size“ ir nurodo 856961 baitų (856.96 Kb) dydį. Į programos dydį įeina ir analizei skirtų duomenų spausdinimas terminalo lange.

```
Total sizes:
Used static DRAM:   43572 bytes ( 137164 remain, 24.1% used)
      .data size:   14684 bytes
      .bss  size:   28888 bytes
Used static IRAM:   83402 bytes (  47670 remain, 63.6% used)
      .text size:   82375 bytes
   .vectors size:    1027 bytes
Used Flash size :  758875 bytes
      .text     :  625259 bytes
      .rodata   :  133360 bytes
Total image size:  856961 bytes (.bin may be padded larger)
```

**36 pav.** DTLS protokolo kliento įrenginio programos atminties išteklių sąnaudos

DTLS serverio įrenginio programos atminties išteklių sąnaudos pavaizduotos 37 pav. Visos programos atminties sąnaudos pažymėtos žyma „Total image size“ ir nurodo 848765 baitų (848.77 Kb) dydį. Į programos dydį įeina ir analizei skirtų duomenų spausdinimas terminalo lange. Pastebėta, kad DTLS serverio programa užima 8196 baitų mažiau nei DTLS kliento programa.

```
Total sizes:
Used static DRAM:   43340 bytes ( 137396 remain, 24.0% used)
      .data size:   14692 bytes
      .bss  size:   28648 bytes
Used static IRAM:   83402 bytes (  47670 remain, 63.6% used)
      .text size:   82375 bytes
   .vectors size:    1027 bytes
Used Flash size :  750671 bytes
      .text     :  618447 bytes
      .rodata   :  131968 bytes
Total image size:  848765 bytes (.bin may be padded larger)
```

**37 pav.** DTLS protokolo serverio įrenginio programos atminties išteklių sąnaudos

Abiejų DTLS įrenginių programų atminties išteklių sąnaudos pateiktos 6 lentelėje.

**6 lentelė.** DTLS protokolo atminties išteklių sąnaudos

| **Programa** | **Atminties išteklių sąnaudos baitais** |
|---|---|
| DTLS kliento įrenginio programa | 856961 |
| DTLS serverio įrenginio programa | 848765 |
| | |
| **Skirtumas** | **8196** |

### 4.3.8. TLS protokolo energijos efektyvumo tyrimo rezultatai

Pirmuoju žingsniu pamatuota kiek įrenginys naudoja energijos prieš TLS protokolo programos paleidimą. Pastebėta, kad įrenginys naudojo apytiksliai 74 mA srovės. Matavimas atliktas taip pat, kaip transporto priemonių lokalaus belaidžio tinklo siūlomo metodo tyrimo metu pavaizduoto 24 pav.

Antruoju žingsniu pamatuota kiek TLS kliento įrenginys naudoja energijos prisijungiant prie TLS serverio įrenginio. Pastebėta, kad įrenginys naudojo apytiksliai 128 mA srovės ir tai truko 1233 ms.

Trečiuoju žingsniu pamatuota kiek TLS kliento įrenginys naudoja energijos sudarant naują komunikacijos sesiją su TLS serverio įrenginiu. Pastebėta, kad įrenginys naudojo apytiksliai 128 mA srovės ir tai truko 1733 ms.

Ketvirtuoju žingsniu pamatuota kiek TLS kliento įrenginys naudoja energijos gaunant duomenų žinutę iš TLS serverio įrenginio. Pastebėta, kad įrenginys naudojo apytiksliai 94 mA srovės ir tai truko 970 ms.

Penktuoju žingsniu pamatuota kiek TLS serverio įrenginys naudoja energijos TLS kliento įrenginiui prisijungiant prie serverio įrenginio. Pastebėta, kad įrenginys naudojo apytiksliai 153 mA srovės ir tai truko 733 ms.

Šeštuoju žingsniu pamatuota kiek TLS serverio įrenginys naudoja energijos siunčiant reikalingą informaciją, TLS kliento įrenginiui, sudaryti naują komunikacijos sesiją. Pastebėta, kad įrenginys naudojo apytiksliai 134 mA srovės ir tai truko 333 ms.

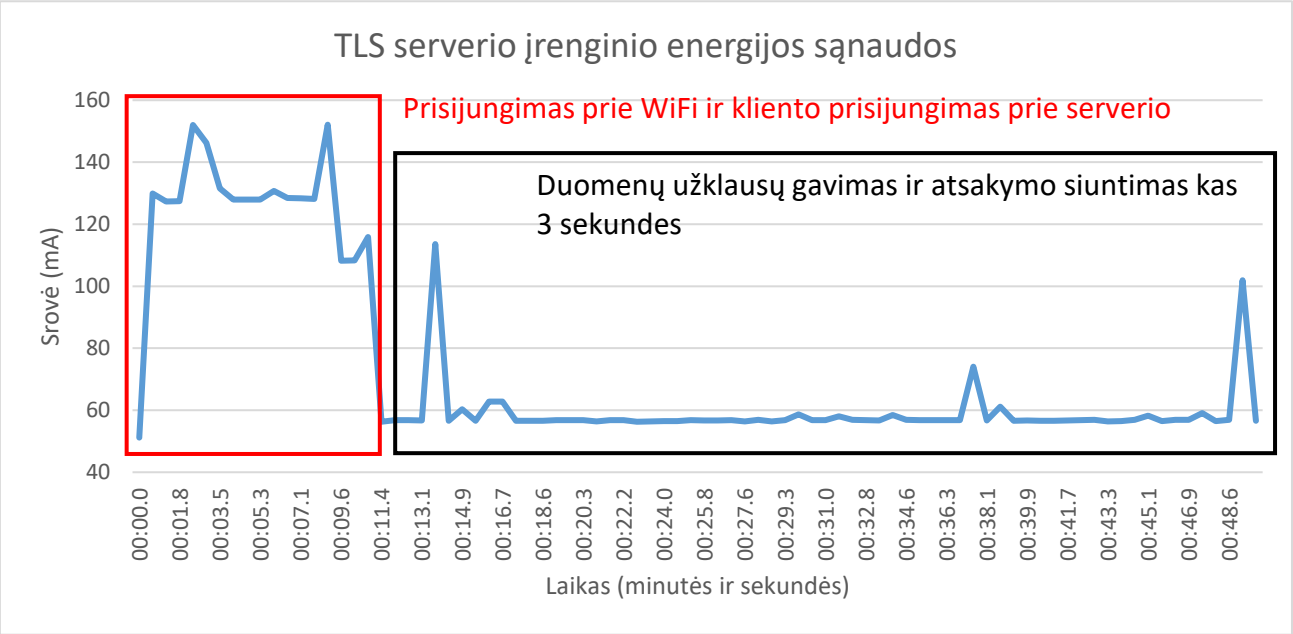
Septintuoju žingsniu pamatuota kiek TLS serverio įrenginys naudoja energijos siunčiant naują duomenų žinutę kliento įrenginiui. Pastebėta, kad įrenginys naudojo apytiksliai 125 mA srovės ir tai truko 300 ms.

Visi matavimai pakartoti 5 kartus. Šių matavimų rezultatai pateikti 7 lentelėje.

**7 lentelė.** TLS protokolo energijos sąnaudos

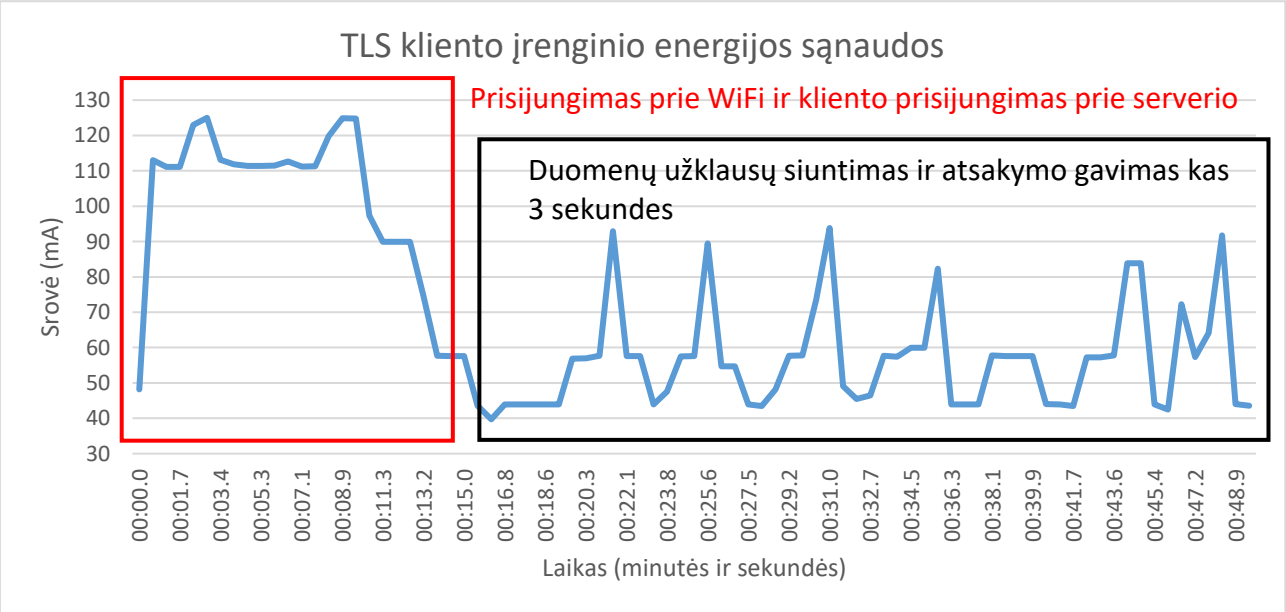
| **Žingsnis / Matavimas** | **Pirmas** | **Antras** | **Trečias** | **Ketvirtas** | **Penktas** | **Šeštas** | **Septintas** |
|---|---|---|---|---|---|---|---|
| 1. | 74 mA | 128 mA | 128 mA | 94 mA | 153 mA | 134 mA | 125 mA |
| 2. | 73 mA | 129 mA | 128 mA | 93 mA | 150 mA | 135 mA | 123 mA |
| 3. | 76 mA | 127 mA | 129 mA | 94 mA | 151 mA | 134 mA | 124 mA |
| 4. | 74 mA | 125 mA | 128 mA | 96 mA | 153 mA | 131 mA | 125 mA |
| 5. | 72 mA | 128 mA | 126 mA | 94 mA | 154 mA | 133 mA | 124 mA |
| **Vidurkis:** | **74 mA** | **127 mA** | **128 mA** | **94 mA** | **152 mA** | **133 mA** | **124 mA** |
| **Bendras vidurkis:** | **118.86 mA** | | | | | | |

Prijungus multimetrą prie kompiuterio ir įrašant kiekvieną srovės pokytį sudaryti energijos sąnaudų grafikai pavaizduoti 38 pav. ir 39 pav. Duomenys, pagal kuriuos sudaryti grafikai, patekti prieduose 13 lentelėje ir 14 lentelėje. Šiuose grafikuose matoma, kad didžiausias energijos suvartojimas buvo įrenginio paleidimo metu. Šis energijos suvartojimas yra skirtas prisijungti prie „WiFi“ tinklo ir TLS kliento įrenginiui prisijungti prie TLS serverio įrenginio. Skaičiuojant kiek tyrimo metu įrenginys sunaudojo energijos galios vatais šie duomenys įtraukiami tam, kad palyginti su siūlomo metodo energijos sąnaudomis.




**38 pav.** TLS serverio įrenginio energijos sąnaudų grafikas

Įrenginys buvo įjungtas 50 sekundžių ir srovės vidurkis su prisijungimu prie „WiFi“ ir kliento įrenginio prisijungimu prie serverio įrenginio buvo 73.098 mA. Kadangi įrenginys prijungtas prie 5 voltų įtampos, pritaikius galios lygtį (1), gauname, kad TLS serverio įrenginio energijos sąnaudos yra 0.3655 Vato (W).




**39 pav.** TLS kliento įrenginio energijos sąnaudų grafikas

Įrenginys buvo įjungtas 50 sekundžių ir srovės vidurkis su prisijungimu prie „WiFi“ ir kliento įrenginio prisijungimu prie serverio įrenginio buvo 69.491 mA. Kadangi įrenginys prijungtas prie 5 voltų įtampos, pritaikius galios lygtį (1), gauname, kad TLS serverio įrenginio energijos sąnaudos yra 0.3475 Vato (W).

### 4.3.9. TLS spartos laiko atžvilgiu tyrimo rezultatai

Spartos laiko atžvilgiu tyrimas vykdomas išsaugojus terminale spausdinamus duomenis nurodančius kada ir kokie veiksmai buvo atlikti. Taip pat, kaip ir siūlomo sprendimo tyrime, pastebėta, kad atspausdinant daugiau nei vieną eilutę, spausdinimo laikas užtrunka 10 mili-sekundes. Šis laikas atskaičiuojamas iš galutinio rezultato. Kaip pavaizduota 40 pav. kairėje, apskliaustas skaičius yra terminalo teksto eilutės atspausdinimo laikas nuo įrenginio paleidimo skaičiuojant mili-sekundėmis. Pirmoji eilutė žymi kliento įrenginio užklausą sudaryti naujai komunikacijos sesijai. Ši eilutė žymi 22.8 sekundes nuo įrenginio paleidimo. Antroji eilutė žymi atsakymą iš serverio, kuris patvirtina, kad kliento įrenginio pateiktas sertifikatas yra teisingas, šis veiksmas užtruko 570 ms. Trečioji eilutė rodo, kad TLS kliento įrenginys ir TLS serverio įrenginys sudarė komunikaciją. Tai užtruko 1590 ms nuo sertifikato patvirtinimo. Tai įvykdžius, TLS serverio įrenginys išsiunčia TLS kliento įrenginiui duomenų žinutę ir uždaro sesiją. Tai užtruko 400 ms nuo sesijos sukūrimo.

```
I (22803) example: https_request using crt bundle
I (23373) esp-x509-crt-bundle: Certificate validated
I (24963) example: Connection established...
I (24963) example: 102 bytes written
I (24963) example: Reading HTTP response...
HTTP/1.1 200 OK
Content-Length: 2091
Access-Control-Allow-Origin: *
Connection: close
Content-Type: application/json
Date: Thu, 13 Apr 2023 13:52:12 GMT
Strict-Transport-Security: max-age=631138519; includeSubdomains; preload

{"given_cipher_suites":["TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384","TLS_ECD
256_CBC_SHA384","TLS_DHE_RSA_WITH_AES_256_CBC_SHA256","TLS_ECDHE_ECDSA_WIT
_RSA_WITH_AES_128_CBC_SHA256","TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA","TLS_
_WITH_AES_128_CCM_8","TLS_DHE_RSA_WITH_AES_128_CCM_8","TLS_RSA_WITH_AES_25
A_WITH_AES_128_CBC_SHA256","TLS_RSA_WITH_AES_128_CBC_SHA","TLS_ECDH_RSA_WI
se,"able_to_detect_n_minus_one_splitting":false,"insecure_cipher_suites":{
I (25363) example: connection closed
```

**40 pav.** TLS protokolo komunikacijos sesijos sudarymo ir žinutės gavimo laiko sąnaudos

Surinkus ir išanalizavus surinktus duomenis atspausdintus terminale, gauti rezultatai pateikti 8 lentelėje. Pastebėta, kad komunikacijos sesijos sudarymas užtrunka ilgiausiai, todėl kiekvieną kartą siunčiant žinutę, sudaryti naują sesija sukeltų per didelį sistemos vėlavimą. Tačiau, kadangi programa buvo modifikuota DTLS protokole palaikyti sesiją, nuspręsta visus tyrimus atlikti TLS protokolui sudarant naują sesiją su kiekviena žinute. Atlikus kelių žinučių siuntimą, žymių skirtumų laiko sąnaudose nepastebėta.

**8 lentelė.** TLS protokolo laiko sąnaudos

| Atliekami veiksmai | Laiko sąnaudos (ms) |
|---|---|
| Naujos komunikacijos sesijos sudarymo užklausos siuntimas, sertifikato patvirtinimas ir atsakymo gavimas | 570 |
| Naujos komunikacijos sesijos sudarymas | 1590 |
| Žinutės gavimas iš serverio įrenginio sudarius komunikacijos sąsają | 400 |

### 4.3.10. TLS atminties išteklių sąnaudų tyrimo rezultatai

TLS kliento įrenginio programos atminties išteklių sąnaudos pavaizduotos 41 pav. Visos programos atminties sąnaudos pažymėtos žyma „Total image size“ ir nurodo 817817 baitų (817.82 Kb) dydį. Į programos dydį įeina ir analizei skirtų duomenų spausdinimas terminalo lange.

```
Total sizes:
Used static DRAM:   30616 bytes ( 150120 remain, 16.9% used)
      .data size:   14400 bytes
      .bss  size:   16216 bytes
Used static IRAM:   83182 bytes (  47890 remain, 63.5% used)
      .text size:   82155 bytes
   .vectors size:    1027 bytes
Used Flash size :  720235 bytes
      .text     :  544939 bytes
      .rodata   :  175040 bytes
Total image size:  817817 bytes (.bin may be padded larger)
```

**41 pav.** TLS protokolo kliento įrenginio programos atminties išteklių sąnaudos

TLS serverio įrenginio programos atminties išteklių sąnaudos pavaizduotos 42 pav. Visos programos atminties sąnaudos pažymėtos žyma „Total image size“ ir nurodo 765757 baitų (765.76 Kb) dydį. Į programos dydį įeina ir analizei skirtų duomenų spausdinimas terminalo lange. Pastebėta, kad TLS serverio programa užima 52060 baitų mažiau nei TLS kliento programa.

```
Total sizes:
Used static DRAM:   30232 bytes ( 150504 remain, 16.7% used)
      .data size:   14400 bytes
      .bss  size:   15832 bytes
Used static IRAM:   83166 bytes (  47906 remain, 63.5% used)
      .text size:   82139 bytes
   .vectors size:    1027 bytes
Used Flash size :  668191 bytes
      .text     :  556239 bytes
      .rodata   :  111696 bytes
Total image size:  765757 bytes (.bin may be padded larger)
```

**42 pav.** TLS protokolo serverio įrenginio programos atminties išteklių sąnaudos

Abiejų TLS įrenginių programų atminties išteklių sąnaudos pateiktos 9 lentelėje.

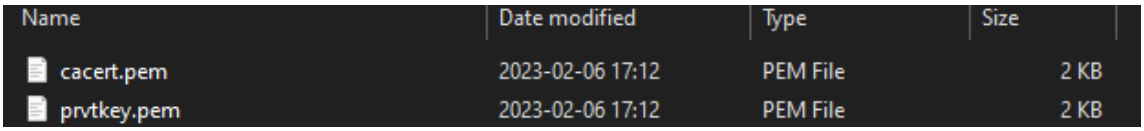
**9 lentelė.** TLS protokolo atminties išteklių sąnaudos

| **Programa** | **Atminties išteklių sąnaudos baitais** |
|---|---|
| TLS kliento įrenginio programa | 817817 |
| TLS serverio įrenginio programa | 765757 |
| | |
| **Skirtumas** | **52060** |

### 4.3.11. DTLS ir TLS kriptografijos saugumo tyrimo rezultatai

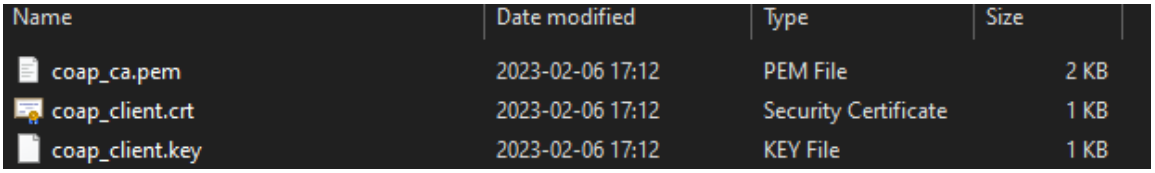
DTLS veikimas yra paremtas TLS veikimo principu. Didžiausias skirtumas yra tai, kad TLS komunikacija sudaroma TCP protokolu (Transmission Control Protocol), o DTLS – UDP protokolu (User Datagram Protocol). Taip pat, DTLS apsaugo komunikacijos tunelį nuo pasiklausymo ar manipuliavimo. Papildomai, išsprendžiama paketų praradimo ir eilės tvarkos tvarkymo problemos, kadangi DTLS išvengia duomenų siuntimo vėlinimo pasikartojančio srautinio perdavimo protokoluose [27]. Tačiau nėra tikrinimo ar visi duomenys buvo nusiųsti teisingai be klaidų, todėl duomenys kartais gali būti klaidingi.

Abu komunikacijos metodai pradžioje apsikeičia pagrindine informacija: TLS versija ir kitus parametrus. Tuomet prasideda kliento ir serverio derybos dėl komunikacijos tunelio šifravimo. Abiem įrenginiams, kliento ir serverio, reikia sutarti kokia TLS versija naudojama, pasirinkti šifravimo algoritmą ir patvirtinti vienas kito sertifikatus. Serverio įrenginys pateikia savo sertifikatą kliento įrenginiui, kuris jį turi patvirtinti. Jeigu kliento įrenginys sertifikatą patvirtiną, pradedama „Diffie-Hellman“ arba RSA algoritmai simetrinio kriptografinio rakto skaičiavimui. Kai šie veiksmai pabaigiami, serveris nusiunčia šifruotą „Finished“ teksto žinutę klientui, kuris šią žinutę iššifruoja ir patikrina fizinį adresą (MAC address). Jeigu viskas atitinka, galima pradėti siųsti duomenų žinutes. Galima pastebėti, kad tinkle atsiradus tinklo vėlavimui, kliento prisijungimas prie serverio stipriai išauga, kadangi duomenų komunikacijos sudarymui reikia apsikeista keliomis informacijos žinutėmis [28]. Atkreipus dėmesį į TLS įrenginių metodo naudojamus sertifikatus pavaizduotus 43 pav. pastebėta, kad iš viso reikia 4 KB atminties saugoti sertifikatų informacijai.




**43 pav.** TLS serverio įrenginio metodo naudojami sertifikatai

Atkreipus dėmesį į DTLS įrenginių naudojamus sertifikatus pavaizduotus 44 pav. pastebėta, kad serverio ir kliento įrenginiams reikia tiek pat atminties, kaip ir TLS metodui, saugoti sertifikatų informacijai – 4 KB.



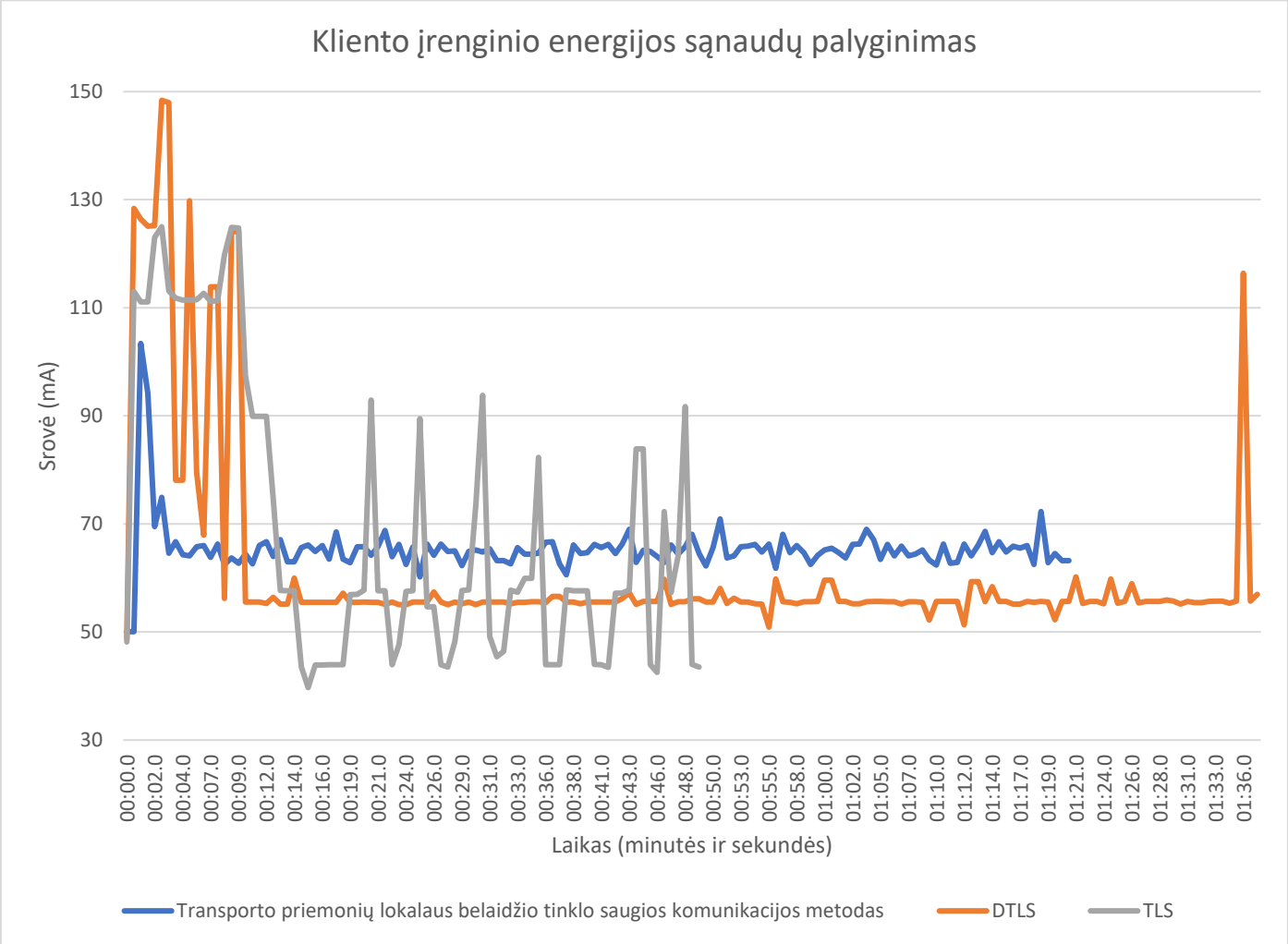

**44 pav.** DTLS metodo naudojami sertifikatai

TLS ir DTLS metodai, naudoja asimetrinį šifravimą, kuriame naudojami du kriptografiniai raktai. Šių dviejų kriptografinių raktų pagalba yra sudaromas saugus komunikacijos tunelis tarp serverio ir kliento įrenginių. Tuomet naudojamas simetrinis šifravimas duomenų apsikeitimui sudarytame saugiame komunikacijos tunelyje.

### 4.3.12. Energijos sąnaudų palyginimas

Atlikus siūlomo sprendimo, DTLS ir TLS metodų energijos sąnaudų tyrimus, atlikti palyginimai, kurių rezultatų grafikai pateikti 45 pav. ir 46 pav., o energijos sąnaudų vidurkių palyginimas pateiktas 10 lentelėje ir 11 lentelėje. Duomenys, pagal kuriuos sudaryti grafikai ir atlikti skaičiavimai, patekti prieduose 13 lentelėje ir 14 lentelėje.

Kliento įrenginio energijos sąnaudų grafike matyti, kad siūlomo sprendimo pirminės energijos sąnaudos, prisijungiant prie tinklo, yra mažesnės už DTLS ir TLS metodų, tačiau įprasto veikimo metu, siūlomo sprendimo kliento įrenginio energijos sąnaudos mažesnės tik už TLS metodo, kuris sukonfigūruotas nesaugoti komunikacijos sesijos.




**45 pav.** Kliento įrenginio energijos sąnaudų palyginimo grafikas

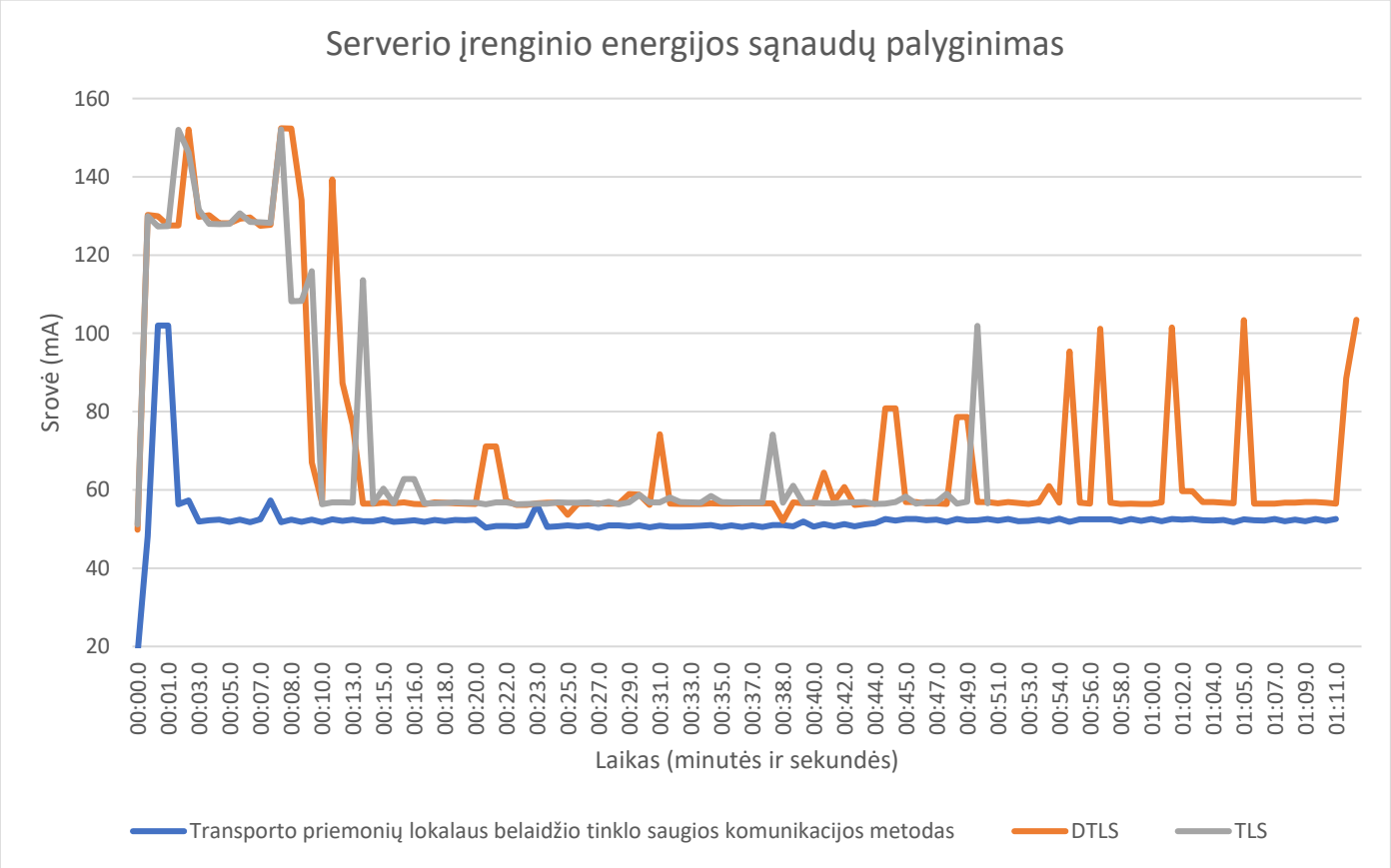
Palyginus apskaičiuotus metodų kliento įrenginio energijos sąnaudų vidurkius matyti, kad DTLS kliento įrenginys naudoja vidutiniškai 3.863 mA srovės arba 0.0195 vato mažiau už siūlomo sprendimo kliento įrenginį. Tačiau pagal grafiką matyti, kad siūlomo sprendimo kliento įrenginys visuomet naudoja apytiksliai tiek pat energijos, o DTLS įrenginiui sąnaudos pakyla kai reikia

atnaujinti „WiFi“ tinklo sąsają. Jeigu „WiFi“ tinklas atsakymo negrąžins, įrenginio sąnaudos didės. Taip pat, verta paminėti, kad siūlomo sprendimo tyrimo metu kliento įrenginys komunikavo su iš viso keturiais įrenginiais, o DTLS ir TLS tik po du įrenginius. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodas yra ypač pranašesnis jeigu komunikacijos sesijos yra trumpos. Pavyzdžiu, jeigu reikia išsiųsti tik vieną ar kelias žinutes ir nutraukti komunikacijos sesiją, siūlomas sprendimas tai atlieka naudojant mažiau energijos nes komunikacija sudaroma greičiau ir mažesnėmis energijos sąnaudomis.

**10 lentelė.** Kliento įrenginio energijos sąnaudų vidurkių palyginimas

| **Tiriamas metodas** | **Energijos sąnaudų vidurkis (mA arba W)** |
|---|---|
| Siūlomas sprendimas | 65.354 mA arba 0.327 W |
| DTLS | 61.4912 mA arba 0.3075 W |
| TLS | 69.491 mA arba 0.3475 W |
| | |
| **Mažiausiai energijos naudojantis metodas: DTLS** | **61.4912 mA arba 0.3075 W** |
| **Skirtumas nuo siūlomo sprendimo:** | **3.863 mA arba 0.0195 W** |

Serverio įrenginio energijos sąnaudų grafike matyti, kad siūlomo sprendimo serverio įrenginio pirminės energijos sąnaudos, prisijungiant prie tinklo, ir viso įrenginio veikimo metu esančios energijos sąnaudos yra mažesnės už DTLS ir TLS metodų. Taip pat, matyti, kad energijos sąnaudos yra stabilesnės su mažai pokyčių, ypač lyginant su DTLS serverio įrenginio energijos sąnaudomis.



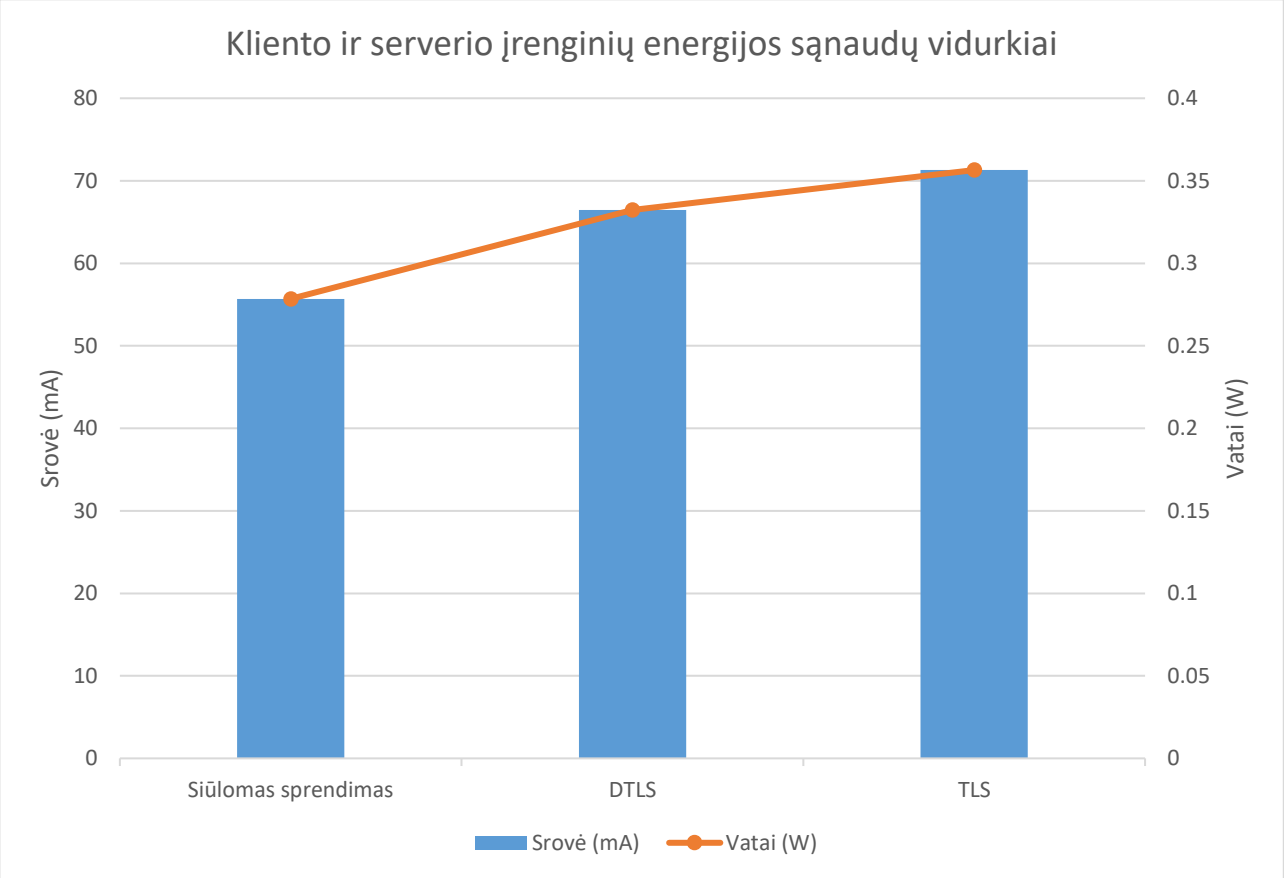

**46 pav.** Serverio įrenginio energijos sąnaudų palyginimo grafikas

Palyginus apskaičiuotus metodų serverio įrenginio energijos sąnaudų vidurkius matyti, kad mažiausiai energijos naudoja siūlomo sprendimo įrenginys. Taip yra todėl, nes siūlomo sprendimo serverio įrenginys neatlieka jokių skaičiavimų pats, o tik kaip gauna duomenų užklausą iš kliento įrenginio.

**11 lentelė.** Serverio įrenginio energijos sąnaudų vidurkių palyginimas

| **Tiriamas metodas** | **Energijos sąnaudų vidurkis (mA arba W)** |
|---|---|
| Siūlomas sprendimas | 52.4564 mA arba 0.2623 W |
| DTLS | 71.4395 mA arba 0.3572 W |
| TLS | 73.098 mA arba 0.3655 W |
| | |
| **Mažiausiai energijos naudojantis metodas: siūlomas sprendimas** | **52.4564 mA arba 0.2623 W** |

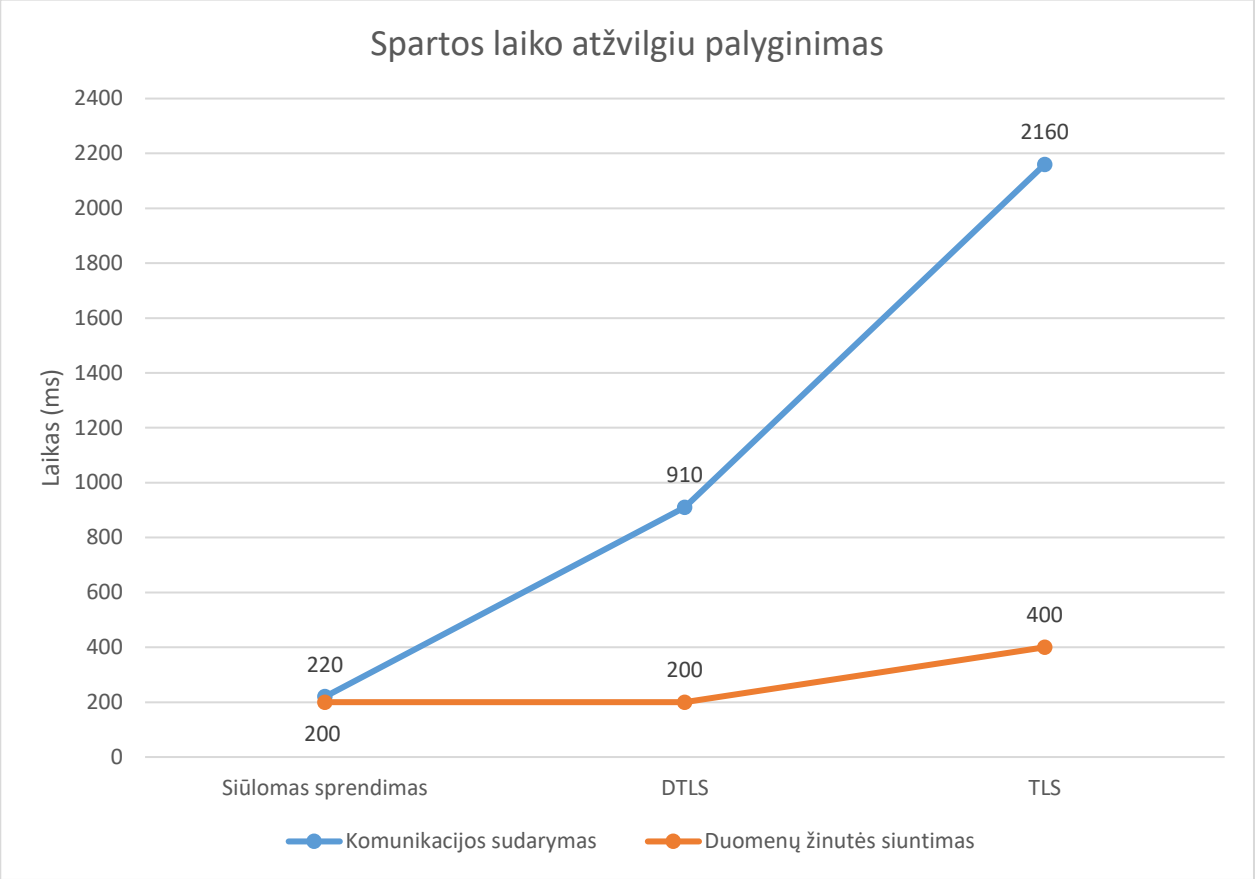
Apskaičiavus abiejų, kliento ir serverio, įrenginių energijos sąnaudų vidurkį, rezultatai pavaizduoti 47 pav. Grafike atvaizduojami energijos sąnaudų vidurkių duomenys skaičiuojant tik vieno kliento ir vieno serverio įrenginio energijos sąnaudas. Šiame grafike matosi, kad siūlomas sprendimas yra taupesnis energijos sąnaudų atžvilgiu palyginus su DTLS ir TLS metodais.




**47 pav.** Kliento ir serverio įrenginių energijos sąnaudų vidurkių grafikas

### 4.3.13. Spartos laiko atžvilgiu palyginimas

Atlikus siūlomo sprendimo, DTLS ir TLS metodų spartos laiko atžvilgiu tyrimus, atlikti palyginimai, kurių rezultatų grafikas pavaizduotas 48 pav.



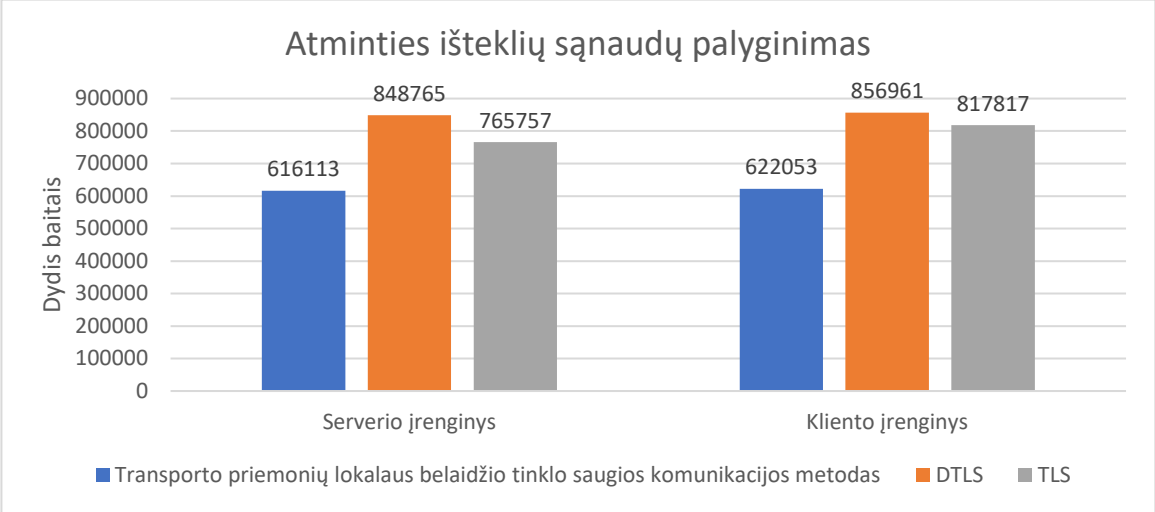

**48 pav.** Spartos laiko atžvilgiu palyginimo grafikas

Grafike matosi, kad laiko atžvilgiu, siūlomas sprendimas yra greitesnis komunikacijos sudarymo stadijoje. Ši stadija skirta apsikeisti reikalinga informacija komunikacijai sudaryti tarp dviejų įrenginių. Į šį laiką neįskaičiuojamas laikas sudaryti komunikaciją su tinklu, pavyzdžiui DTLS ir TLS įrenginiams reikalingas papildomas laikas prisijungti prie „WiFi“ tinklo ir gauti tinklo adresą. Šis laikas nėra skaičiuojamas. Siūlomo sprendimo atveju tai yra kliento įrenginio kriptografinio sesijos rakto generavimas, rakto nusiuntimas serverio įrenginiui ir atsakymo gavimas. DTLS ir TLS atveju tai yra kliento įrenginio sertifikato patvirtinimas serverio įrenginyje ir kliento prisijungimas prie serverio. Verta paminėti, kad tyrimo metu pastebėta, kad siūlomo sprendimo kliento įrenginiui antrą kartą generuojant ir siunčiant kriptografinė sesijos raktą užtrunkama trumpiau, nei tai atliekant pirmą kartą įrenginį paleidus.

Palyginus siūlomo sprendimo ir DTLS metodo duomenų žinutės siuntimo spartą laiko atžvilgių matosi, kad šie metodai užtrunka tiek pat laiko (200 ms) nusiųsti ir gauti atsakymą duomenų žinutei. Tačiau palyginus su TLS, siūlomas sprendimas ir DTLS metodas, yra spartesni du kartus.

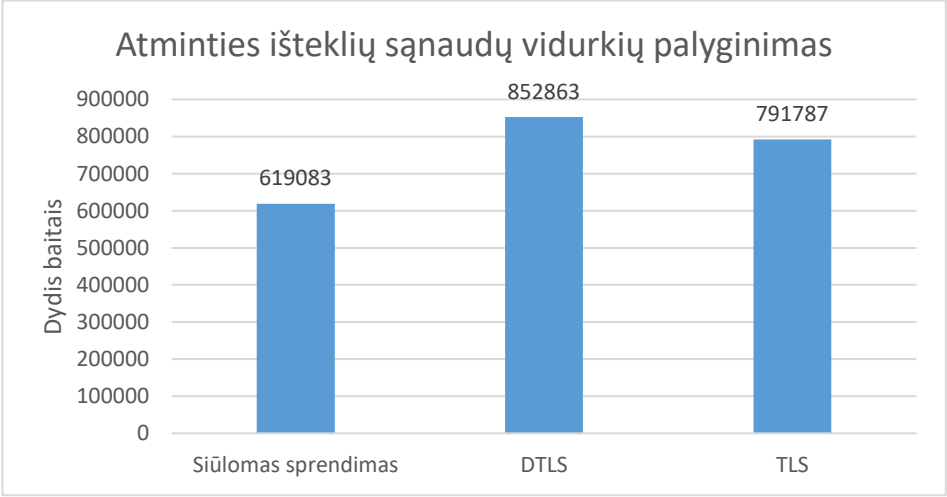
### 4.3.14. Atminties išteklių sąnaudų palyginimas

Atlikus siūlomo sprendimo, DTLS ir TLS metodų atminties išteklių sąnaudų tyrimus, atlikti palyginimai, kurių rezultatų grafikai pavaizduoti 49 pav. ir 50 pav.




**49 pav.** Atminties išteklių sąnaudų palyginimo grafikas

Grafike matosi, kad metodo programos atminties išteklių sąnaudos kliento įrenginyje yra didesnės už serverio įrenginio, tačiau didesnis skirtumas matosi tik TLS metodo atminties sąnaudose. Taip pat, siūlomo sprendimo serverio ir kliento įrenginių programos naudoja mažiausiai atminties išteklių palyginus su DTLS ir TLS metodais. Daroma išvada, kad kriptografinių raktų dydžio sumažinimas siūlomame sprendime, palyginus su sertifikatais, stipriai paveikė ir atminties išteklių sąnaudas. Taip yra nes, sistemoje nereikia saugoti sertifikatų ir sertifikatų institucijos (certificate authority), kitų parametrų ir funkcijų reikalingų atlikti skaičiavimus su šiais sertifikatais. Taip sumažėja ne tik atminties išteklių sąnaudos, bet ir padidėja metodo efektyvumas ir sumažėja energijos sąnaudos. Tačiau verta paminėti, kad DTLS kliento įrenginys naudoja mažiau energijos negu siūlomo sprendimo kliento įrenginys. Pastebėta, kad DTLS metodo energijos ir spartos laiko atžvilgiu sąnaudos buvo mažesnės už TLS, tačiau atminties išteklių sąnaudos yra didžiausios palyginus su siūlomo sprendimo ir TLS metodų atminties išteklių sąnaudomis. Tai galima pastebėti atminties išteklių sąnaudų vidurkių palyginimo grafike pateiktame 50 pav.




**50 pav.** Atminties išteklių sąnaudų vidurkių palyginimo grafikas

Atminties sąnaudų vidurkių palyginimo grafike matosi, kad DTLS protokolas naudoja apytiksliai 37.76 procentais daugiau atminties išteklių už siūlomo sprendimo metodą, o TLS metodas apytiksliai 27.9 procentais daugiau. DTLS metodas naudoja 7.71 procentais daugiau atminties išteklių palyginus su TLS metodu. Šie duomenys procentais pateiki 12 lentelėje.

**12 lentelė.** Atminties išteklių sąnaudų palyginimas procentais

| Tiriamas metodas | Atminties išteklių sąnaudos lyginant su siūlomu sprendimu procentais (%) |
|---|---|
| Siūlomas sprendimas | 100 % |
| DTLS | 137.76 % |
| TLS | 127.9 % |

### 4.3.15. Kriptografijos saugumo tyrimo rezultatų palyginimas

Skirtumas tarp siūlomo sprendimo ir TLS bei DTLS metodų yra tai, kad siūlomas sprendimas nenaudoja asimetrinio šifravimo komunikacijos tunelio sudarymui. Tai atliekama naudojant vieną sesijos kriptografinį raktą, kuris yra keičiamas periodiškai. Tačiau visi trys tirti metodai, po saugios komunikacijos sudarymo, šifruoja duomenis vienu simetrinius kriptografiniu raktu. Tačiau, siūlomas sprendimas, tyrimo metu, sukonfigūruotas naudoti 128 bitų kriptografinį raktą, o TLS ir DTLS naudoja 2048 bitų raktą. Matosi, kad TLS šeimos metodai suteikia saugesnį duomenų šifravimą, tačiau tai papildomai kainuoja skaičiavimo galios išteklių. Nors tyrimo metu DTLS ir siūlomo sprendimo siunčiamos duomenų žinutės užtruko lygiai tiek pat laiko, naudojant skaičiavimo galios atžvilgiu silpnesnę įrangą, ilgesnis kriptografinis raktas gali sulėtinti žinučių siuntimą. Taip pat, siūlomas sprendimas nesudaro saugaus komunikacijos tunelio kaip DTLS ir TLS metodai, tačiau tai padengia iš karto sistemoje įgalintais įrenginiais. Šiuo atveju, tinkle esantys įrenginiai yra iš anksto sukonfigūruoti komunikuoti tik su keliais artimais įrenginiais. Tokia sistema nėra dinaminė kaip DTLS ar TLS metoduose. Šis faktorius padidina sistemos saugumą, kadangi sumažinama galimybė į tinklą įterpti išorinį įrenginį. Saugios komunikacijos sesijos sudarymas nėra svarbus siūlomame sprendime ir todėl, kad pirmoji žinutė išsiųsta iš kliento įrenginio yra šifruota pagrindiniu raktu, kuris nėra platinamas tinkle, o yra įdiegiamas į įrenginį sistemos konfigūravimo metu. Palyginus su TLS ar DTLS tai atliekama su sertifikatais ir jų sertifikavimų institucija (certificate authority), kuri patikrina ar sertifikatai autentifikuojami. Be to, siūlomo sprendimo tinkle nėra siunčiami parametrai, kurie naudojami duomenų srauto šifravimui reikalingo kriptografinio rakto generavimui. Vietoje šios funkcijos, tinkle yra vienas įrenginys, kuris atsakingas už kriptografinių raktų generavimą ir jų paskirstymą. Verta paminėti, kad DTLS ir TLS metoduose, autentikavimas atliekamas komunikacijos sesijos sudarymo metu pagal sertifikatus. Siūlomame sprendime yra pateikiama žinutės autentifikavimo reikšmė (MAC), kuri patikrinama prieš pradedant iššifruoti ar kaip kitaip apdoroti gautus duomenis.

### 4.3.16. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo tyrimo rezultatų išvados

1. Siūlomas sprendimas yra efektyvus ir naudoja mažiau atminties resursų palyginus su DTLS ir TLS metodais, nes naudoja silpnesnę kriptografija. Reikia pabrėžti tai, kad siūlomas sprendimas yra skirtas lokaliam tinklui, kuris neturi komunikacijos ryšio su jokiais išoriniais tinklais. Kadangi nėra ryšio su išoriniu internetu, rizika tinklui yra sumažinama, todėl galima palengvinti naudojamą kriptografiją. Dėl šios priežasties matosi kodėl siūlomo sprendimo energijos sąnaudos ir atminties ištekliai yra mažesni, o sparta laiko atžvilgiu yra efektyvesnė palyginus su DTLS ir TLS metodais. Energijos sąnaudos mažesnės todėl, nes siūlomas sprendimas atlieka mažiau skaičiavimo veiksmų šifruojant ir iššifruojant duomenis, skaičiuojant žinučių autentifikacijos reikšmes ir skaičiuojant naujus kriptografinius sesijos raktus.

2. Sparta laiko atžvilgiu yra efektyvesnė dėl tos pačios priežasties – atliekama mažiau skaičiavimo veiksmų, o visi skaičiavimai atliekami naudojant trumpesnius kriptografinius raktus. Palyginus su DTLS ir TLS, trumpesni kriptografiniai raktai gali papildomai pagreitinti metodo veikimą.

3. Atminties išteklių sąnaudos mažesnės dėl trumpesnio programos kodo, trumpesnių kriptografinių raktų ir kitų trumpesnių saugumo parametrų.

4. Siūlomas sprendimas nėra tinkamas naudoti dinaminiame tinkle. Taip yra todėl, nes kiekvienas įrenginys tinkle turi būti sukonfigūruotas atskirai ir nurodant su kokiais įrenginiais bus atliekama komunikacija. Taip pat, įrenginio konfigūravimo metu būtina nustatyti kriptografinius parametrus ir pagrindinį raktą. Šių parametrų nežinant, papildomo įrenginio pridėti į tinklą negalima. Kitas svarbus aspektas yra tai, kad siūlomas sprendimas nėra tinkamas naudoti tinkluose, kurie turi sąsają su internetu, kadangi siūlomo sprendimo naudojama kriptografija neatitinka saugumo reikalavimų ir nėra efektyvus sprendimas dėl tokių pat priežasčių kaip netikimas dinaminiam tinklui.

## Išvados

Siūlomo sprendimo, pavadinimu „Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodas“, projektavimo, realizacijos ir tyrimo metu iškilusios išvados:

1. Siūlomo sprendimo taikymas yra specializuotas nekintančiam ir trumpo nuotolio tinklui, kuris neturi sąsajos su internetu. Nors pagrindinis sprendimo vystymas buvo skirtas transporto priemonėms, pastebėta, kad siūlomą sprendimą galima taikyti ir kitose įterptinėse sistemose ar daiktų interneto technologijose (IoT – Internet of Things), kuriuose ištekliai yra riboti. Verta paminėti, kad siūlomas sprendimas netinka sistemoms, kurios komunikuoja su internetu, tačiau tokiose sistemose tiktų izoliuotoms posistemėms.

2. Atlikus analizę pastebėta, kad tinklo saugumo problemos kyla dėl autentifikacijos trūkumo ir naudojamos kriptografijos silpnumo. Kadangi įterptinės sistemos dažnai yra ribotų išteklių sistemos, jose neįmanoma įgyvendinti stipriausių ir griežčiausių rinkoje esančių saugumo standartų ir funkcijų. Todėl yra poreikis saugumo metodų, kurie padidintų įterptinių sistemų saugumą. Tačiau verta paminėti, kad tokie metodai visuomet reikalauja sistemos ribojimų, tokių kaip statinė tinklo architektūra, sistemos izoliavimas nuo išorinių tinklų ar įrenginių ir kita.

3. Transporto priemonės lokalaus belaidžio tinklo saugios komunikacijos metodo projektavimo metu pastebėta, kad daug saugumo problemų išsprendžiama izoliavus tinklą nuo išorinių tinklų kaip internetas. Taip pat, pastebėta, kad atliekant kiekvienos žinutės autentifikaciją išvengiama nuskaitytų duomenų įterpimo atakų. Išsaugojus pagrindinį raktą kiekviename sistemos įrenginyje, sutrumpinamas komunikacijos tunelio sudarymas ir išvengiama komunikacijai sudaryti reikalingų parametrų siuntimo tinkle. Pastebėta, kad tokiu metodu tinklo architektūra yra statinė, kiekvienas įrenginys sukonfigūruotas specifiškai komunikuoti tik su nurodytais tinklo įrenginiais sumažinant sistemos atakos paviršių.

4. Siūlomo sprendimo realizacijos metu pastebėta, kad daugiausiai atminties išteklių sąnaudų sudaro ne saugios komunikacijos metodas, o komunikacijos protokolas, kuris yra apsaugomas siūlomu sprendimu. Didėjant skaičiui tinklo įrenginių su kuriais yra komunikuojama, sistemos efektyvumas didėja, kadangi kiekvienai komunikacijai prisideda tik GATT profilis (prototipo realizacijos metu naudojant „Bluetooth“ technologiją) ir kliento įrenginyje saugomas sesijos raktas. Taip pat, sumažėja ir tinklo energijos sąnaudos, kadangi vienas įrenginys gali komunikuoti su daugiau tinklo mazgų vietoje daug įrenginių komunikuojančių su mažai tinklo mazgų.

5. Siūlomo sprendimo tyrimo metu pastebėta, kad transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo energijos sąnaudos yra mažesnės lyginant su DTLS ar TLS metodais. Tai ypač pastebima trumpų komunikacijos sąsajų metu, kada reikia sudaryti naują sesiją apsikeisti duomenimis. Taip pat, pastebėta, kad atminties išteklių sąnaudos yra mažesnės lyginant su DTLS ir TLS metodus. Tačiau, tyrimo metu lyginant spartą laiko atžvilgiu, DTLS metodo žinučių siuntimas truko tiek pat laiko kaip ir siūlomo sprendimo žinučių siuntimas. Verta paminėti, kad DTLS užtruko apytiksliai keturis kartus ilgiau sudaryti komunikacijos sesijai lyginant su siūlomu sprendimu. Jeigu duomenų apsikeitimo sesijos yra trumpos, siūlomo sprendimo spartos laiko atžvilgiu efektyvumas yra didesnis už DTLS ar TLS metodus.

## Literatūros sąrašas


[1] R. W., J. H., T. G. ir Dhasarathy Parthasarathy. *An in-vehicle wireless sensor network for heavy vehicles*. Sweeden: Gothenburg, Volvo Group Trucks Technology, Advanced Technology and Research, 2016.

[2] Z. E.-R., S. S. J. ir P. Ranganathan. *Cybersecurity challenges in vehicular communications*. 2020.

[3] H. Onishi, *Approaches for vehicle cyber security*. 2014.

[4] F. R. Y., F. I., P. Z. ir Xiaoqiang Sun. *A Survey on Cyber-Security of Connected and Autonomous Vehicles (CAVs)*. 2021.

[5] S. K. ir K. Nishchal. *Known-plaintext attack on encryption domain independent optical asymmetric cryptosystem*. 2013.

[6] S. Lucks. *Ciphers Secure against Related-Key Attacks*.

[7] A. B. ir D. Wagner. *Slide Attacks*.

[8] I. D. ir A. Shamir. *Cube Attacks on Tweakable Black Box Polynomials*. 2009.

[9] A. C., C. V., A. S., V. I. ir R. A. Gheorghiu. *Overview of network topologies for V2X communications*. 2017.

[10] K. H., M. A., N. V. ir P. T. O. Olawumi. *Three practical attacks against ZigBee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned*. 2014.

[11] S. A. S., O. H. ir A. R. Madeline Cheah. *Towards a systematic security evaluation of the automotive Bluetooth interface*. 2017.

[12] G. K. ir S. G. C. Kolias. *Attacks and Countermeasures on 802.16: Analysis and Assessment*. 2012.

[13] B. P. Dave Singel´ee. *Security Overview of Bluetooth*. COSIC: 2004.

[14] A. Giousouf. *Bluetooth security*. Bochum: Ruhr University Communication Security Department.

[15] A. D. E. E. ir E. A. E.-W. Mohammed Aly Abdrabou. *LTE Authentication Protocol (EPS-AKA) Weaknesses Solution*. Egyp: Cairo, Dept. of Communication Military Technical College, 2015.

[16] Q. W., X. C., G. Q., Y. L., Z. L. ir Z. Lu. *LEAP: A Lightweight Encryption and Authentication Protocol for In-Vehicle Communications*. 2019.

[17] M. W., N. K., M. K., K. K.-K., R. C. ir Y. P. Ashok Kumar Das. *Design of Secure and Lightweight Authentication Protocol for Wearable Devices Environment*. IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, 2018.

[18] A. N., B. D., B. S. ir Surendran. *A survey of cryptographic algorithms for IoT devices*. 2018.

[19] B. D., M. E.-S. ir A. Aboshosha. *Immunity of Lightweight DES Algorithm (DESL) Against Linear Cryptanalysis Attack*. 2019.

[20] S. P. ir S. Smagin. *Lightweight Cryptography: Underlying Principles and Approaches*. 2011.

[21] R. C., A. A., F. L., d. S. R., M. A., B. C. A., C. B. M. Geovandro ir C. C. F. Pereira. *Performance Evaluation of Cryptographic Algorithms over IoT Platforms and Operating Systems*. 2017.

[22] E. Systems. *Espressif*. 2022. [interaktyvus]. Prieiga per: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Žiūrėta 2022-11-05].

[23] *espressif*. 2023-01. [interaktyvus]. Prieiga per: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [žiūrėta 2023-02-20].

[24] J. L. Sandeep Kamath. *Measuring Bluetooth Low Energy Power Consumption*. 2012. [interaktyvus]. Prieiga per: https://discourse-production.oss-cn-shanghai.aliyuncs.com/original/3X/7/c/7c84d1dc683e86d61f4db95f90223453fc25861f.pdf. [žiūrėta 2023-04-15].

[25] S. Frankel. *rfc-editor*. 2013-09. [interaktyvus]. Prieiga per: https://www.rfc-editor.org/rfc/rfc3602. [žiūrėta 2023-04-21].

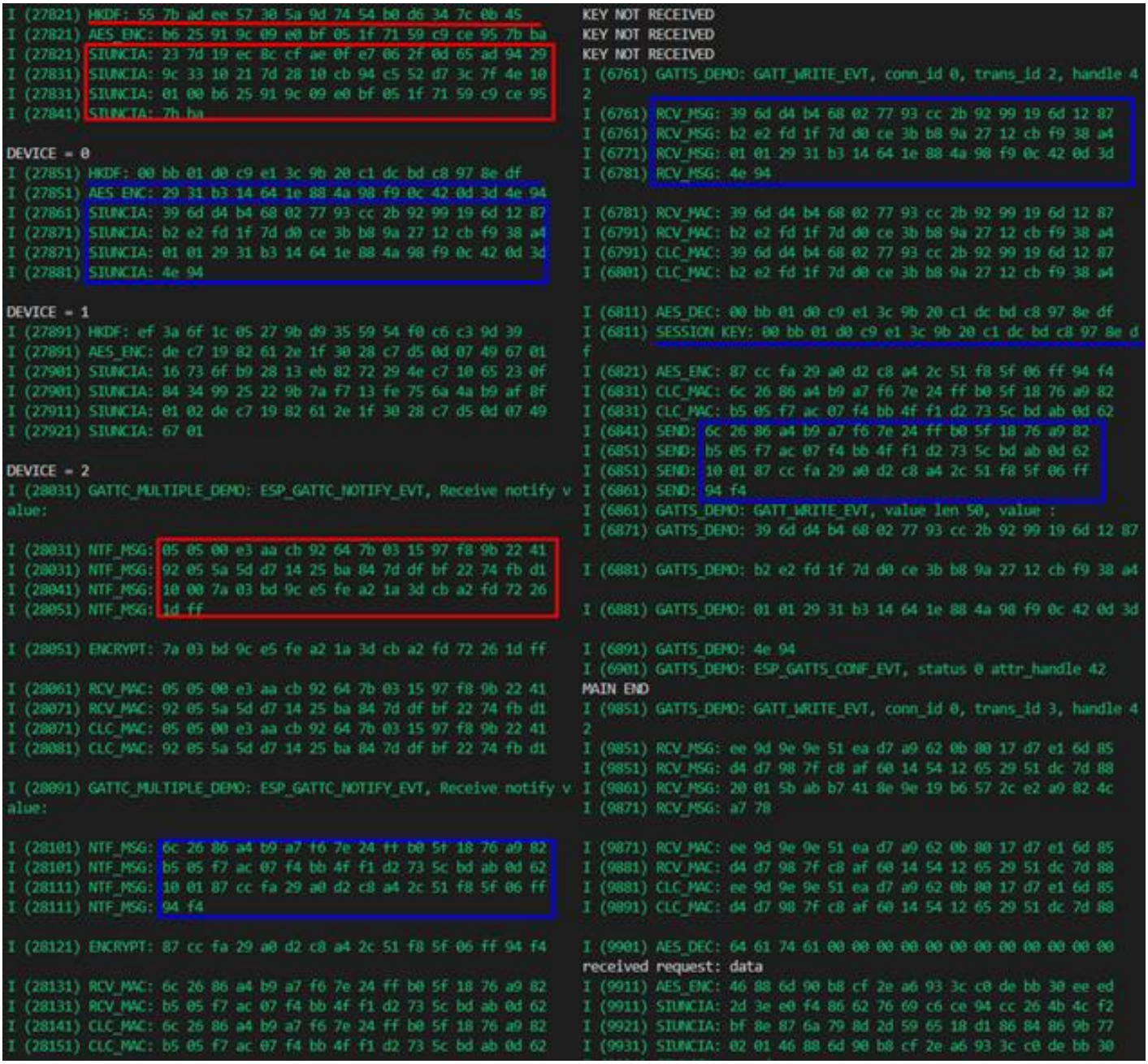
[26] H. Krawczyk. *rfc-editor*. 2010-05. [interaktyvus]. Prieiga per: https://www.rfc-editor.org/rfc/rfc5869. [žiūrėta 2023-04-19].

[27] *Hackcontrol*. [interaktyvus]. Prieiga per: https://hackcontrol.org/blog/what-is-dtls-and-how-is-it-used/. [žiūrėta 2023-04-20].

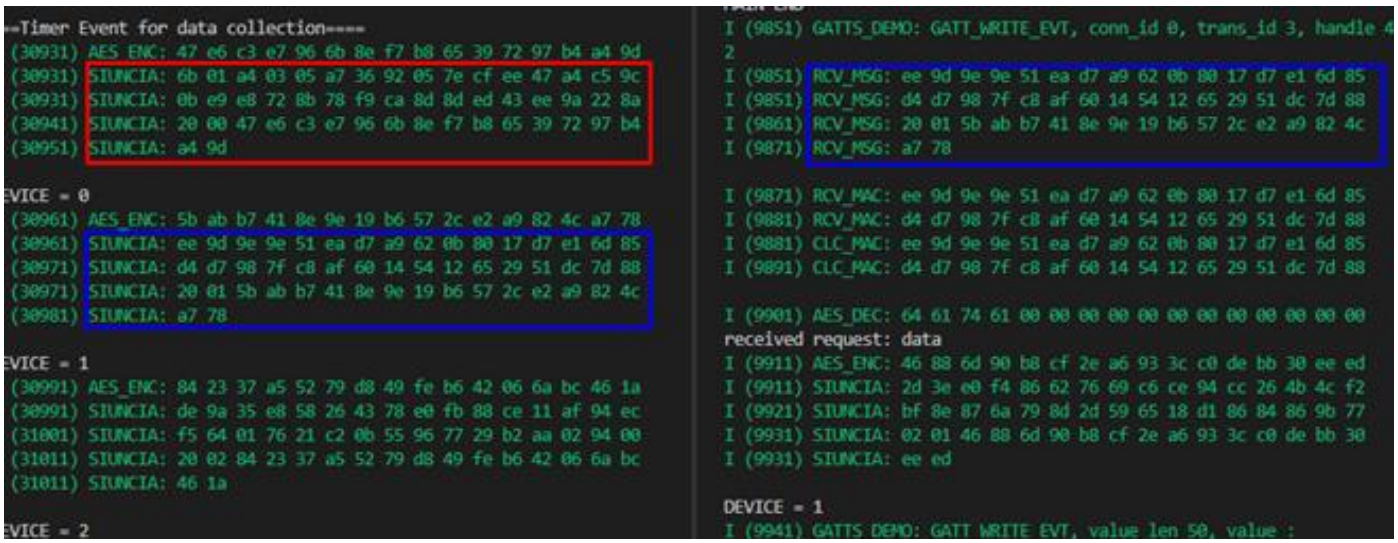
[28] Baeldung, *baeldung*. 2022-11-04. [interaktyvus]. Prieiga per: https://www.baeldung.com/cs/tls-vs-dtls. [žiūrėta 2023-04-20].

## Priedai

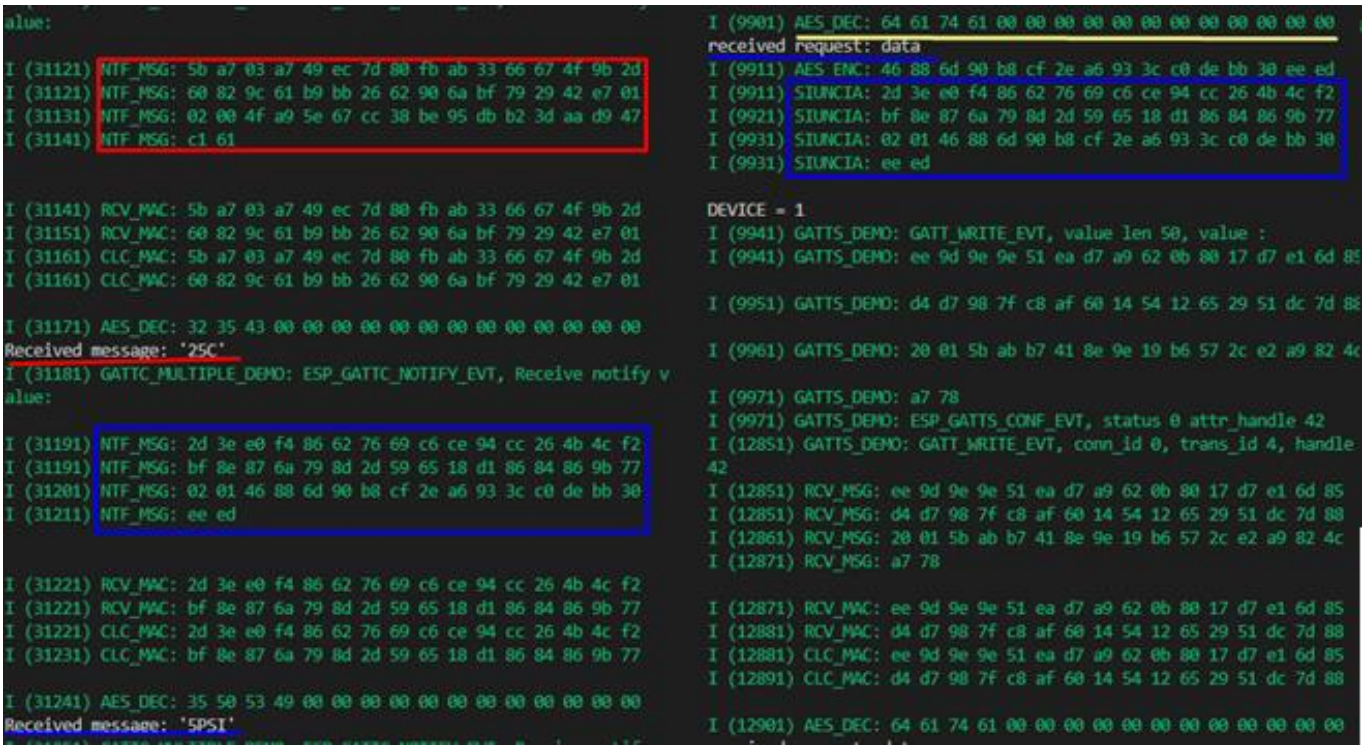
### 1 Priedas. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo prototipo rezultatai spausdinamai terminale



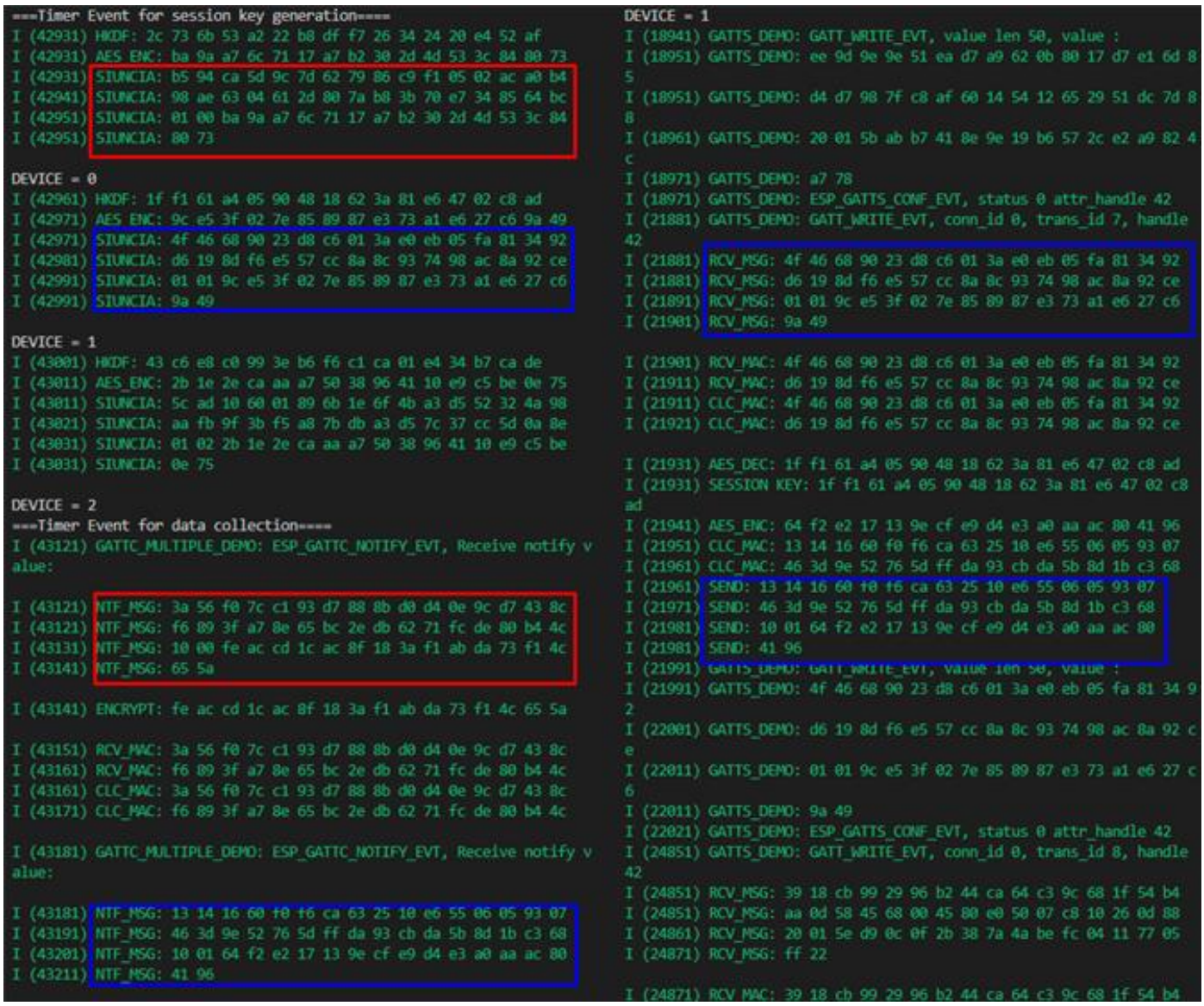

**51 pav.** Pagrindinio valdiklio raktų generavimas ir paskirstymas antram jutiklių posistemės valdikliui




**52 pav.** Pagrindinio valdiklio duomenų užklausos antram jutiklių posistemės valdikliui




**53 pav.** Pagrindinio valdiklio duomenų užklausų atsakymo žinutė iš antrojo posistemių valdiklio




**54 pav.** Pagrindinio valdiklio periodinis kriptografinių sesijos raktų generavimo laiko įvykis – antro įrenginio duomenys

## 2 Priedas. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo, DTLS ir TLS kliento įrenginio energijos sąnaudų rodmenys

**13 lentelė.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo DTLS ir TLS kliento įrenginio energijos sąnaudų rodmenys

| Laikas (minutės ir sekundės) | Siūlomas sprendimas energijos sąnaudos (mA) | DTLS energijos sąnaudos (mA) | TLS energijos sąnaudos (mA) |
|---|---|---|---|
| 00:00.0 | 50.06 | 48.72 | 48.15 |
| 00:00.6 | 50.06 | 128.4 | 113 |
| 00:01.2 | 103.4 | 126.4 | 111.1 |
| 00:01.8 | 94.2 | 125.1 | 111.1 |
| 00:02.4 | 69.5 | 125.2 | 123 |
| 00:03.1 | 74.9 | 148.4 | 125 |
| 00:03.7 | 64.6 | 148 | 113.1 |

| | | | |
|---|---|---|---|
| 00:04.3 | 66.7 | 78.1 | 111.8 |
| 00:05.0 | 64.3 | 78.1 | 111.4 |
| 00:05.5 | 64.1 | 129.8 | 111.4 |
| 00:06.1 | 65.8 | 79.5 | 111.5 |
| 00:06.7 | 66 | 67.9 | 112.7 |
| 00:07.3 | 63.8 | 113.9 | 111.2 |
| 00:08.0 | 66.3 | 113.9 | 111.3 |
| 00:08.6 | 62.5 | 56.2 | 119.8 |
| 00:09.2 | 63.7 | 124.1 | 124.9 |
| 00:09.8 | 62.7 | 124.1 | 124.8 |
| 00:10.4 | 64.4 | 55.56 | 97.4 |
| 00:11.0 | 62.6 | 55.52 | 89.9 |
| 00:11.5 | 66 | 55.52 | 89.9 |
| 00:12.1 | 66.7 | 55.3 | 89.9 |
| 00:12.7 | 64 | 56.4 | 74.6 |
| 00:13.2 | 67.1 | 55.17 | 57.7 |
| 00:13.9 | 63 | 55.17 | 57.6 |
| 00:14.5 | 63 | 59.96 | 57.6 |
| 00:15.1 | 65.6 | 55.47 | 43.57 |
| 00:15.7 | 66.1 | 55.49 | 39.71 |
| 00:16.3 | 64.9 | 55.49 | 43.9 |
| 00:16.9 | 66 | 55.48 | 43.9 |
| 00:17.5 | 63.5 | 55.49 | 43.96 |
| 00:18.1 | 68.5 | 55.49 | 43.92 |
| 00:18.7 | 63.5 | 57.18 | 43.92 |
| 00:19.3 | 62.8 | 55.51 | 56.92 |
| 00:19.9 | 65.8 | 55.51 | 56.97 |
| 00:20.6 | 65.8 | 55.57 | 57.72 |
| 00:21.2 | 64.2 | 55.51 | 92.9 |
| 00:21.8 | 65.8 | 55.51 | 57.6 |
| 00:22.5 | 68.8 | 55.1 | 57.6 |
| 00:23.1 | 63.9 | 55.53 | 43.93 |
| 00:23.7 | 66.2 | 55.08 | 47.59 |
| 00:24.3 | 62.5 | 55.08 | 57.5 |
| 00:24.9 | 65.7 | 55.54 | 57.6 |
| 00:25.5 | 60.2 | 55.53 | 89.5 |
| 00:26.1 | 66.3 | 55.53 | 54.66 |
| 00:26.7 | 64.2 | 57.4 | 54.66 |
| 00:27.3 | 66.3 | 55.54 | 43.92 |
| 00:27.9 | 64.9 | 55.06 | 43.5 |

| | | | |
|---|---|---|---|
| 00:28.4 | 65 | 55.52 | 48.19 |
| 00:29.0 | 62.3 | 55.24 | 57.7 |
| 00:29.6 | 64.9 | 55.52 | 57.8 |
| 00:30.2 | 65.2 | 55.03 | 73.5 |
| 00:30.9 | 64.8 | 55.52 | 93.8 |
| 00:31.4 | 65.4 | 55.52 | 49.13 |
| 00:32.0 | 63.2 | 55.53 | 45.42 |
| 00:32.7 | 63.2 | 55.53 | 46.42 |
| 00:33.3 | 62.6 | 55.23 | 57.72 |
| 00:33.9 | 65.6 | 55.51 | 57.38 |
| 00:34.5 | 64.4 | 55.51 | 59.94 |
| 00:35.1 | 64.4 | 55.61 | 59.94 |
| 00:35.8 | 64.6 | 55.61 | 82.3 |
| 00:36.4 | 66.6 | 55.45 | 43.96 |
| 00:37.0 | 66.7 | 56.6 | 43.94 |
| 00:37.7 | 62.6 | 56.6 | 43.93 |
| 00:38.2 | 60.6 | 55.52 | 57.8 |
| 00:38.8 | 66.1 | 55.52 | 57.6 |
| 00:39.4 | 64.5 | 55.2 | 57.6 |
| 00:40.0 | 64.7 | 55.53 | 57.6 |
| 00:40.6 | 66.2 | 55.52 | 43.99 |
| 00:41.3 | 65.6 | 55.52 | 43.92 |
| 00:41.9 | 66.2 | 55.53 | 43.48 |
| 00:42.5 | 64.5 | 55.55 | 57.2 |
| 00:43.1 | 66.3 | 56.23 | 57.2 |
| 00:43.7 | 69 | 57.26 | 57.79 |
| 00:44.3 | 62.9 | 55.12 | 83.9 |
| 00:44.9 | 65.1 | 55.66 | 83.9 |
| 00:45.4 | 64.9 | 55.66 | 43.96 |
| 00:46.1 | 64 | 55.65 | 42.52 |
| 00:46.7 | 62.9 | 59.74 | 72.3 |
| 00:47.3 | 66.1 | 55.11 | 57.3 |
| 00:47.9 | 64.4 | 55.58 | 64.1 |
| 00:48.5 | 65.8 | 55.58 | 91.7 |
| 00:49.1 | 68.1 | 56.16 | 44.01 |
| 00:49.7 | 64.5 | 56.16 | 43.52 |
| 00:50.3 | 62.2 | 55.57 | |
| 00:50.9 | 65.6 | 55.57 | |
| 00:51.5 | 70.9 | 58.08 | |
| 00:52.1 | 63.7 | 55.3 | |

| | | | |
|---|---|---|---|
| 00:52.7 | 64.1 | 56.25 | |
| 00:53.2 | 65.8 | 55.57 | |
| 00:53.9 | 65.9 | 55.57 | |
| 00:54.5 | 66.2 | 55.23 | |
| 00:55.1 | 64.8 | 55.17 | |
| 00:55.8 | 66.3 | 50.92 | |
| 00:56.3 | 61.8 | 59.8 | |
| 00:56.9 | 68.1 | 55.61 | |
| 00:57.5 | 64.7 | 55.49 | |
| 00:58.1 | 66 | 55.2 | |
| 00:58.7 | 64.7 | 55.6 | |
| 00:59.3 | 62.5 | 55.62 | |
| 00:59.9 | 64.2 | 55.63 | |
| 01:00.5 | 65.2 | 59.6 | |
| 01:01.1 | 65.5 | 59.6 | |
| 01:01.7 | 64.7 | 55.63 | |
| 01:02.4 | 63.7 | 55.63 | |
| 01:03.0 | 66.2 | 55.21 | |
| 01:03.5 | 66.3 | 55.23 | |
| 01:04.1 | 69 | 55.61 | |
| 01:04.7 | 67.1 | 55.63 | |
| 01:05.3 | 63.4 | 55.64 | |
| 01:05.9 | 66.2 | 55.61 | |
| 01:06.5 | 64.1 | 55.61 | |
| 01:07.2 | 65.9 | 55.14 | |
| 01:07.8 | 64.1 | 55.62 | |
| 01:08.4 | 64.4 | 55.62 | |
| 01:09.0 | 65.2 | 55.5 | |
| 01:09.6 | 63.3 | 52.21 | |
| 01:10.2 | 62.4 | 55.68 | |
| 01:10.8 | 66.3 | 55.66 | |
| 01:11.4 | 62.7 | 55.66 | |
| 01:12.0 | 62.9 | 55.64 | |
| 01:12.5 | 66.3 | 51.33 | |
| 01:13.2 | 64.1 | 59.3 | |
| 01:13.7 | 66.1 | 59.3 | |
| 01:14.3 | 68.6 | 55.6 | |
| 01:15.0 | 64.7 | 58.39 | |
| 01:15.6 | 66.7 | 55.67 | |
| 01:16.1 | 64.8 | 55.67 | |

| | | | |
|---|---|---|---|
| 01:16.7 | 65.9 | 55.18 | |
| 01:17.3 | 65.5 | 55.18 | |
| 01:17.9 | 66 | 55.66 | |
| 01:18.5 | 62.5 | 55.48 | |
| 01:19.0 | 72.3 | 55.66 | |
| 01:19.6 | 62.8 | 55.52 | |
| 01:20.1 | 64.5 | 52.26 | |
| 01:20.7 | 63.2 | 55.67 | |
| 01:21.4 | 63.2 | 55.68 | |
| 01:22.0 | | 60.2 | |
| 01:22.6 | | 55.3 | |
| 01:23.2 | | 55.66 | |
| 01:23.7 | | 55.67 | |
| 01:24.3 | | 55.23 | |
| 01:24.9 | | 59.83 | |
| 01:25.4 | | 55.32 | |
| 01:26.0 | | 55.68 | |
| 01:26.5 | | 58.95 | |
| 01:27.1 | | 55.39 | |
| 01:27.8 | | 55.67 | |
| 01:28.4 | | 55.67 | |
| 01:29.0 | | 55.68 | |
| 01:29.6 | | 55.93 | |
| 01:30.2 | | 55.7 | |
| 01:30.8 | | 55.19 | |
| 01:31.4 | | 55.64 | |
| 01:32.0 | | 55.44 | |
| 01:32.6 | | 55.44 | |
| 01:33.2 | | 55.66 | |
| 01:33.8 | | 55.69 | |
| 01:34.4 | | 55.69 | |
| 01:34.9 | | 55.34 | |
| 01:35.5 | | 55.71 | |
| 01:36.2 | | 116.4 | |
| 01:36.9 | | 55.7 | |
| 01:37.4 | | 56.99 | |

## 3 Priedas. Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo, DTLS ir TLS serverio įrenginio energijos sąnaudų rodmenys

**14 lentelė.** Transporto priemonių lokalaus belaidžio tinklo saugios komunikacijos metodo, DTLS ir TLS serverio įrenginio energijos sąnaudų rodmenys

| Laikas (minutės ir sekundės) | Siūlomas sprendimas energijos sąnaudos (mA) | DTLS energijos sąnaudos (mA) | TLS energijos sąnaudos (mA) |
|---|---|---|---|
| 00:00.0 | 18.1 | 49.82 | 51.18 |
| 00:00.6 | 48.12 | 130.3 | 129.9 |
| 00:01.2 | 102 | 129.9 | 127.3 |
| 00:01.8 | 102 | 127.6 | 127.4 |
| 00:02.5 | 56.3 | 127.6 | 152 |
| 00:03.0 | 57.3 | 152.1 | 146.2 |
| 00:03.6 | 51.94 | 129.8 | 131.6 |
| 00:04.2 | 52.26 | 130.2 | 128 |
| 00:04.8 | 52.4 | 128.1 | 127.9 |
| 00:05.4 | 51.85 | 128.1 | 128 |
| 00:06.0 | 52.44 | 129.3 | 130.7 |
| 00:06.6 | 51.76 | 129.6 | 128.5 |
| 00:07.2 | 52.46 | 127.5 | 128.4 |
| 00:07.7 | 57.29 | 127.7 | 128.2 |
| 00:08.3 | 51.72 | 152.4 | 152.1 |
| 00:09.0 | 52.39 | 152.3 | 108.2 |
| 00:09.6 | 51.81 | 134 | 108.3 |
| 00:10.2 | 52.37 | 67 | 115.9 |
| 00:10.7 | 51.71 | 56.7 | 56.3 |
| 00:12.0 | 52.5 | 139.3 | 56.8 |
| 00:12.6 | 52.09 | 87.3 | 56.8 |
| 00:13.2 | 52.4 | 76.6 | 56.7 |
| 00:13.8 | 51.99 | 56.5 | 113.6 |
| 00:14.4 | 52.02 | 56.5 | 56.6 |
| 00:15.0 | 52.45 | 56.7 | 60.3 |
| 00:15.7 | 51.85 | 56.6 | 56.6 |
| 00:16.3 | 52.03 | 56.78 | 62.8 |
| 00:16.9 | 52.24 | 56.43 | 62.8 |
| 00:17.5 | 51.86 | 56.35 | 56.6 |
| 00:18.0 | 52.31 | 56.83 | 56.6 |
| 00:18.6 | 51.96 | 56.77 | 56.65 |
| 00:19.3 | 52.36 | 56.6 | 56.85 |
| 00:19.8 | 52.27 | 56.5 | 56.76 |
| 00:20.4 | 52.44 | 56.4 | 56.76 |

| | | | |
|---|---|---|---|
| 00:21.0 | 50.36 | 71.1 | 56.35 |
| 00:21.6 | 50.77 | 71.1 | 56.85 |
| 00:22.2 | 50.78 | 57.24 | 56.85 |
| 00:22.8 | 50.67 | 56.2 | 56.3 |
| 00:23.4 | 50.92 | 56.2 | 56.4 |
| 00:24.0 | 56.28 | 56.56 | 56.5 |
| 00:24.5 | 50.54 | 56.76 | 56.5 |
| 00:25.1 | 50.72 | 56.76 | 56.85 |
| 00:25.7 | 50.92 | 53.61 | 56.7 |
| 00:26.3 | 50.69 | 56.5 | 56.7 |
| 00:26.9 | 50.95 | 56.5 | 56.85 |
| 00:27.5 | 50.27 | 56.6 | 56.38 |
| 00:28.1 | 50.94 | 56.5 | 56.95 |
| 00:28.7 | 50.93 | 56.5 | 56.36 |
| 00:30.0 | 50.65 | 58.9 | 56.83 |
| 00:30.5 | 50.9 | 58.8 | 58.7 |
| 00:31.1 | 50.43 | 56.2 | 56.79 |
| 00:31.7 | 50.89 | 74.2 | 56.84 |
| 00:32.3 | 50.58 | 56.5 | 58.03 |
| 00:32.9 | 50.63 | 56.4 | 56.88 |
| 00:33.5 | 50.65 | 56.43 | 56.85 |
| 00:34.2 | 50.83 | 56.43 | 56.72 |
| 00:34.8 | 51.04 | 56.6 | 58.44 |
| 00:35.4 | 50.53 | 56.5 | 56.87 |
| 00:36.1 | 50.9 | 56.5 | 56.85 |
| 00:36.6 | 50.5 | 56.6 | 56.85 |
| 00:37.2 | 50.97 | 56.6 | 56.8 |
| 00:37.9 | 50.52 | 56.6 | 56.8 |
| 00:38.5 | 51.04 | 56.6 | 74.1 |
| 00:39.0 | 51.04 | 52.09 | 56.7 |
| 00:39.6 | 50.67 | 56.79 | 61.1 |
| 00:40.2 | 51.89 | 56.58 | 56.6 |
| 00:40.8 | 50.6 | 56.58 | 56.7 |
| 00:41.3 | 51.23 | 64.4 | 56.6 |
| 00:41.9 | 50.66 | 57.1 | 56.6 |
| 00:42.5 | 51.24 | 60.7 | 56.72 |
| 00:43.1 | 50.7 | 56.2 | 56.84 |
| 00:43.6 | 51.14 | 56.4 | 56.88 |
| 00:44.2 | 51.54 | 56.5 | 56.37 |
| 00:44.8 | 52.53 | 80.8 | 56.48 |

| | | | |
|---|---|---|---|
| 00:45.3 | 52.15 | 80.8 | 56.87 |
| 00:45.9 | 52.58 | 56.88 | 58.29 |
| 00:46.5 | 52.55 | 56.88 | 56.47 |
| 00:47.2 | 52.22 | 56.6 | 56.9 |
| 00:47.8 | 52.4 | 56.6 | 56.86 |
| 00:48.3 | 51.85 | 56.4 | 59.06 |
| 00:48.9 | 52.53 | 78.6 | 56.51 |
| 00:49.5 | 52.18 | 78.6 | 56.96 |
| 00:50.1 | 52.22 | 56.87 | 101.9 |
| 00:50.7 | 52.58 | 56.86 | 56.6 |
| 00:51.3 | 52.17 | 56.58 | |
| 00:51.9 | 52.57 | 56.88 | |
| 00:52.5 | 51.98 | 56.67 | |
| 00:53.1 | 52.09 | 56.42 | |
| 00:53.7 | 52.43 | 56.83 | |
| 00:54.3 | 51.97 | 61 | |
| 00:55.0 | 52.61 | 56.7 | |
| 00:55.5 | 51.86 | 95.4 | |
| 00:56.1 | 52.45 | 56.7 | |
| 00:56.7 | 52.47 | 56.5 | |
| 00:57.3 | 52.46 | 101.2 | |
| 00:57.9 | 52.45 | 56.7 | |
| 00:58.5 | 51.88 | 56.4 | |
| 00:59.2 | 52.58 | 56.5 | |
| 00:59.8 | 52.08 | 56.4 | |
| 01:00.4 | 52.57 | 56.4 | |
| 01:01.0 | 51.99 | 56.8 | |
| 01:01.6 | 52.54 | 101.5 | |
| 01:02.3 | 52.43 | 59.7 | |
| 01:02.9 | 52.53 | 59.7 | |
| 01:03.5 | 52.2 | 56.86 | |
| 01:04.1 | 52.14 | 56.86 | |
| 01:04.7 | 52.36 | 56.7 | |
| 01:05.3 | 51.76 | 56.6 | |
| 01:05.8 | 52.49 | 103.4 | |
| 01:06.4 | 52.24 | 56.5 | |
| 01:07.1 | 52.12 | 56.5 | |
| 01:07.7 | 52.56 | 56.5 | |
| 01:08.3 | 51.97 | 56.7 | |
| 01:08.9 | 52.44 | 56.7 | |

| | | | |
|---|---|---|---|
| 01:09.5 | 51.97 | 56.93 | |
| 01:10.1 | 52.57 | 56.91 | |
| 01:10.6 | 52.08 | 56.7 | |
| 01:11.2 | 52.56 | 56.5 | |
| 01:11.8 | | 88.5 | |
| 01:12.4 | | 103.5 | |