Kauno technologijos universitetas

Informatikos fakultetas

Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo sudarymas ir tyrimas

Baigiamasis magistro studijų projektas

Tadas Malinauskas

Projekto autorius

Prof. Algimantas Venčkauskas

Vadovas

Kaunas, 2020

Kauno technologijos universitetas

Informatikos fakultetas

Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo sudarymas ir tyrimas

Baigiamasis magistro studijų projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Tadas Malinauskas

Projekto autorius

Prof. Algimantas Venčkauskas

Vadovas

Doc. Gedeiminas Činčikas

Recenzentas

Kaunas, 2020

Kauno technologijos universitetas

Informatikos fakultetas

Tadas Malinauskas

Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo sudarymas ir tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Tado Malinauskas, baigiamasis projektas tema „Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo sudarymas ir tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)	(parašas)

Malinauskas, Tadas. Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo sudarymas ir tyrimas. Magistro studijų baigiamasis projektas / vadovas prof. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos inžinerija, informatikos mokslai.

Reikšminiai žodžiai: autonominis, robotas, aplinka, stebėjimas, saugos, komunikavimo, metodas, saugumas, vaizdo, transliacija, kriptografija, ribotų, pajėgumų, įrenginiai

Kaunas, 2020. 85 p.

Santrauka

Autonominių robotų rinkai stipriai plečiantis, šiuos, dažniausiai, savarankiškai veikiančius įrenginius galima pamatyti visur: namuose, karo, medicinos, pramonės, gynybos sektoriuose. Autonominiai robotų atliekamos funkcijos įvairios, pradedant nuo atliekamų operacijų medicinos sektoriuje, tęsiant aplinkos stebėjimu namuose, baigiant kariniais veiksmais gynybos sektoriuje. Autonominiai aplinkos stebėjimo robotai įprastai veikia aplinkose, kur apstu privačių ir konfidencialių duomenų, todėl būtina susirūpinti tokių duomenų saugiu perdavimu ir robotų kibernetiniu saugumu. Aplinkos pokyčių duomenis siunčiant galiniam įrenginiui, privalu, jog duomenų komunikavimo terpė būtų saugi, o duomenys apsaugoti nuo galimo pasisavinimo, modifikavimo ar jų kilmės suklastojimo.

Atsižvelgus į staigų autonominių robotų rinkos augimą ir prognozes, kurios rodo, jog ateityje didžiąją dalį interneto duomenų sudarys multimedijos duomenys, pagrindinis baigiamojo darbo objektas yra autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo realizavimas. Apžvelgus esamus komunikavimo metodus, jų privalumus ir trūkumus, iškeltas darbo tikslas pasiūlyti veiksmingą, efektyvų saugų komunikavimo metodą, kuris būtų pritaikomas ribotus skaičiuojamosios galios ir energijos išteklius turinčiam autonominiam aplinkos stebėjimo robotui.

Metodas turi tenkinti iškeltus vaizdo perdavimo operacijos reikalavimus: užtikrinti gerą vaizdo kokybę, pasižymėti greitaveika, mažu vaizdo vėlavimu ir taupiu energijos suvartojimu. Atsižvelgus į aplinkos stebėjimo roboto atliekamą funkciją, veikimo terpę, apibrėžiamas reikiamas, neperteklinis saugumo lygis. Prototipe pasiūlytas ir realizuotas saugaus komunikavimo metodas: vaizdo perdavimas modifikuotu SRTP protokolu, naudojant iš anksto pasidalinto rakto technika. Eksperimentinėje dalyje komunikavimo metodo veikimo parametrai lyginami komunikavimui naudojant nesaugų RTP ir SRTP-DTLS protokolų porą. Eksperimentinėje dalyje gauti darbo tyrimo rezultatai yra palyginami ir aptariami, pateikiamos darbo išvados.

Malinauskas, Tadas. Development and Research of Secure Communication Method for Autonomous Environmental Monitoring Robot. Master's Final Degree Project / supervisor prof. Algimantas Venčkauskas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics engineering, Computing

Keywords: autonomous, robot, environmental, monitoring, secure, communication, method, video, streaming, cryptography, resource, constrained, devices.

Kaunas, 2020. 85 p.

Summary

With the strong expansion of the market for autonomous robots, these, mostly, self-acting devices can be seen everywhere: in the home, military, medical, industrial and defense sectors. Autonomous functions performed by robots range from operations in the medical sector, to environmental monitoring at home and finishing with military actions in the defense sector. Autonomous environmental monitoring robots typically operate in environments which are rich in private and confidential data, regarding this, there is a need to be concerned about the secure transmission of such data and the cyber security of robots itself. When sending data to the end device, the data communication channel must be secure, and the data must be protected from possible misappropriation, modification or falsification of their origin.

Taking into account the rapid growth of the autonomous robot market and forecasts that show that in the future the bulk of Internet data will be mostly multimedia data, the main object of this final work is the implementation of an autonomous environmental monitoring robot secure communication method. After reviewing the existing communication methods, comparing their advantages and disadvantages, the aim of this is work is to propose an efficient, effective secure communication method that can be adapted to an autonomous environmental monitoring robot with limited computing power and constrained energy resources.

The method must meet the requirements of the video transmission operation: to ensure good video quality, high operational speed, low video delay and energy saving. Depending upon the function performed by the autonomous environmental monitoring robot, operating environment, the required, non-redundant level of security is defined. In the prototype developing phase, it was proposed to implement a secure communication method to transmit video data using modified SRTP protocol with pre-shared key technique. In the experimental part, the performance parameters of the proposed secure communication method are compared against other communication methods which are implemented using an insecure RTP and SRTP-DTLS protocol stacks. The results of the research obtained in the experimental part are compared, discussed and the final conclusions of the work are presented.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Santrumpos:

Wi-Fi – belaidis vietinis tinklas

CAN – komunikavimo metodas, magistralė, skirta mikrokontroleriams ir kitiems įrenginiams bendrauti be pagrindinio (angl. *Host*) kompiuterio (angl. *Controller Area Network*)

OBD-II – automobiliuose naudojama jungtis (angl. *On-board diagnostics*)

TCP – duomenų transportavimo protokolas (angl. *Transmission Control Protocol*)

RTSP – realaus laiko transliavimo protokolas (angl. *Real Time Streaming Protocol*)

WEP – apsaugos algoritmas, naudojamas bevieliuose tinkluose (angl. *Wired Equivalent Privacy*)

WPA – apsaugos algoritmas, programa (angl. *Wi-Fi Protected Access*)

M2M – tiesioginis komunikavimas tarp dviejų įrenginių, įrenginys-irenginiui principu (angl. *Machine to Machine*)

ROS – robotų operacinė sistema (angl. *Robot Operating System*)

3G – trečios kartos bevielė mobiliųjų komunikavimo technologija (angl. *Third Generation*)

LTE – telekomunikacijų standartas bevieliams ryšiams (angl. *Long Term Evolution*)

WLAN – belaidis, vietinis tinklas (angl. *Wireless LAN*)

IP – interneto protokolas (angl. *Internet Protocol*)

IoT – daiktų internetas (angl. *Internet of Things*)

QoS – paslaugų kokybės matavimo gairės (angl. *Quality of Service*)

UDP – duomenų transportavimo protokolas (angl. *User Datagram Protocol*)

RTP – realaus laiko transportavimo protokolas (angl. *Real-time Transporting Protocol*)

RTCP – realaus laiko kontrolės protokolas (angl. *Real-time Control Protocol*)

RSVP – resursų rezervavimo protokolas (angl. *Resource Reservation Protocol*)

HTTP – hipertekstų persiuntimo protokolas (angl. *HyperText Transfer Protocol*)

DES – duomenų šifravimo algoritmas, naudojant simetrinius raktus (angl. *Data Encryption Standard*)

CBC – šifro blokų grandinių šifravimo režimas (angl. *Cipher Block Chaining*)

SRTP – Saugus realaus laiko transliavimo protokolas (angl. *Secure Real-time Transport Protocol*)

AES – Nacionalinio standartų ir technologijos universiteto (NIST) sukurtas šifravimo algoritmas (angl. *Advanced Encryption Standard*)

3DES – trigubas duomenų šifravimo algoritmas (angl. *Triple Data Encryption Standard*)

DASH – adaptyvus transliavimo metodas, aukštos kokybės multimedijos duomenims perduoti internetu (angl. *Dynamic Adaptive Streaming over HTTP*)

MPD – multimedijos duomenų transliavimo aprašymas (angl. *Media presentation description*)

MP4, MPEG-4 – vaizdo ir garso duomenų konteineriavimo technologija (angl. *Moving Picture Experts Group 4*)

MPEG-TS – vaizdo ir garso duomenų konteineriavimo technologija (angl. *MPEG transport stream*)

H.265 – vaizdo pakavimo standartas (angl. *High Efficiency Video Coding*)

PKCS7 – viešųjų raktų kriptografijos standartas (angl. *Cryptographic Message Syntax Standard*)

URI – charakterių seka, aprašanti resursą (angl. *Uniform Resource Identifier*)

TLS – transport lygmens saugumo protokolas (angl. *Transport Layer Security*)

HTTPS – saugus hipertekstų persiuntimo protokolas (angl. *Secure HyperText Transfer Protocol*)

AIMD – atsiliepimų algoritmas, naudojamas TCP perkrovos kontrolės mechanizmui (angl. *Additive-increase/multiplicative-decrease*)

RSA – viešųjų raktų infrastruktūros kriptosistema (angl. *Rivest–Shamir–Adleman*)

ECDSA – eliptinių kreivių šifravimo algoritmas (angl. *Elliptic Curve Digital Signature Algorithm*)

HMAC-SHA-256 – maišos funkcijų reikšmės autentifikavimo kodas, naudojantis SHA-256 maišos funkciją (angl. *Hash-based Message Authentication Code*)

X.509 – kriptografijos standartas, paremtas viešųjų raktų sertifikatų infrastruktūra

DTLS – saugumo protokolas, skirtas apsaugoti UDP duomenis (angl. *Datagram Transport Layer Security*)

TPM – tarptautinis standartas, aprašantis saugų mikroprocesorių (angl. *Trusted Platform Module*)

ECDH – raktų apsikeitimo algoritmas (angl. *Elliptic-curve Diffie–Hellman*)

ECC – eliptinių kreivių kriptografija (angl. *Eliptic Curve Cryptography*)

IKE – protokolas naudojamas IPsec operacijoms (angl. *Internet Key Exchange*)

DH – Diffi Hellmano raktų apsikeitimo mechanizmas (angl. *Diffie–Hellman*)

PKI – viešųjų raktų infrastruktūra (angl. *Public Key Infrastructure*)

FPS – kadrai per sekundę (angl. *Frame Per Second*)

720P – HD kokybę apibrėžiančios vaizdo parametrų gairės

MITM – susidūrimo viduryje ataka (angl. *Man In The Middle*)

AASR – autonominis aplinkos stebėjimo robotas

RC4 – kriptografinis šifras, algoritmas (angl. *Rivest Cipher 4*)

XOR – kriptografinis šifras, algoritmas (angl. *Exclusive Disjunction (XOR) Operation*)

OSI – struktūrinis modelis, skirtas aprašyti telekomunikacinių ar kompiuterinių sistemų architektūras (angl. *Open Systems Interconnection*)

MKI – pagrindinio rakto identifikatorius, naudojamas SRTP protokolo (angl. *Master Key Identifier*)

MAC – pranešimo autentifikavimo žyma (angl. *Message Authentication Code*)

ID - identifikatorius

CSRC – dalyvaujančių šaltinių identifikatorius (angl. *Contributing Sources Identifier*)

AES-CM – AES šifravimo algoritmas, naudojantis kontrolinį režimą (angl. *Advanced Encryption Standard Counter Mode*)

SIP – sesijos inicijavimo protokolas (angl. *Session Initiation Protocol*)

SAP – sesijos pranešimo protokolas (angl. *Session Announcement Protocol*)

VoIP – internet telefonija (angl. *Voice Over IP*)

IPsec – saugus interneto protokolas (angl. *Internet Protocol Security*)

S/MIME – kriptografijos standartas, MIME duomenims (angl. *Secure/Multipurpose Internet Mail Extensions*)

DDoS – paskirstyta sutrikdymo ataka (angl. *Distributed Denial of Service*)

BPMN – verslo veiklos procesų modeliavimo kalba (angl. *Business Process Model and Notation*)

OS – operacinė sistema (angl. *Operating System*)

H.264 – vaizdo pakavimo standartas (angl. *Advanced Video Coding*)

SSRC – sinchronizacijos šaltinio identifikatorius (angl. *Synchronization Source Identifier*)

HMAC-SHA-80 – maišos funkcijų reikšmės autentifikavimo kodas, naudojantis SHA-80 maišos funkciją (angl. *Hash-based Message Authentication Code*)

CPU – procesorius (angl. *Central Processing Unit*)

PIR – pasyvus infraraudonųjų spindulių jutiklis (angl. *Passive Infrared Sensor*)

UML – programinės įrangos inžinerijoje vartojama modeliavimo kalba artefaktams specifikuoti, projektuoti, vizualizuoti, konstruoti ir dokumentuoti kuriant programinės įrangos sistemas. (angl. *Unified Modeling Language*)

RAM – operatyvioji atmintis (angl. *Random-access memory*)

MIPI CSI – kameros jungtis (angl. *Camera Serial Interface of the Mobile Industry Processor Interface*)

GPIO – įvesties ir išvesties kontaktų grandinė (angl. *General-purpose input/output*)

VDC – voltai nuolatinės srovės įtampoje (angl. *Volts Direct Current*)

USB – universali, nuoseklioji kompiuterinė jungtis (angl. *Universal Series Bus*)

AC/DC – kintamosios srovės konvertavimas į nuolatinę srovę. (angl. *Alternating Current to Direct Current*)

LCD – skystųjų kristalų atvaizdavimo technologija (angl. *Liquid Crystal Display*)

ITU-T G.114 – tarptautinio telekomunikacijų standartizavimo sektorius rekomendacinės gairės vienos krypties transliacijoms (angl. *Telecommunication Standardization Sector*)

RPI – „Raspberry PI”

SVC – vaizdo pakavimo standartas (angl. *Scalable Video Coding*)

CDD – vėlavimas, nuo vaizdo užfiksavimo iki pavaizdavimo (angl. *Capture To Display Delay*)

NTP – tinkle laiko protokolas, skirtas sinchronizuoti sistemų laikui (angl. *Network Time Protocol*)

Įvadas

Dvidešimt pirmame amžiuje, autonominių sistemų, įskaitant savarankiškus robotus, panaudojimas jau matomas ne tik virtuliame pasaulyje ar mokslinės fantastikos filmuose, tačiau ir kasdieniniame gyvenime. Šiais laikais, gana įprasta gatvėse pamatyti važinėjančius savaeigius automobilius, autonominius dulkių siurblius ar vaizdo stebėjimo robotus savo namuose, autonominius robotus-gidus muziejuose, kelionėse. Šios autonominės robotikos sistemos yra kuriamos ir paremtos kompiuterinio pagrindo sistemomis, o tai reiškia, jog šie įrenginiai, kaip ir įprastas kompiuteris, gali nukentėti nuo įvairaus pobūdžio kibernetinių atakų [1].

Autonominiai robotai yra pasmerkti susidurti su panašiomis kibernetinėmis atakomis, su kuriomis įprastiniai kompiuteriai kovoja jau ištisus dešimtmečius, nuo tada, kai šie įrenginiai veikia ir yra valdomi bendro naudojimo tinkle. Dėl itin spartaus autonominių robotų paplitimo ir vis naujų panaudojimo atvejų, pagrindiniai nusikaltėlių taikiniai yra ne tik robotai, kurie atlieka kritinės svarbos užduotis medicinos ar kariniame sektoriuose, tačiau ir paprastesnes funkcijas atliekantys namų ūkio ar paslaugų sektoriaus robotai [2].

Tokie robotai neatlieka sudėtingų medicininių operacijų ar karo veiksmų, kurie gali sužeisti, tačiau įprastai randasi aplinkose, kur sukompromitavus įprastą roboto veikimą, galima nesunkiai pasisavinti privačius duomenis ar sufabrikuoti tokių duomenų sandarą, kilmę. Autonominiai robotai, ypatingai, kurių pagrindinė užduotis yra fiksuoti aplinkos pokyčius, turi gerą priėjimą prie konfidencialių ir tik dedikuotam gavėjui skirtų duomenų, kurie yra perduodami tinklu. Tai gali būti bendra žmogaus informacija (ūgis, svoris, įpročiai), privačios nuotraukos, vaizdo ar garso įrašai, kasdienė dienotvarkė, konfidencialūs finansiniai duomenys, karo strategijos ar politinės paslaptys. Tokių duomenų sukompromitavimas gali būti paaiškinamas, kaip duomenų prieinamumo (duomenų pertraukimo), konfidencialumo (duomenų perėmimo) ir vientisumo (duomenų modifikavimo) sutrikdymai. Nepaisant privatumo pažeidimų, autonominiai robotai, dažniausiai, yra mobilūs, turi judančių dalių, todėl nusikaltėliui perėmus tokio roboto veikimą, gali būti sužeisti aplink esantys žmonės ar sukelti kiti fiziniai nuostoliai.

Plečiantis autonominių robotų panaudojimo sritims, natūralu, jog atsiranda daugiau rizikų, pažeidžiamumų ir saugumo grėsmių, kurias išnaudojus, atsiranda galimos atakų rūšys, nukreiptos prieš autonominius robotus, ypatingai, jeigu šie robotai siunčia ar gauna duomenis tinklu, atliekant komunikavimą su kitais įrenginiais. Pagrindinės atakos, nukreiptos prieš autonominius robotus gali būti apsimetinėjimo, pasyvaus ir aktyvaus šniukštinėjimo, atsisakymo aptarnauti, prieigos pasisavinimo ir buferio perpildymo atakos [3].

Kol autonominių robotų industrija dar nepasiekė galutinio potencialo, labai svarbu užkirsti kelią tolimesniam kibernetinių atakų vystymuisi. Dėl šios priežasties, darbo tikslas yra išsiaiškinti galimas kibernetinio saugumo grėsmes, nustatyti pagrindines rizikas, ištirti galimas atakas. Taip pat, pasiūlyti, kaip užtikrinti saugų komunikavimą tarp autonominio aplinkos stebėjimo roboto ir galinio įrenginio, atsižvelgus į ribotus roboto skaičiavimo pajėgumus ir elektros energijos išteklius.

Darbo tikslas: Pasiūlyti ir įgyvendinti efektyvų autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodą.

Darbo uždaviniai:

- Atlikti analizę, ištirti esamas aplinkos stebėjimo robotų saugumo problemas, nustatyti pažeidžiamumus, identifikuoti atakas ir apžvelgti apsisaugojimo būdus;
- Išanalizuoti robotų naudojamus vaizdo komunikavimo metodus, protokolus ir jų saugumo savybes;
- Suprojektuoti ir sukurti aplinkos stebėjimo roboto saugaus komunikavimo metodą, kuris būtų pritaikomas vaizdo transliavimui iš ribotų skaičiavimo ir energijos išteklių įrenginių;
- Praktiškai realizuoti pasiūlytą saugos komunikavimo metodą, atlikti šio metodo charakteristikų kiekybinę analizę, palyginti gautus rezultatus.

Darbo struktūra:

1. Autonominių robotų saugumo problemų ir pažeidžiamumų apžvalga, naudojamų komunikavimo metodų palyginimas, saugumo užtikrinimas multimedijos duomenų transliacijose.
2. Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo projektavimas.
3. Prototipo realizavimo priemonės, pagrindiniai eksperimentinės dalies kriterijai.
4. Saugaus komunikavimo metodo praktinis įgyvendinimas, eksperimentiniai tyrimai ir rezultatų palyginimas.

1. Autonominių aplinkos stebėjimo robotų analizė

1.1. Autonominių robotų saugumo problemos apibrėžimas

Panašiai kaip ir visos naujos technologijos, žinoma, jog didesnė dalis robotų technologijų vis dar yra nesaugios, o šis trūkumas gali kelti rimtą grėsmę žmonėms, gyvūnams ir terpėms, kuriose robotai atlieka patikėtas užduotis. Tobulėjant ir evoliucionuojant žmogaus ir roboto sąveikai, atsiranda naujos galimos atakų rūšys. Robotams turint vis daugiau mechaninių ar periferinių įrankių ir žmonėms vis labiau jais pasitikint, kuriasi sritis, kur kibernetinio saugumo pažeidimai gali iššaukti pavojų žmonėms ir padaryti negrįžtamą žalą aplinkai.

Garsieji rašytojo Aizeko Azimovo robotikos įstatymai teigia [4]:

1. Robotas negali daryti žalos žmogui arba savo neveikimu leisti, kad žmogui būtų padaryta žala;
2. Robotas turi klausyti komandų, kurias jam duoda žmogus, išskyrus tuos atvejus, kai šios komandos prieštarauja Pirmajam dėsniui;
3. Robotas turi rūpintis savo saugumu, jeigu tai neprieštarauja Pirmajam ir Antrajam dėsniams.

Šie įstatymai buvo paminėti prieš beveik aštuoniasdešimt metų, tačiau sunku patikėti, bet šiais laikais jie įgauna labai rimtą prasmę. Jungtinėse Amerikos Valstijose buvo atliktas tyrimas, kuris parodė, jog žmonės bijo, kad autonominiai robotai gali juos sužeisti ar kitaip jiems pakenkti [5]. Ši baimė yra pagrįsta, kadangi kibernetiniams nusikaltėliams sugebėjus įsilaužti, perimti arba sugadinti robotų veikimą ir valdymą, šis gali pakenkti žmonėms, esantiems aplink jį.

Pagal robotų apibrėžimą, visi robotai yra aprūpinti įranga, kuri suteikia galimybę justi, apdoroti ir įrašinėti aplinkinį pasaulį [6]. Netolimoje ateityje, robotai vis dažniau užpildys fizines erdves, kuriose žmonės gyvena ar dirba, stebėdami žmonių būvį, interpretuodami tai, ką sakome ir ką darome [7].

Nuo 2018 metų įsigaliojus bendrajam duomenų apsaugos reglamentui, privačių duomenų saugumas tapo kaip niekad aktualus. Autonominių robotų technologijoms dar nesant pilnai išsivysčiusioms ir nusistovėjusioms, saugumo aspektai, taip pat, nėra pilnai išdirbti, todėl šios aplinkybės sukelia puikias sąlygas nusikaltėliams pasisavinti privačią ar kitą vertingą informaciją, kurią kaupia ar perduoda robotai.

Tendencijos rodo, jog ateityje robotai bus visur, tai yra, karinėse misijose, atliks medicininės operacijas, statys dangoraižius, asistuos žmonėms prekybos centruose ar gydymo įstaigose, artimai sąveikaus su mūsų šeimos nariais. Sąveika tarp robotų ir žmonių tik didės, todėl labai svarbu užtikrinti, jog robotai ir jų komunikavimo terpės būtų apsaugotos nuo galimų pašalinių įsikišimų į jų veikimą, kadangi pasekmės, sukeltos iškraipyto veikimo, gali būti baisios ir neatstatomos.

Taigi, galima išskirti vienas iš pagrindinių priežasčių, kodėl būtina nagrinėti ir tobulinti robotų fizinį ir kibernetinį saugumą:

- Privatumas;
- Žmonių fizinis saugumas.

1.2. Autonominių robotų apžvalga

Norint geriau suprasti, kodėl yra svarbu užtikrinti autonominių robotų saugumą, būtina apžvelgti jų tipus, panaudojimo sritis ir kitus niuansus, kurie padėtų apibrėžti tikėtinas saugumo spragas ir galimas pasekmes įvairiuose sektoriuose, dėl šių spragų išnaudojimo.

1.2.1. Autonominis robotas

Autonominio roboto sąvoką galima paaiškinti pažodžiui. Žodis autonomija – reiškia sistemas, kurios geba savarankiškai veikti ir priimti sprendimus realaus pasaulio aplinkoje be jokio išorinio valdymo, tam tikrą, dažniausiai, ribotą laiką. Roboto sąvoką galima apibrėžti kaip mašiną, kuri jaučia, galvoja ir veikia. Žinoma, tam robotas turi turėti jutiklius, kurie skirti išgauti informaciją iš aplinkos, kompiuterį, kuris skirtas užduočių apibrėžimui ir informacijos apdorojimui, pavaras, kurios skirtos robotui atlikti tam tikrus veiksmus, judinant savo elementus [8]. Šias tris sudedamąsias dalis galima sulyginti su žmogaus organais, pavyzdžiui, jutikliai atitinka akis, kompiuteris atitinka smegenis, o pavaros gali būti kojos ar rankos.

1.2.2. Tipai, panaudojimo sritys ir saugumo problemos

Kaip jau minėta, autonominiai robotai atlieka savo funkcijas skirtingose aplinkose, priklausomai kokio tipo ir kokioms užduotims atlikti jie buvo sukurti. Jie gali būti skirti gynybai kariniame sektoriuje, apsaugai, namų ūkiui, medicinos sektoriui, aplinkos stebėjimui, o būtent šiuose sektoriuose robotams patikimos sudėtingiausios ir daugiausia atsakomybės nešančios užduotys. Šiose aplinkose yra fiksuojama ir perduodama daugiausia privačių ir jautrių duomenų, todėl kibernetinė apsauga yra vienas iš svarbiausių faktorių, siekiant užtikrinti nepriekaištingą veikimą ir funkcijų atlikimą [3].

Norint geriau išnagrinėti saugumo spragas robotuose, pirmiausia, reikia išanalizuoti galimas atakų situacijas prieš robotus, naudojamus skirtinguose sektoriuose ir aplinkose.

Autonominiai automobiliai. Vienas iš autonominių robotų tipų gali būti autonominiai, bepiločiai automobiliai. Tai automobiliai, kurie geba važiuoti, fiksuojant aplinkos veiksnius be jokio žmogaus valdymo. Šie automobiliai gausiai aprūpinti įvairiais jutikliais, kameromis, radarais ir kompiuteriais. Važiuojant, nuolat vyksta matavimai, skaičiavimai ir komunikacija su kitomis sistemomis. Įrodyta, jog 90 procentų nelaimingų atsitikimų kelyje įvyksta dėl žmogaus klaidos. Naudojant autonominius automobilius, būtų galima ženkliai sumažinti incidentų skaičių, jų sukeliamus nuostolius ir išgelbėti ne vieną gyvybę [9]. Realus prototipas buvo įgyvendintas Australijoje, Kurtin universitete. Čia sukurtas bepilotis autonominis autobusas pavadinimu „Kip“, kursuojantis po universiteto miestelį [10].

Net ir esant ne vienam atvejui, kai autonominiai automobiliai parodė savo neįtikėtinas galimybes, didžioji dalis šios srities specialistų vieningai sutaria, jog vienas didžiausių iššūkių yra apsaugoti bepiločius automobilius nuo išorinių kibernetinių įsilaužimų [11]. Šių autonominių automobilių valdymas paremtas vidinių kompiuterių veikimu, kurie prijungti prie vidaus laidinių tinklų. Dėl plataus spektro įrenginių, kurie yra sujungti tarpusavyje ir su kitomis sistemomis, įsilaužėliams patogu patekti į valdymo sistemas per skirtingų technologijų ryšius. Tai gali būti BlueTooth, Wi-Fi, beraktės atrakinimo sistemos, signalizacijos, radijo ryšys. Saugumo pažeidimus ir atakas, naudojamas prieš autonominius automobilius, galima suskirstyti į šias grupes [12]:

- **Fizinės atakos**. Didžioji dalis variklio valdymų blokų ir programinės įrangos veikia CAN tinklo principu. Nusikaltėliui pavykus gauti priėjimą prie OBD-II ar kitos jungties, kuri suteikia tiesioginį priėjimą prie CAN magistralės, atsiveria kelias atlikti perprogramavimus ir atakuoti skirtingas automobilio valdymo sistemas, kurios valdo kritinius elementus, kaip spidometras, stabdžiai, signalizacija ir kita. Taip pat, gavus prieigą nusikaltėliui nesunku įrašyti kenkėjišką programinę įrangą, kuri ateityje leis valdyti automobilį nuotoliniu būdu.
- **Trumpo nuotolio atakos**. Autonominiams automobiliams turint begalę skirtingų jutiklių ir reaguojant į informaciją gautą iš jų, nusikaltėlis gali paveikti automobilio veikimą fabrikuojant ir siunčiant klaidingus signalus į sistemą (angl. *Spoofing attack*). Elektromagnetinėmis pulsacijomis paveikus radarus, LIDAR ar kitus jutiklius, gali būti pavogtas automobilis, iškraipomas jo maršrutas, gali būti imituojama kliūtis, kuri neleis automobiliui pajudėti [13].
- **Nuotolinės atakos**. Net jeigu ir CAN magistralė su kitais mikroprocesoriais nėra fiziškai sujungti su išoriniais prietaisais, tačiau automobilis yra prijungtas prie kitų infrastruktūrų per bevielį ryšį. Jis skirtas automobiliams komunikuoti tarpusavyje, įrašyti atnaujinimus, perduoti įvairių formų informaciją. Tai atveria galimybes kenkėjiškai veiklai. Esant neapsaugotam komunikavimo kanalui, galimos įvairios atakų rūšys. Kaip pavyzdys, saugumo tyrinėtojai Mileris ir Valasekas 2013 metais, pademonstravo ataką nuotoliniu būdu prieš Chrysler automobilį. Per bevielį ryšį jiems pavyko perimti autonominio automobilio variklio ir stabdžių valdymą [14].

Namų ūkio robotai. Namų ūkiui skirti autonominiai robotai pasižymi skirtingomis funkcijomis. Vieni jų atlieka kasdienes užduotis, kurios gerina gyvenimo komfortą, kiti palaiko kompaniją arba atlieka tam tikrų subjektų priežiūrą. Priklausomai nuo robotų paskirties ir atliekamų funkcijų, skiriasi ir saugumo problemos. Per pastaruosius metus, buvo matyti, jog vis daugiau robotų yra gaminama namų rinkai [15]. Iš namų ūkio robotų grupės galima išskirti dar vieną, sparčiai populiarėjantį tipą, tai socialiniai robotai. Šie gali ne tik būti asmeniniais pagalbininkais, atliekant kasdienes užduotis, tačiau, taip pat gali būti puikūs kompanionai, išmanantys šeimininko įpročius, skonius ir kitas asmenines savybes. Sparčiai besivystantis dirbtinis intelektas suteikia plačias galimybes lengviau integruoti robotus į namų ūkį.

Puikus tyrimas atliktas su modifikuotu *„Savioke“* mobiliuoju robotu, kurio originali paskirtis yra pristatinėti mažus krovinius vidaus patalpose [16]. Šis robotos autonomiškai gebėjo naudotis navigacija, pristatyti vandenį ir paraginti žmones prasimankštinti.

Apmaudu, tačiau dažnai siekiant sumažinti kaštus, tokio tipo robotuose saugumas yra pamirštamas. Tyrimas atskleidė, jog didžioji dauguma socialinių robotų, skirtų namų aplinkai, neatlieka jokio autentifikavimo norint prisijungti prie valdymo. Taip pat, nusikaltėliai žinodami roboto IP adresą ir prievadą, per komunikavimo tinklą, be didesnių pastangų, gali perimti jų valdymą ir pasisavinti perduodamus duomenis, atliekant paketų analizę [15].

Autonominiams namų robotams darantis vis pažangesniems, grėsmių atsiranda vis daugiau. Sukompromituotas robotas gali padegti namus, pavyzdžiui, atlikus trumpąjį sujungimą elektros tinkle arba užnuodyti gėrimus ar maistą, kuriuos tiekia šeimos nariams [17].

Populiarėjant „išmanių namų“ aplinkai, robotai dažnai integruojami į bendrą namų automatikos sistemą. Tai suteikia galimybę sukompromituotam robotui atrakinti duris, išjungti signalizaciją,

valdyti namų termostatą ir kitas sistemas. Netgi, jei robotas nėra integruotas į sistemą, nusikaltėlis minėtus veiksmus gali atlikti panaudojus robotų galimybę komunikuoti balsu, per integruotos garsiakalbius, kadangi dauguma sistemų veikia balso kontrolės principu.

Su tikslu detaliau išnagrinėti saugumo spragas namų ūkio robotuose, atliekama populiarų Jungtinių Amerikos Valstijų rinkos robotų modelių apžvalga ir nustatomi jų saugumo pažeidžiamumai [18].

„WowWee Rovio“ yra mobilus vaizdą fiksuojantis robotas, skirtas nuotoliniam bendravimui ir namų stebėjimui. Jis turi vaizdo kamerą, mikrofoną ir garsiakalbį. Jo kontrolė ir konfigūracijos atliekamos belaidžiu tinklu. Žinant roboto IP adresą ir prievadą, nusikaltėlis gali prisijungti prie roboto sistemos, kadangi standartinėje konfigūracijoje, roboto paskyra nėra apsaugota jokiais slaptažodžiais ar kitais autentifikavimo būdais.

„Erector SpyKee“ robotas yra žaislinis šnipinėjimo robotas su kamera, mikrofonu ir garsiakalbiu. Šio roboto kontrolė vykdoma gamintojo pateikiama programa, kurią galima rasti internete. Standartinė administratoriaus paskyra nėra apsaugota slaptažodžiu.

Atlikus bendrą robotų veikimo tyrimą, nustatyta, jog tiek „Rovio“ tiek „SpyKee“ robotų būvį namuose nesunku aptikti, aktyviau patyrinėjus namų tinklą, į kurį jie prijungti. Panagrinėjus duomenų paketus, nustatyta, jog periodiškai pats „SpyKee“ perduoda identifikacijos paketus programos adresu spykeeworld.com, taip pat, robotų būvį galima patikrinti išsiuntus TCP užklausas į prievadus 80 arba 9001. Tyrimas parodė, jog abu robotai perduoda duomenis RTSP garso-vaizdo transliacijomis jų neužšifravus. Panaudojus duomenų analizės programinę įrangą, iš perduodamų paketų galima atkurti ir peržiūrėti transliuojamus vaizdo įrašus.

Minėtų robotų saugumas paremtas faktu, kad vartotojų namų tinklas, į kurį bus prijungti robotai, bus teisingai sukonfigūruotas, apsaugotas ir užšifruotas. Deja, dažniausiai taip nėra ir įprastai tinkluose naudojamas WEP šifravimas, kuris yra nustatytas kaip pažeidžiamas [19] arba saugumas paremtas silpnais WPA raktais, kurie gali būti atspėti naudojant brutalios jėgos ataką [18].

Apibendrinus, namų ūkio robotai, atliekantys kasdienes užduotis, gali atnešti begalę naudos, palengvinant žmonių rutiną, taip pat jie gali būti puikūs kompanionai ir pagalbininkai. Tačiau būtina sutikti, jog namai yra ta aplinka, kur galima surinkti bene daugiausia privačios informacijos, kuriai gali grėsti pavojus, jeigu kas nors įsilauš į roboto sistemą. Taip pat, pažeidus priežiūrą atliekančių robotų veikimą, šie gali lengvai suklaidinti ar net sužeisti senyvo amžiaus žmones ar vaikus.

Robotai karo pramonėje ir gynybos sektoriuje. Deja, net ir gyvenant civilizuotame šių laikų pasaulyje, karinių konfliktų išvengti nepavyksta. Vystantis technologijoms, robotikos sistemoms pasiekiant vis didesnį autonomijos lygį, perspektyvos sukurti autonominį robotą-karį darosi vis šviesesnės. Bepiločiai, autonominiai arba valdomi, žemės ir oro robotai, jau naudojami Afganistano ir Irano karuose [20]. Gynybos sektoriuje naudojami robotai atlieka pavojingiausias ir daugiausia atsakomybės nešančias užduotis. Didžiausias autonominių robotų privalumas karo zonoje yra tas, jog nesėkmės atveju būtų prarastas tik robotas, tačiau ne kario gyvybė. Dar viena iš priežasčių, kodėl autonominiai robotai tapo tokie populiarūs gynybos sektoriuje yra tai, jog kariniai robotai nepavargsta, jiems nereikia miegoti, jie nejaučia baimės, todėl tokias užduotis, kaip bombų išminavimas gali atlikti tiksliau ir saugiau nei žmogus. Autonominiams robotams vis dažniau patikimos patruliavimo užduotys, kadangi tokiu būdu gali būti išgelbėta begalė karių gyvybių. Tačiau būtina atsižvelgti į tai, jog vaizdo perdavimas komunikavimo kanalu turi būti apsaugotas nuo

piktavalių, kadangi fiksuojamas vaizdas yra ypatingos svarbos. Taip pat, piktavališkais tikslais pavykus gauti prieigą prie šio sektoriaus robotų valdymo, tokie robotai tampa kone pavojingiausiais iš visų, kadangi jų paskirtis dažnai būna karo veiksmams atlikti, todėl jie gausiai apginkluoti sprogmenimis ir ginklais [17].

Viena iš situacijų gali būti susijusi su autonominio, lokalizacijos roboto kibernetinio saugumo pažeidžiamumais. Šis robotas, savarankiškai tyrinėdamas nežinomą aplinką, kuria jos žemėlapį, lokalizuodamas žemėlapyje save. Komandų ir duomenų perdavimui reikalingas dvikryptis ryšys tarp roboto ir operatoriaus. Karinėse misijose šie duomenys yra labai svarbūs, vykdant patruliavimo ar kovines misijas. Užpuolikui pavykus įsilaužti į sistemą per komunikacijos kanalą, prarandamas duomenų vientisumas, prieinamumas ir konfidencialumas. Perėmus roboto kontrolę ir turint prieigą prie jautrių duomenų, užpuolikas, perėmęs privačius duomenis, gali sudaužyti robotą ir užbaigti karinę operaciją. Negana to, užpuolikas gali sukelti tiesioginį pavojų civiliams ar kariams, panaudodamas roboto turimą karinę įrangą. Dar vienas iš galimų scenarijų gali būti, kai užpuolikas išgavęs roboto ar operatoriaus lokaciją, gali sukelti grėsmę jų fiziniam saugumui. Šių atakų metu, nelegaliai perimta įslaptinta informacija, duomenys apie karinę misiją, gali būti pagrindas, norint atlikti teroristinį išpuolį prieš karinės misijos dalyvius, operatorių ar naudojamą karinę įrangą [21].

Aplinkos stebėjimo robotai. Robotai, kurie turi galimybę stebėti aplinką, yra naudojami įvairiuose sektoriuose ir sferose. Įprastai jų paskirtis būna fiksuoti kintančius aplinkos veiksnius, dėl tam tikrų priežasčių ar siekiant konkrečių rezultatų. Stebėjimas gali būti atliekamas dėl saugumo užtikrinimo, parametrų matavimo ar tiesiog informacijos rinkimo ir perdavimo. Šie robotai dažniausiai pasižymi mobilumu, nes turi galimybę judėti ant žemės, ore ar vandenyje. Jie sugeba pasiekti vietas, kur paprastas žmogus prieiti negali, todėl geba užfiksuoti vaizdus, kurių žmogus pamatyti negali.

Pats aplinkos stebėjimas yra veikla, reikalaujanti labai daug darbo jėgos. Aplinkai stebėti yra sukurta ne viena dešimtis autonominių robotų, atsižvelgiant į mobilumo ir autonomijos poreikius. Pavyzdžių yra įvairiausių, pradedant nuo autonominio vandens stebėjimo roboto, kuris skirtas fiksuoti vandens savybių pokyčius po avarijų [22], tęsiant aplinkos stebėjimo robotu, kurio paskirtis atliktį periodinę elektros, šilumos jėgainių apžiūrą, naudojant lokalizacijos įrangą [23] ir baigiant namų ūkiui ar gynybos sektoriui skirtais robotais.

Pagrindinis autonominių aplinkos stebėjimo robotų iššūkis yra duomenų vientisumo, konfidencialumo ir prieinamumo užtikrinimas. Autonominiai stebėjimo robotai įprastai pasižymi puikiu mobilumu, yra aprūpinti vaizdo stebėjimo įranga, todėl gali užfiksuoti jautrius vaizdus. Namuose, aplinką stebintys robotai, gali būti socialiniai, skirti kompanijai ar nuotoliniam bendravimui, taip pat, darosi populiaru naudoti aplinką stebinčius robotus vietoj tradicinių vaizdo stebėjimo apsaugos sistemų [24]. Autonominiai aplinkos stebėjimo robotai, kaip ir kiti daiktų interneto įrenginiai, dažniausiai turi ribotus tinklo, skaičiuojamosios galios, atminties ir energijos resursus [25]. Resursų stoka įprastai lemia neapsaugotą arba prastai apsaugotą komunikavimą tarp įrenginio ir galinio vartotojo. Praktikoje užfiksuotas ne vienas pavyzdžių, kai dėl neapsaugotos komunikacijos, pašaliniai asmeniai turėdavo galimybę gauti prieigą prie vaizdo transliacijų duomenų. Atlikto tyrimo rezultatai parodė, jog rinkoje populiarus Wi-Fi robotas yra pažeidžiamas dėl neapsaugotų domenų adresų ir RTSP protokolo. Tai suteikė galimybę neautorizuotam vartotojui ne tik klausytis ir stebėti vaizdo transliaciją iš įmontuotos kameros, tačiau ir pačiam perduoti garsą per robote įmontuotą garsiakalbį [26].

Aplinkos stebėjimo robotai yra glaudžiai susiję su kariniu sektoriumi, todėl šis sektorius, stipriai investuodamas, skiria ypatingą dėmesį autonominių karinių technologijų plėtrai [22]. Karinių robotų industrijos rinkos pokyčiai Jungtinėse Amerikos Valstijose rodo, jog periode nuo 2000 m. iki 2015 m. rinkos vertė pakilo nuo $2,4 milijardų iki $7,5 milijardų, o 2025 metais tikimasi, jog rinka pasieks $16,5 milijardus [27]. Bepiločių dronų ir dirbtinio intelekto derinimas, leidžia pasiekti visišką autonomiją ir suteikia galimybę iš esmės pakeisti veiksmus karo zonoje ir pačią karinę pramonę. Autonominiai, bepiločiai aplinkos stebėjimo robotai, karo pramonėje gali būti skirti žemės, vandens, kosmoso ir oro patruliavimui [28]. Verta paminėti, kad dažniausiai šie robotai yra gausiai apginkluoti ir be galo pavojingi. Dėl šių priežasčių, investuojama ne tik į pačios technologijos plėtojimą, tačiau ir į roboto fizinį ir kibernetinį saugumą, kadangi nelegaliai perėmus karinio roboto valdymą, šis gali sukelti katastrofiškas pasekmes.

Vienas iš pavyzdžių gali būti ši situacija, kai 2009 metais vykusioje karinėje operacijoje, kurioje buvo pagauti keli nusikaltėliai iš teroristų grupuotės, buvo atlikta ataka prieš autonominį bepilotį droną, kuris transliavo visą operaciją [29]. Teroristai, naudodami „SkyGrabber“ programinę įrangą ir palydovinę anteną, sugebėjo perimti vaizdo transliaciją, kadangi pastaroji nebuvo užšifruota. Visų nuostabai, valdžios organams ši saugumo spraga buvo žinoma nuo 1990.

1.3. Autonominių robotų saugumas

Panašu, jog pamažu robotai užpildys įvairiausias roles žmonių namuose, versle, gamybos pramonėje, apsaugos ir gynybos sektoriuose. Prognozės tai patvirtina, tyrimai rodo, jog pasaulinės išlaidos robotikai 2020 metais sieks $188 milijardus [17]. Kadangi dauguma šių „protingų“ mašinų yra savaeigės ir autonomiškos, labai svarbu užtikrinti jų apsaugą nuo įvairių įsilaužimų. Neapsaugojus robotų, jų pagalbiniai resursai ir gebėjimai gali greitai tapti įrankiais, galinčiais pakenkti ir sukelti didelę žalą aplinkai ir žmonėms, kuriems jie skirti „tarnauti“ [17]. Bilo Geitso nuomone, esama situacija su robotų sparčiu populiarumo augimu, primena dvidešimto amžiaus aštuntą dešimtmetį, kai atsiradus personaliniams kompiuteriams, juos turėjo tik privilegijuoti, pasiturintys asmenys, o dabar jų yra milijardai [6]. Šiais laikais, autonominiai robotai yra priversti susidurti su panašiomis saugumo problemomis, su kuriomis susidūrė ankstyvieji kompiuteriai Interneto technologijos atsiradimo laikais [30].

Pilna roboto sistema įprastai sudaryta iš pačio roboto, operacinės sistemos, programinės įrangos, valdymo sistemos taikomųjų programų, interneto paslaugų, debesijos paslaugų, tinklų ir kitų elementų. Visos sistemos elementai suteikia plačias galimybes atlikti daugybę kibernetinių atakų prieš robotą.

1.3.1. „Daiktų internetas“ ir robotai

Kelis pastaruosius dešimtmečius internetas sparčiai evoliucionavo. Rinkoje vis dažniau atsirandant naujiems „išmaniems“ įrenginiams pamažu ėmė kurtis daiktų interneto sąvoka. Tikslaus apibrėžimo nėra, tačiau šią sąvoką galima paaiškinti kaip įrenginių tinklą, kuris sujungia išmaniuosius įrenginius, suteikia jiems galimybę komunikuoti vienam su kitu prietaiso-prietaisui architektūros principu (angl. *M2M*) ir perduoti informacija į Internetą. Dažnai, procesai daiktų internete vyksta esant tam tikram autonomijos lygiui, be žmogaus įsikišimo. Tačiau tai reiškia, jog didžioji dalis įrenginių daiktų internete turi ribotus atminties, skaičiavimo ir energijos išteklius, kas apsunkina jų apsaugojimą. Didžiąją dalį robotų galima priskirti šių įrenginių grupei, todėl daiktų interneto saugumo problemos yra tiesiogiai susijusios su robotais, ypatingai autonominiais. Jau dabar susiduriama su didelėmis

kibernetinio saugumo problemomis daiktų interneto įrenginiuose, kurios daro neigiamą poveikį buitiniams vartotojams, kompanijoms, pramonės, gamybos, gynybos sektoriams.

1.3.2. Dažniausiai pasitaikančios saugumo spragos

Po atliktų mokslinių tyrimų, nustatyta beveik 50 robotų saugumo pažeidžiamumų, tačiau verta aptarti tik dažniausiai pasitaikančias ir išnaudojamas kibernetinio saugumo problemas [17]:

- **Autentifikavimo, identifikavimo problemos**. Vartotojų identifikavimo procesas yra svarbus ir robotų sistemose. Tik autentifikuoti vartotojai turi turėti prieigą prie roboto valdymo sistemos ir serverio, į kurį perduodami duomenys. Esant nesaugiam autentifikavimo funkcionalumui, nusikaltėliai gali nelegaliai perimti roboto valdymą ar jo perduodamus duomenis. Nors autentifikavimas yra plačiai taikomas kompiuterinėse sistemose, tačiau praktika rodo, jog robotų sistemose autentifikavimas visai netaikomas arba yra silpnas, todėl lengva jį „nulaužti“;
- **Silpna kriptografija**. Duomenų apsauga yra labai svarbi norint užtikrinti vientisumą ir konfidencialumą. Robotai gali kaupti arba perduoti jautrią informaciją, tai gali būti slaptažodžiai, šifravimo raktai, įvairių paskyrų prisijungimo duomenys ir kt. Būtent dėl to, norint apsisaugoti nuo duomenų vagysčių, komunikavimo kanalas turi naudoti šifravimą. Taip pat, robotai gauna atnaujinimus, todėl tinkama kriptografija būtina, kad būtų užtikrinta, jog vietoj atnaujinimų nėra įrašoma kenkėjiška programa (vientisumo užtikrinimas);
- **Privatumo spragos**. Priklausomai nuo roboto paskirties ir atliekamų funkcijų, jis renka ir perduoda tam tikrus duomenis. Kai kurie duomenys turėtų būti privatūs ir neatskleisti niekam kitam, tik robotų šeimininkui. Vartotojai turi turėti visišką ir nepriklausomą savo duomenų kontrolę. Keletas robotų per savo mobilias programas siunčia privačią informaciją serveriams, be vartotojo žinios. Ši informacija gali būti renkama stebėjimo ir sekimo tikslais;
- **Silpna numatytoji konfigūracija**. Dauguma robotų suteikia galimybę atlikti modifikacijas ir konfigūruoti roboto funkcijas, naudojant programinę įrangą. Galimybė atlikti konfigūracijas turėtų būti suteikta tik autorizuotiems vartotojams, identifikavus save slaptažodžiais ar kitomis priemonėmis. Praktika rodo, jog dėl naudojamų standartinių slaptažodžių ir vartotojų vardų, kurie yra viešai skelbiami, nusikaltėlis lengvai gali gauti prieigą prie programinės įrangos, kuri leidžia atlikti modifikacijas roboto veikime;
- **Pažeidžiamos atvirojo kodo robotų sistemos ir bibliotekos**. Daug robotų naudoja atvirojo kodo sistemas ir bibliotekas. Viena iš populiariausių Robotų Operacinė Sistema (angl. *ROS*) naudojama daugelyje robotų, kurie yra sukurti skirtingų gamintojų. Šioje operacinėje sistemoje nustatyti tokie kibernetinio saugumo pažeidimai, kaip autentifikavimo problemos, nešifruota komunikacija atviro teksto formatu, silpna autorizavimo sistema. Praktika dalintis programinės įrangos kodais, bibliotekomis ar operacinėmis sistemomis nebūtų bloga, jeigu šie programiniai resursai būtų apsaugoti, apmaudu, tačiau daugeliu atvejų taip nėra;
- **Nesaugūs ryšiai ir komunikavimas**. Komunikavimas yra gyvybiškai svarbi dedamoji robotų sistemoje, kuri leidžia vartotojams ir robotams sklandžiai bendrauti. Kaip pavyzdys, gali būti robotų komunikavimas su kompiuteriu. Vartotojai gali siųsti komandas realiu laiku, naudojant mobilias programas arba robotai gali prisijungti prie interneto tiekėjo, debesijos paslaugų su tikslu gauti atnaujinimus ar kitą informaciją. Būtina išlaikyti saugų komunikavimo metodą, naudojant kriptografiškai apsaugotus komunikacijos būdus. Priešingu atveju, nusikaltėliai gali nesunkiai įsiterpti į komunikavimo kanalą ir pavogti konfidencialią informaciją arba sutrikdyti kritinių robotų komponentų veikimą. Labai gaila, tačiau dauguma kuriamų robotų

naudoja nesaugius komunikavimo metodus. Dažniausiai Wi-Fi, Bluetooth ar kitomis technologijomis siunčiama kritiškai svarbi informacija, būna atviro teksto formato arba silpnai užšifruota.

1.3.3. Atakos prieš autonominius robotus

Galimos atakos prieš autonominius robotus būna įvairios. Kaip pavyzdys, aktyvus nešifruotų duomenų srautų šnipinėjimas, siekiant išgauti jautrią informaciją, pasyvus neapsaugotų komunikavimo kanalų stebėjimas, siekiant iššifruoti silpnai apsaugotą srautą, išgaunant autentifikavimo informaciją, taip pat, trumpo nuotolio atakos ir kt. Įprastas kibernetines atakas prieš autonominius robotus ir kitus, daiktų interneto įrenginius, galima suskirstyti į šias pagrindines grupes [31]:

- **Fizinės atakos**. Įgyvendinamos sugadinant ar kitaip pažeidžiant įrenginių komponentus. Dažniausiai, autonominiai robotai veikia aplinkose, kuriose jie nėra apsaugoti nuo išorinių fizinių išpuolių;
- **Žvalgybos atakos** (angl. *Reconnaissance attacks*). Neleistinas pažeidžiamumų atradimas, naudojant įvairius žvalgybos būdus. Tai gali būti tinklo prievadų skenavimas, paketų analizavimas, srauto analizavimas, užklausų siuntinėjimas, siekiant išgauti IP ar kitą informaciją;
- **Paslaugos trikdymo ataka** (angl. *Denial Service Attack*). Šios atakos tikslas yra padaryti įrenginį ar tinklo išteklius neprieinamus naudotojams. Dėl mažų atminties resursų, ribotų skaičiavimo išteklių, dauguma IoT tipo įrenginių yra pažeidžiami išteklių išnaudojimo išpuoliams (angl. *Resource enervation attacks*);
- **Prieigos atakos** (angl. *Access attacks*). Neautorizuoti asmenys gauna prieiga prie tinklo arba įrenginių, prie kurių jie neturi teisės prisijungti. Prieigos atakos yra dviejų tipų: fizinės ir nuotolinės;
- **Atakos, kėsinasint į privatumą** (angl. *Attacks on privacy*). Tokias atakas apibrėžti sunku, tačiau tai atakos, kai vienokiu ar kitokiu būdu kėsinamąsi į privačius duomenis. Gauti privačius duomenis ar prieigą prie jų galima įvairiais būdais:
    - Duomenų kasyba (angl. *Data mining*);
    - Kibernetinis šnipinėjimas (angl. *Cyber espionage*);
    - Slapto pasiklausymo atakos (angl. *Eavesdropping*);
    - Sekimo atakos;
    - Slaptažodžių pasisavinimo atakos.

1.4. Autonominių aplinkos stebėjimo robotų charakteristikos

Autonominiai aplinkos stebėjimo robotai gali komunikuoti su kitais įrenginiais ir sistemomis. Šie įrenginiai bendrauja skirtingomis priemonėmis, įskaitant belaidžių mobiliųjų ryšių technologijas (3G arba LTE), WLAN, belaidėmis Wi-Fi ar kitomis technologijomis. Tokia robotų klasifikacija priklauso nuo jų dydžio, judėjimo tipo, mobilumo, maitinimo šaltinio pobūdžio, veikimo laiko, automatizacijos lygio, taip pat, veikimo principo, tai yra, kokio tipo užduotys atliekamos (loginės, fizinės ar mišrios) ir, galiausiai, ar jie yra IP tinkliniai įrenginiai.

Šios charakteristikos lemia minėtų robotų ir kitų IoT prietaisų gebėjimą veikti ir/arba jausti, įtakoja galios poreikį, sąsają su fiziniu pasauliu, lemia nepertraukiamo ryšio stabilumą ir mobilumą. Vieni jų

turi būti greiti ir patikimi, užtikrinantys įvairiapusį funkcionalumą ir saugumą, tuo tarpu kiti – ne. Visa tai priklauso nuo naudojimo srities ir robotui patikimų užduočių.

Dažnai didžioji dalis robotų resursų yra išnaudojami patikėtam specifiniam funkcionalumui vykdyti. Ši priežastis liemia, jog yra labai sunku įgyvendinti ir naudoti tvirtą saugumo mechanizmą roboto kibernetiniam saugumui užtikrinti, dėl stipriai ribotų autonominių robotų išteklių: ribotos energijos, skaičiuojamosios galios ir atminties.

1.5. Autonominių aplinkos stebėjimo robotų komunikavimo kanalo sauga

Iš apžvelgtų situacijų, grėsmių ir atakų pavyzdžių, galima daryti prielaidą, jog didelė dalis saugumo pažeidžiamumų, prieš autonominius aplinkos stebėjimo robotus, yra įgyvendinti dėl neapsaugoto duomenų perdavimo, tarp roboto ir galinių įrenginių. Pasikliaudamos kanalinio, tinklo, transportavimo ar taikomųjų programų lygmenyse perduodamų duomenų šifravimo, autentifikavimo funkcionalumais ir kitais saugos mechanizmais, tradicinės internetinės sistemos sumažina atakų, kurios įvyksta dėl nesaugaus komunikavimo metodo, grėsmę. Nors, daliai IoT įrenginių šie saugos sprendimai pritaikomi, tačiau didžioji dalis įrenginių nėra pajėgūs naudoti visaverčių apsaugos mechanizmų ir saugiausių kriptografinių algoritmų, dėl ribotų komunikavimo galimybių, skaičiavimo išteklių, atminties kiekio ir energijos trūkumo.

Naujausi skaičiavimai ir prognozės rodo, jog iki 2020 metų, pasaulyje bus virš 50 milijardų tinkle prijungtų, įvairiausių, ribotus išteklius turinčių įrenginių, tokių kaip tipiniai daiktų interneto įrenginiai, mažos įterptinės sistemos, kompiuteriai, belaidžiai jutikliai, robotai. Tikimasi, jog didžiąją dalį duomenų srautų, kuriuos sugeneruos šie įrenginiai, sudarys multimedijos duomenys (vaizdo ir garso duomenys). Prognozuojama, jog šie duomenys užims 80% viso Interneto protokolo duomenų srauto 2019 metais [32]. Šios prognozės patvirtina, jog labai svarbu rasti efektyvius būdus, kaip užtikrinti tokių duomenų saugumą.

Aplinkos stebėjimo robotai ar kiti IoT įrenginiai, gebantys fiksuoti vaizdą, gali būti skirti patruliavimo ir saugumo užtikrinimo funkcijai atlikti, namų stebėjimui, taip pat, tai gali būti išmaniosios sankryžų kameros ir kt. Žmonės vis dažniau renkasi pigesnius, mobilesnius, žymiai kompaktiškesnius įrenginius vaizdo transliacijoms atlikti ir vis rečiau naudoja senas vaizdo stebėjimo ir transliavimo sistemas. Aplinką fiksuojančių robotų vaizdo duomenų turinys, tipiškais atvejais, yra labai svarbus ir negali būti pasiekiamas pašaliniams asmenims. Vis dėlto užtikrinti patikimą perduodamų vaizdo transliacijos duomenų saugumą dėl ribotų energijos, skaičiavimo ir atminties išteklių įrenginiuose yra didelis iššūkis, kurį sunku įgyvendinti. Didžioji dalis, rinkoje esančių vaizdo transliavimo įrenginių, nešifruoja vaizdo duomenų, kadangi šis funkcionalumas išbrangina įrenginio pagaminimą. Įprastai vienintelis apsaugos mechanizmas būna paprasčiausias autentifikavimasis tarp serverio ir kliento, tačiau sugebėjus gavus prieigą prie tinklo, duomenys gali būti nesunkiai atkuriami, naudojant duomenų srautų analizės įrankius.

Būtent dėl šių priežasčių, saugių vaizdo transliacijų perdavimas ribotų išteklių įrenginiais, yra aktuali tema, kurią būtina nagrinėti. Siekiant užtikrintį saugų komunikavimą tarp aplinkos stebėjimo roboto ir galinio įrenginio, būtina atlikti detalią protokolų, kurie naudojami vaizdo duomenų perdavimui, analizę.

1.5.1. Vaizdo duomenų srautų perdavimas

Vaizdo transliacijos duomenų perdavimą (angl. *Video streaming*) galima apibrėžti, kaip iš serverio siunčiamus supakuotus vaizdo duomenis klientui, pagal jo apdorojamų duomenų normą (angl. *Rate of consumption*) [33].

Formaliai žinomi trys transliuojamų medijos duomenų perdavimo tipai [34]:

1. Saugomų garso ir vaizdo duomenų transliavimas;
2. Tiesioginis garso ir vaizdo duomenų transliavimas;
3. Realaus laiko interaktyvus garso ir vaizdo transliavimas.

1.5.2. Vaizdo duomenų srautų transliavimo ypatumai

Vaizdo transliavimo duomenų perdavimas taikomosiose programose yra susijęs su nemažai parametrų ir metodikų. Pagrindiniai aspektai, susiję su vaizdo perdavimu, yra tinkamas vaizdo suspaudimas (angl. *Compression*), kodavimas, turinio pristatymo tinklo architektūra (angl. *Content Delivery Network*) ir jos valdymas, duomenų perdavimo/pristatymo technikos, tinklo kokybės serviso mechanizmai, kliento medijos grotuvų parametrai ir galimybės, duomenų vėlavimas, trikdymas, praradimas, jų apsaugojimas. Vaizdo duomenų perdavimą gali apsunkinti keletas faktorių, tai yra, realaus laiko duomenų pristatymo reikalavimai, patikimų paslaugų kokybės (angl. *Quality of Service*) mechanizmų nebuvimas daugumoje Interneto maršrutų galinių taškų, duomenų srautų apkrovos svyravimai, maršrutų perkrovos Internete, taip pat, kintama transliavimų kanalų kokybė, kurią ypatingai įtakoja mobiliųjų, belaidžių įrenginių paplitimas. „Daiktų interneto“ įrenginių paplitimas, reiškia ribotus energijos, skaičiavimo ir atminties išteklius, kurie lemia ir apsunkina kokybės, saugumo, patikimumo faktorių užtikrinimą [33]. Atsižvelgiant į šiuos pagrindinius veiksnius vaizdo pristatymo kokybei, saugai užtikrinti, norint kokybiškai transliuoti vaizdą, išgaunant priimtino veikimo reikalavimus, reikalinga kruopšti duomenų perdavimo, vaizdo transliavimo protokolų analizė.

1.5.3. TCP, UDP protokolai

Pagrindiniai transportavimo lygmens protokolai medijos duomenų perdavime yra TCP ir UDP. Šie protokolai palaiko tokias funkcijas, kaip duomenų sutankinimas, klaidų kontrolė, tinklo perkrovos ir srautų kontrolės. Verta detaliau panagrinėti protokolų bendras savybes ir funkcijas.

UDP ir TCP gali sutankinti duomenų srautus iš skirtingų taikomųjų programų, veikiančių tame pačiame įrenginyje, su tokiu pat IP adresu. Kalbant apie klaidų kontrolės paskirtį, TCP ir daugelio UDP diegimų skaičiuoja kontrolinę sumą, kad aptiktų klaidas. Jeigu gaunamame pakete aptinkama pavienė ar kelios bitų klaidos, TCP/UDP protokolo sluoksnis pašalina paketą taip, kad protokolas esantis viršuje TCP/UDP negautų pažeisto paketo. Priešingai nei UDP, TCP naudoja retransliavimą, kad prarasti paketai būtų sugražinti. Būtent dėl to, TCP pasižymi patikimu paketų perdavimu, kai, tuo tarpu, UDP patikimumo užtikrinti negali. TCP veikia perkrovos kontrolė, kuri skirta valdyti siunčiamų duomenų srautą, siekiant išvengti tinklo perkrovos. Tai dar vienas funkcionalumas, kuris išskiria TCP nuo UDP. Taip pat, TCP vykdo srauto kontrolę, kad būtų išvengta buferio perpildymo, kas lemtų paketų praradimą arba jų retransliavimą. UDP neturi jokio panašaus mechanizmo [35].

TCP vykdant pakartotinį paketų perdavimą sukuriamas vėlavimas, kuris kai kuriais atvejais nėra priimtinas tam tikroms transliacijoms. Priklausomai nuo vaizdo transliacijos poreikio, būtina

apsvarstyti kas svarbiau, greitis ar patikimumas. Kaip pavyzdį galima panagrinėti situaciją, kur aplinką stebintis robotas karinėje operacijoje patruliuoja pasienio poste. Šiam robotui fiksuojant aplinką, svarbu, jog vaizdo duomenys būtų perduodami realiu laiku, su kuo mažesniu vėlavimu, kadangi kiekviena sekundė gali būti lemiama. Tokiu atveju puikiai tinka UDP veikiantis transporto lygmenyje. Su kelių paketų praradimu vaizdas perduodamas su minimaliu vėlavimu. Kaip jau minėta, UDP negarantuoja paketų pristatymo, tačiau tai gali padėti užtikrinti UDP pagrindu veikiantis protokolas, pavyzdžiui, RTP.

Nagrinėjant svarbias charakteristikas, tai yra vėlavimo laiką ir vėlavimą tarp paketų (angl. *Delay jitter*), matoma, jog šios gali lemti kokybišką multimedijos duomenų transliavimą todėl, UDP protokolas yra priimtinesnis realaus laiko transliacijoms nei TCP. Tačiau būtina paminėti, jog nekontroliuojami UDP duomenų srautai, gali stipriai pakenti TCP srautams ar net bendram interneto stabilumui [36].

Išnagrinėjus transporto sluoksnio protokolų pranašumus ir trūkumus, vykdant multimedijos transliacijas, galima apibrėžti esminius aspektus ir skirtumus:

1. UDP pasižymi mažesniu vėlavimu nei TCP, tačiau nėra toks patikimas;
2. UDP palaiko IP Multicast transliacijas;
3. Laikui jautrių transliacijų atveju, UDP yra greitesnis protokolas, kadangi jis nelaukia, kol bus gautas patvirtinimas iš kliento pusės, o prarasti paketai iš naujo nėra siunčiami, transliacija tęsiama net ir gavus pažeistus paketus (kas gali lemti pablogėjusią transliacijos kokybę);
4. TCP klaidų tikrinimo mechanizmas leidžia iš naujo nusiųsti prarastus paketus, tačiau tai naudoja papildomus resursus;
5. TCP originaliai buvo sukurtas patikimam statiniam duomenų perdavimui [37], todėl vaizdo duomenų perdavimas per TCP labiau tinkamas vaizdo stebėjimui pagal pareikalavimą, transliuojant išsaugotus duomenis (angl. *On demand*).

1.5.4. RTP/RTCP ir RTSP protokolai

Multimedijos duomenų perdavimui internetu buvo sukurta protokolų šeima, kurią sudaro Realaus laiko transporto protokolas RTP (angl. *Real-Time Transport Protocol*), šio protokolo kontrolę vykdantis RTP valdymo protokolas RTCP (angl. *RTP Control Protocol*) ir Realaus laiko transliavimo protokolas RTSP (angl. *Real-time Streaming Protocol*).

RTP yra standartizuotas protokolas skirtas realaus laiko duomenų transportavimui, tai yra vaizdo ir garso perdavimui. Duomenys šiuo protokolu nebūtinai gali būti perduodami realiu laiku. Jis gali transliuoti vaizdą ir garsą pagal paklausą. Šis protokolas, dažniausiai, veikia poroje su minėtu RTCP protokolu. RTP naudojamas multimedijos duomenų srautų perdavimui, tuo tarpu RTCP atlieka perduodamų duomenų kontrolę [38]. Įprastai ši protokolų pora veikia kartu su UDP transporto sluoksnyje.

RTP yra protokolas, kuris padeda įgyvendinti realaus laiko duomenų perdavimą, gebantis sekti paketų praradimus, turinio identifikavimą ir laiko žymėjimą (angl. *Timestamping*). Minėtoji savybė yra viena svarbiausių, kadangi gavus duomenų paketus, gavėjas panaudoja laiko žymę, kad galėtų atkurti originalų laiką ir ištransliuoti duomenis teisingu dažniu [39]. Protokolas nereaguliuoja tinklo pralaidumo parametrų ir neužtikrina paslaugų kokybės rodiklių, tačiau šį funkcionalumą užtikrina RTCP.

RTP yra labai lanksčių mechanizmų protokolas, kuris gali būti lengvai pritaikomas naujoms sistemoms. Skirtingų pobūdžių programoms RTP apibrėžia skirtingus profilius ir vieną arba daugiau duomenų tipo formatų (angl. *Payload format*). Profilis gali apibrėžti atliktus RTP papildymus ir modifikacijas, kurios pritaikomos taikomųjų programų specifinei klasei. Taigi, šis protokolas gali būti nesunkiai pritaikomas ypatingai klasei įrenginių, pavyzdžiui, įrenginiams, turintiems ribotus skaičiavimo, atminties ir energijos išteklius.

RTCP padeda surinkti informaciją apie kokybės rodiklius iš gavėjų, duomenis, reikalingus skirtingų medijų sinchronizacijai, taip pat, kaupia informaciją apie sesijos dalyvius [40]. Protokolas renka statistiką apie išsiųstų ir prarastų paketų skaičių, paketų vėlavimo informaciją. Ši funkcija gali būti naudinga dinamiškoms programoms, kurios priklausomai nuo gautų atsakymų, siųs geros arba prastesnės kokybės duomenis, priklausomai nuo tinklo apkrautumo. RTCP leidžia korealiuoti ir sinchronizuoti skirtingus duomenų srautus, kurie atkeliavo iš to paties siuntėjo.

Pagrindinės RTP ir RTCP savybės:

- RTP vykdo realaus laiko vaizdo ir garso transliacijos duomenų perdavimą nuo mazgo iki mazgo;
- RTP neatlieka laiko kontrolės;
- RTP dažniausiai veikia kartu su UDP, kad galėtų pasinaudoti multipleksavimo ir kontrolinės sumos patikrinimo funkcija;
- Protokolai neatlieka patikimumą užtikrinančių srauto ar apkrovos kontrolių;
- RTP protokolo karkasas lengvai pritaikomas įvairaus pobūdžio taikomosioms programoms;
- RTP/RTCP suteikia funkcionalumo ir kontrolės mechanizmus, reikalingus tinkamam realaus laiko duomenų perdavimui užtikrinti. RTP suteikia laiko žymes, sekos numerius, kas padeda įgyvendinti duomenų srauto ir perkrovos kontrolę kituose lygmenyse.

RTSP yra taikomųjų programų lygmens protokolas, kuris leidžia kontroliuoti realaus laiko transliacijų duomenų perdavimo, per Interneto protokolą, parametrus. Suteikiamos tokios funkcijos kaip pauzė, atsukimas iki norimos ankščiau ištransliuotos dalies, transliacijos prasukimas ir kt. Pats protokolas tipiškai neperduoda transliacijos duomenų, o kaip autorius apibrėžia yra labiau kaip *"Nuotolinio valdymo pultelis multimedijos duomenų serveriams"* [40]. Iš esmės jis sukurtas dirbti su kitais žemesnio lygmens protokolais RTP ar RSVP.

RTSP gali veikti daugiaadresiame režime (angl. *Multicast*) arba atlikti pavienę transliaciją. Protokolo pagrindės funkcijos yra duomenų išieškojimas medijos serveriuose, pakvietimų serveriams dėl konferencijos išsiuntimas, pranešimų apie naujas transliacijas išsiuntimas.

Pagrindinės savybės:

- RTSP taikomųjų programų lygmens protokolas, kurio sintaksė ir operacijos primena HTTP, tačiau jos vykdomos su garso ir vaizdo duomenimis. Protokolas ištaiso HTTP trūkumus ir, netgi, atlieka operacijas geriau [34];
- RTSP duomenų perdavimo metu išlaiko serverio būseną, priešingai nei HTTP;
- RTSP pranešimai siunčiami nepriklausomai, tai yra, atskiru kanalu nuo pagrindinių perduodamų duomenų srautų. (angl. *Out-of-band*);
- Priešingai nei HTTP, RTSP ir klientas, ir serveris gali išduoti prašymus;
- Kliento-serverio architektūra;

– RTSP gali būti įdiegiamas skirtingose operacinėse sistemose, o tai suteikia suderinamumą tarp kliento ir serverio, jeigu jie veikia skirtingose platformose;
– RTSP atlieka skirtingų transliacijų sinchronizavimą.

1.5.5. RTP protokolų šeimos sauga

RTP ir RTCP protokolų pagrindiniai saugumo funkcionalumai įgyvendinami žemesniuose tinklo ir transporto lygmens sluoksniuose, užtikrinant konfidencialumą, autentifikavimą ir vientisumą. Konfidencialumo užtikrinimas vaizdo duomenų perdavime, yra vienas iš svarbiausių tikslų, susijusių su saugumu. Konfidencialumas reiškia, jog tik specifinis gavėjas gali dekoduoti gautus paketus, o kitiems ši informacija turi būti bereikšmė – be jokios naudingos informacijos. Konfidencialumas įgyvendinamas naudojant šifravimą. Vienas iš siūlomų šifravimo būdų yra, jog visi baitai, kurie bus perduodami žemesnio sluoksnio pakete, bus užšifruojami kaip vienetas [38].

RTCP protokole prieš šifravimą paruošiamas atsitiktinis 32 bitų skaičius ir priskiriamas kiekvienam vienetui. RTP atveju jokio prefikso nėra, vietoj to atsitiktinai inicijuojami sekų numeriai ir laiko žymės. Toks būdas yra laikomas pažeidžiamu vektoriumi dėl silpnų atsitiktinių parametrų. Standartiškai RTP duomenų šifravimo algoritmas yra DES algoritmas, šifravimo blokų grandinės CBC režimu. Šis šifravimo būdas buvo pasirinktas dėl savo praktiškai patogaus pritaikymo vaizdo ir garso duomenų perdavimui Internetu. Tačiau, įrodyta, jog DES yra lengvai pažeidžiamas. Priklausomai nuo programų pobūdžio, rekomenduojama naudoti stipresnius šifravimo algoritmus, tokius kaip 3DES (angl. *Triple-DES*) arba kaip apibrėžia SRTP (angl. *Secure real time transport protocol*) – AES algoritmo šifravimą [41]. Gali būti naudojami ir kiti šifravimo algoritmai įgyvendinami ne RTP priemonėmis. Standartinė RTP specifikacija suteikia tik ribotas medijos duomenų apsaugojimo galimybes ir neapibrėžia jokio raktų apsikeitimo mechanizmo [42]. Autentifikavimo ir duomenų vientisumo mechanizmai nėra apibrėžti RTP protokolo lygmenyje, kadangi šios funkcijos nebūtų įgyvendinamos be raktų valdymo infrakstruktūros. Šios paslaugos teikiamos žemesnio lygio protokoluose.

RTSP pakartotinai naudoja visus žiniatinklio saugos mechanizmus. Visi HTTP autentifikavimo mechanizmai, tokie kaip bazinis ir asimiliuotasis (angl. *Digest*) autentifikavimai yra tiesiogiai pritaikomi. Įprasta praktika taikyti transporto ir tinklo lygmenų saugos mechanizmus [43]. Kartu su autentifikavimo mechanizmais, aplinkose, kur reikalingas papildomas saugumas, rekomenduojama šifruoti RTSP valdymo pranešimus.

1.5.6. Dinamiškas adaptyvusis transliavimas per HTTP

Dinamiškas adaptyvusis transliavimas per HTTP (angl. *Dynamic Adaptive Streaming over HTTP*) – DASH – tai vadinamasis traukos (angl. *Pull-based*) tipo duomenų transliavimo standartas, kur transliuojantis įrenginys atlieka pagrindinį vaidmenį adaptyviai transliuojant medijos duomenis [44]. DASH mechanizmai apibrėžia, kaip išskaidyti vaizdo transliacijos turinį į daug dalių, kad jos galėtų būti perduodamos per HTTP, labiausiai naudojamą šių dienų protokolą, tokiu būdu, kad klientas žinotų apie prieinamus duomenis. Toks skaidymas leidžia klientui atkurti vaizdo transliaciją iš skirtingų duomenų dalelių, kurios yra iš skirtingų lokacijų su skirtingomis duomenų perdavimo spartomis [45]. Adaptyvaus transliavimo technologija leidžia reguliuoti vaizdo transliacijos kokybę, priklausomai nuo kliento tinklo pralaidumo parametrų ir įrenginio procesoriaus pajėgumų, todėl transliuojamas vaizdas visada yra geriausios, kokios tik gali būti, kokybės [46]. DASH sudėtyje yra

tiek tradicinių transliavimo, tiek progresyvaus atsisiuntimo elementų. TCP yra transporto protokolas, naudojamas kartu su HTTP.

Stipriai didėjant interneto duomenų perdavimo spartoms, toks multimedijos duomenų transliavimas, perduodant duomenis HTTP duomenų segmentais, įgauna vis didesnę prasmę. HTTP transliavimas gali būti pritaikomas transliavimui pagal paklausą ir tiesioginiam transliavimui. Pagrindinis skirtumas tarp šių būdų yra laikas, kada segmentai yra prieinami. Tiesioginio transliavimo metu, laikas tarp dviejų užklausų yra apytiksliai lygus pirmojo segmento užklausos trukmei [47]. Dar vienas svarbus aspektas, susijęs su DASH, yra tai, jog perduodamų duomenų maršruto tinklo pralaidumas gali būti įvertinamas kiekvienos transliacijos kliento. Yra keletas algoritmų, kurie gali būti panaudojami esamo tinklo pralaidumo tarp galinių mazgų apskaičiavimui [48].

Pagrindinės savybės [49]:

- Adaptyvusis transliavimas – vaizdo transliacijos kokybės reguliavimas, pagal kliento tinklo pralaidumą ir įrenginio procesoriaus pajėgumus;
- Segmentavimas – DASH vaizdo komponentai yra užkoduoti ir išskirstyti į kelis segmentus, kuriuose yra informacija dekoderiui ir medijos segmentai su vaizdo duomenimis;
- Medijos pristatymo aprašymas – MPD (angl. *Media presentation description*) aprašo, kaip segmentai suformuoja vaizdo prezentaciją. Naudojant MPD klientas prašo segmentų sklandaus vaizdo ištransliavimo, o pagal tinklo pralaidumą esamu momentu – tinkamo duomenų perdavimo greičio;
- Kodekų (angl. *Codec*s) nepriklausomumas – DASH yra kodekų agnostikas, o jo pagrindinis konteineris yra MP4 ir MPEG-TS. Protokolas, taip pat, leidžia sklandžiai priimti vis dažniau naudojamus patobulintus HEVC vaizdo kodekus (pavyzdžiui, H265);
- HTTP duomenų transliavime, tinklapis (serveris) dažniausiai turi labai mažai informacijos apie kliento/tinklo būklę;
- Klientas, kad būtų išlaikyta gera paslaugų kokybė (angl. *QoS*), pats priima sprendimus dėl turinio pateikimo.

1.5.7. DASH sauga

Protokolas duomenis perduoda naudojant HTTP, todėl įprastai taikomi tie patys saugumo mechanizmai kaip ir HTTP. Transliuojant vaizdo duomenis naudojant HTTP, medijos duomenų segmentai yra šifruojami naudojant AES algoritmą su 128 bitų raktu, kartu su šifravimo blokų grandinės algoritmu (angl. *CBC*) ir viešųjų raktų infrastruktūros standartu PKCS7. Šifravimo raktai pateikiami naudojant išteklių identifikatorių (angl. *URI*). Raktų pristatymas turi būti apsaugotas mechanizmu, tokiu HTTP su TLS [50].

Saugus HTTP (HTTPS) yra vis sparčiau naudojamas HTTP protokolo variantas, kuris transliuojant vaizdą, leidžia pasiekti didesnį privatumo ir saugumo lygį galiniams vartotojams [51]. Saugus ryšio kanalas HTTPS užtikrinamas naudojant autentifikavimo procesą, kai trečiosios šalies sertifikavimo institucija užtikrina, jog vaizdo transliacijos duomenų perdavimas iš transliuotojo yra autentiškas. Segmento autentifikavimo sistema leidžia naudoti skaitmeninius autentiškumo žymenis visiems DASH segmentų tipams, siekiant patikrinti jų kilmę ir turinio autentiškumą. Ši sistema veikia apskaičiuojant maišos funkcijos vertę, laikant ir gabenant ją išorėje, pavyzdžiui, naudojant HTTPS. Klientas atkurdamas maišos funkcijos vertes arba parašo duomenis ir apskaičiavęs juos lokaliai, lygina reikšmes ir nesutapimo segmentą atmeta [52].

1.5.8. Protokolų funkcionalumo ir savybių palyginimas

Protokolų, sukurtų ir standartizuotų komunikacijai tarp klientų ir transliavimo serverių – vos keletas. Atsižvelgus į jų funkcionalumus, protokolai, tiesiogiai susiję su vaizdo transliacijomis internetu, gali būti suskirstyti į šias kategorijas [35]:

1. Tinklo protokolai teikia tinklo serviso paslaugas, pavyzdžiui, tinklo adresavimą. Interneto protokolas IP įprastai naudojamas, kaip tinklo sluoksnio protokolas vaizdo transliacijoms internetu;
2. Transporto protokolai suteikia tinklo transportavimo funkcijas, transliavimo apklikacijoms. Pagrindiniai transportavimo protokolai, kaip jau minėta 1.5.3. skyrelyje, yra TCP ir UDP. Šie protokolai yra žemesnio sluoksnio transportavimo protokolai, ant kurių diegiami aukštesnieji (angl. *Upper-layer*) transporto protokolai, tokie kaip RTP, RTCP;
3. Sesijos protokolai apibrėžia pranešimus ir procedūras, skirtas valdyti multimedijos duomenų perdavimą sesijos metu. Sesijos protokolai gali būti RTSP ir sesijos inicijavimo protokolas SIP.

Kaip žinoma, TCP yra dominuojantis protokolas, naudojamas duomenų perdavimui internete, todėl nenuostabu, jog TCP naudojamas ir vaizdo duomenų transliacijoms. Tačiau, būtina paminėti problemas, su kuriomis susiduriama transliuojant vaizdo duomenis naudojant TCP. Pirmoji problema yra duomenų perdavimo spartos reguliavimas. Perduodant duomenis TCP Internete, duomenų sparta kinta vadinamuoju „saw-tooth“ principu, tai yra AIMD (angl. *Additive-increase/multiplicative-decrease*) algoritmu, kuris naudojamas TCP perkrovos kontrolės mechanizme. Antroji problema yra vėlavimas dėl retransliavimo, atliekamo tarp galinių mazgų tuo pačiu metu. Šias problemas, iš dalies, apsprendžia tinkamas duomenų buferio pildymas (angl. *Buffering the data*).

Vienas iš esminių aspektų, kodėl TCP nėra tobulai pritaikytas vaizdo duomenų transliavimui – prarastų paketų retransliavimo mechanizmas, kuris sukelia vaizdo transliacijos vėlavimą ir netolygumą.

UDP yra kitas transporto protokolas, naudojamas vaizdo duomenų perdavimui. UDP leidžia paketams būti išmestiems, jeigu jų galiojimas yra pasibaigęs arba jie yra pažeisti. Tai reiškia, jog klientas galės matyti vaizdą transliacijai nesustojant, net jeigu vaizdo duomenys bus apgadinti. Tai galima laikyti ir problema ir pranašumu. Būtina paminėti, jog perduodant vaizdo duomenis su UDP, dauguma tinklo ugniasienių blokuoja juos.

Puiki alternatyva vaizdo transliavimui yra realaus laiko tranporto protokolas RTP kartu su jo kontroliniu protokolu RTCP. RTP suteikia laiko atkūrimo, praradimų aptikimo, saugumo ir turinio identifikavimo funkcijas. Kontrolinė RTCP dalis leidžia identifikuoti šaltinį ir suteikia paramą šliuzams (angl. *Gateways*), pavyzdžiui, vaizdo ir garso tilteliams, grupinio-vienatipio transliavimo vertėjams. RTCP suteikia galimybę gauti serviso kokybės atsiliepimus iš gavėjų ir grupių, palaiko skirtingų medijos srautų sinchronizavimą.

RTP/RTCP veikia kartu su UDP ir suteikia nemažai funkcijų ir lankstumo medijos duomenų transportavimui, tačiau RTP negali garantuoti serviso kokybės, nenurodo rezervacijos ir nepalaiko formatų suderinamumo [53], taip pat, yra su pažeidžiamu saugumu.

Pastaruoju metu, dėl debesijos serverių paplitimo, HTTP/TCP protokolų architektūra vaizdo transliavimui naudojama vis dažniau, paliekant užnugaryje klasikinę RTP/UDP protokolų

architektūrą. RTP buvo sukurtas apibrėžiant garso ir vaizdo turinio paketų formatus kartu su duomenų srauto sesijos kontrole, kas leido pristatyti duomenis su mažais resursų ištekliais (angl. *Low Overhead*). Tačiau, didėjant interneto duomenų perdavimo srautų pralaidumams, šis RTP privalumas perduoti vaizdo duomenis mažais paketais tapo nebe toks aktualus. Dabar multimedijos duomenys gali būti efektyviai pristatomi didesniuose segmentuose, naudojant HTTP. Žinoma, kalbant apie vaizdo duomenų transliavimą iš ribotus išteklius turinčių įrenginių, RTP turi pranašumą prieš HTTP vaizdo duomenų transliavimo metodą.

Lyginant šiuos protokolus, juos galima išskaidyti į „push-based“ ir „pull-based“ tipus. Minėti „push-based“ protokolai sudaro ryšio seansus tarp kliento ir serverio, kur klientas yra atsakingas už ryšio užmezgimą, tuo tarpu serveris siunčia duomenų srautus, kol klientas sustabdo arba nutraukia komunikavimą. Serveris išlaiko sesiją, kad galėtų klausyti komandų iš kliento. Tai protokolai, dažniausiai naudojantys UDP transporto lygmenyje, tačiau galintys veikti ir TCP, pavyzdžiui, realaus laiko transporto protokolas RTP.

„Pull-based“ protokolai yra paremti HTTP protokolu, sudaro ryšį tarp kliento ir serverio, kur klientas yra atsakingas už prašymo išsiuntimą serveriui, tuo tarpu serveris užmezga ryšį, o klientas gauna vaizdo duomenis. Šio tipo protokoluose, vaizdo transliavimo duomenų perdavimo spartos, priklauso nuo kliento tinklo pralaidumo. „Pull-based“ protokolo pavyzdys – DASH [54].

RTP transliavimo metu serveris turi valdyti atskiras transliavimo sesijas kiekvienam klientui, todėl atliekant transliacijas didesniu mastu, ištekliai gali būti naudojami labai intensyviai, taip pat, nevisi serveriai palaiko RTP protokolą . HTTP atveju, klientas pats valdo srautą ir neprivalo išlaikyti sesijos su serveriu, serverio diegimas yra paprastesnis, o transliavimas didesniam kiekiui klientų neiššaukia jokių papildomų nuostolių serverio resursams [55].

Nepaisant to, jog naudojant “Pull-based” protokolus gavėjo išlaidos yra mažesnės, tačiau vaizdo transliavimas yra mažiau efektyvus, turint omenyje, naudojamus resursus, dėl pagrindinio transliavimo protokolo TCP. Lyginant su HTTP/TCP, protokolų rinkinys RTP/UDP imponuoja mažesnius bendruosius duomenų perdavimo resursus (angl. *Overhead*), tačiau veikiant kartu su UDP, prarandama duomenų retransliavimo dinamika ir perkrovos kontrolės mechanizmai [56], kurie, siekiant mažesnio vaizdo vėlavimo ir taupesnio resursų naudojimo, gali būti laikomi, kaip neigiami mechanizmai transliacijai.

Pateikiama 1.1 lentelė, kurioje apibendrinamos pagrindinės vaizdo transliavimo protokolų savybės, jų privalumai ir trūkumai.

1.1 lentelė. Protokolų funkcionalumų palyginimas

Protokolas /Protokolų architektūra	**Privalumai**	**Trūkumai**
TCP	– Dominuojantis protokolas duomenų perdavimui Internete; – Transliavimas pro ugniasienes; – Patikimas.	– Paprastai reikia didelio buferio, kuris reikalingas duomenų perdavimo spartos kitimui valdyti; – Prarastų paketų sugražinimui reikalingas retransliavimas, kuris sukelia vėlavimus; – Nepalaiko grupinio transliavimo.

UDP	– Pritaikomas vaizdo transliavimui; – Praradus paketus, transliacija nenutraukiama (tęsiama); – Nereikalingas paketų retransliavimas.	– Daugumos tinklų ugniasienės blokuoja UDP duomenis; – Reikalinga klaidų kontrolė, prarandant paketus; – Nepalaikomi perkrovos kontrolės mechanizmai; – Negalima transliuoti naudojant populiarius grotuvus, pavyzdžiui, *QuickTime*.
RTP/RTCP	– Skirtingų pobūdžio apklikacijoms apibrėžia ir leidžia pritaikyti skirtingus profilius; – Palaiko realaus laiko transliacijas; – Suteikia laiko atstatymo, prarastų paketų aptikimo, funkcionalumus (RTCP); – Leidžia išgauti įdomią ir naudingą tinklo statistiką (RTCP);	– RTP negali garantuoti serviso kokybės (QoS). Tam naudojamas RTCP; – Naudoja silpną šifravimo algoritmą; – Neturi autentifikavimo mechanizmų; – Antraštė didesnė nei UDP; – Žymiai sudėtingesnės sandaros nei UDP; – Nepalaikomi perkrovos kontrolės mechanizmai.
DASH	– Suteikia patikimumo, kadangi naudojamas kartu su TCP; – Suderinamas su duomenų perdavimu pro tinklo ugniasienes; – Palaikomi perkrovos kontrolės mechanizmai; – Paprastai įdiegiamas ir plačiai naudojamas; – Taikomi panašūs saugumo mechanizmai kaip ir HTTP protokole.	– Esant nepatikimam ar prastesnių parametrų internetiniam ryšiui, paketų praradimas, jų retransliavimas ir didelis klaidų pranešimų skaičius, gali lemti vaizdo transliacijos trikdžius, ypatingai transliuojant didelius duomenų kiekius; – Didesnis vėlavimas; – Dėl griežtų buferio reikalavimų, paketų dydžio ir kadrų intervalo transliacija gali atsilikti iki minutės; – Sunkiai pritaikomas ribotų išteklių įrenginiuose, daugiau naudojamas žiniatinklio programose.

1.5.9. Vaizdo transliavimo protokolų pritaikymas ribotų išteklių įrenginiuose

Autonominių savybių turinčiuose įrenginiuose, komunikacija įprastai atliekama naudojant belaidžio ryšio technologijas. IoT ir kiti ribotų resursų įrenginiai turi ribojimus, turint omenyje, tokius parametrus, kaip tinklo pralaidumas, skaičiuojamoji galia, energijos suvartojimas ir kt. Suprantama, jog transliuojant vaizdą, visi šie ištekliai ir apkrovos stipriai padidėja, todėl įprastiniai duomenų perdavimo, vaizdo transliavimo ir kiti protokolai, duomenų apsaugojimo mechanizmai, kurie taikomi įprastinių kompiuterių sistemose, dažnai negali būti pritaikomi ribotus išteklius turinčiuose įrenginiuose. Dėl šios priežasties yra būtinos protokolų, šifravimo ir kitų apsaugos algoritmų modifikacijos, siekiant saugiai perduoti vaizdo duomenis, pavyzdžiui, iš vaizdą transliuojančio autonominio aplinkos stebėjimo roboto.

Pasvėrus ir atidžiai įvertinus įrenginio ir tinklo galimybes, apsvarsčius, kokiam tikslui atliekama vaizdo transliacija, kokie ir kokios svarbos duomenys yra perduodami, būtina apsibrėžti svarbiausius transliacijos parametrus:

- Vėlavimo trukmė;
- Vaizdo kokybė;
- Greitaveika;
- Saugumas.

Visų šių parametrų užtikrinimas naudoja resursus, energiją ir atmintį, todėl labai svarbu tinkamai balansuoti tarp kokybės, saugumo rodiklių ir kitų parametrų, atsižvelgiant į įrenginio charakteristikas ir vaizdo transliacijos paskirtį.

Kalbant apie vaizdo transliaciją iš ribotų išteklių įrenginių, kelios savybės yra itin svarbios. Sėkmingos transliacijos įgyvendinimui reikalingas lankstumas, kuris leidžia transliuoti daugeliui, skirtingas charakteristikas turinčių įrenginių. Būtina paminėti, jog transliacija turi būti įgyvendinama optimizuotai, kadangi turi būti kiek įmanoma efektyviau panaudoti turimi tinklo ir įrenginio resursai. Žinoma, ypatingai bevieliuose tinkluose, negalima apseiti ir be saugumo, kuris užtikrina, jog perduodamas turinys yra apsaugotas nuo neteisėto informacijos perėmimo (angl. *Eavesdropping*) ir kitų atakų.

1.5.10. Saugumo užtikrinimas vaizdo transliacijose

Perduodamų vaizdo duomenų konfidencialumui ir vientisumui užtikrinti, įprastai naudojami autentifikavimo ir kriptografijos mechanizmai. Gerai suprojektuotas autentifikavimo metodas užtikrina, jog duomenys bus prieinami tik asmenims, turintiems tam įgaliojimą. Kriptografijos mechanizmai užtikrina, jog jautri ir privati informacija yra apsaugota ją perduodant, kaupiant ar apdorojant.

Dėl turimų ribotų skaičiavimo pajėgumų, atminties ir ribotos baterijos talpos, daugelis daiktų interneto įrenginių nėra pajėgūs įgyvendinti tradicinių autentifikavimo ir saugos mechanizmų.

Atliekant vaizdo transliaciją, svarbu suprasti esmines saugumo strategijas ir metodus, kurie padėtų įgyvendinti vaizdo transliacijos apsaugą, ribotus išteklius turinčių įrenginių terpėje. Pagrindiniai saugos elementai, reikalingi įgyvendinti saugą vaizdo transliacijoje, aprašomi toliau. Saugaus komunikavimo metodo projektavime svarbu nustatyti, koks saugos lygis reikalingas, remiantis aplinka, perduodamu duomenų pobūdžiu, potencialaus nusikaltėlio gebėjimais. Nuo to priklausys, kokie saugos mechanizmai bus naudojami transliacijoje, kadangi kiekvienas iš jų „kainuoja“ tam tikrus resursus.

1. **Raktų valdymas.** Raktų valdymo infrastruktūra yra viena iš sudėtingiausių kriptografinio mechanizmo dalių, todėl, nenuostabu, jog ši dalis yra viena svarbiausių. Dažniausiai, įsilaužėliai perima duomenis gavę slaptus raktus, todėl jų apsikeitimas ir laikymas turi būti apsaugotas. Raktai gali būti skirtingų tipų, priklausomai nuo naudojamo šifravimo algoritmo, pavyzdžiui, simetriniai raktai (AES, 3DES), privatūs raktai (RSA, ECDSA), maišos funkcijoms (HMAC-SHA-256) apskaičiuoti skirti raktai;
2. **Autentifikavimas.** Autentifikavimas įgyvendinamas skirtingomis metodikomis. Ši funkcija gali būti atliekama panaudojant ID-slaptažodžio identifikavimą tarp serverio ir kliento, prieigos kontrolės mechanizmus, sertifikatus, pavyzdžiui, X.509, maišos funkcijas ir kitus kriptografinius metodus;
3. **Šifravimas.** Perduodant duomenis iš įrenginių, reikalingas stiprus duomenų šifravimas. Paprastai, mažai vietos užimančių duomenų, tokių kaip raktai, valdymo komandos, perspėjimai, šifravimas

ribotų išteklių ir kituose daiktų interneto įrenginiuose gali būti pilnai įgyvendinamas. Tuo tarpu, daugialypės terpės, multimedijos duomenys sudaro milžiniškus duomenų srautus. Tokių duomenų pilnavertis šifravimas gali perkrauti procesorių, jeigu jis nėra pajėgus dirbti su tokiu sudėtingu ir daug išteklių reikalaujančiu procesu. Ryšys tarp procesoriaus galios, skaičiojamosios atminties ir energijos suvartojimo apibrėžia pagrindinę problemą, su kuria susiduriama, norint atlikti visavertį, stiprų duomenų šifravimą, ribotų išteklių įrenginiuose. Ši problema verčia ieškoti kitų šifravimo metodikų ir technikų (pavyzdžiui, DTLS ar TLS modifikacijos, selektyvusis šifravimas, iš anksto pasidalinto rakto technika), kurios būtų „lengvesnės“ resursų atžvilgiu ir galėtų būti pritaikomos IoT įrenginių grupei [57].

1.5.11. TLS/DTLS naudojimas ribotų išteklių įrenginiuose

TLS saugumo protokolas, kuris užtikrina saugų komunikavimą tarp įrenginių tinkluose ir internete. Jis yra rekomenduojamas daugelio standartų, kurie yra patvirtinti IETF, tačiau TLS nėra išmintingiausias pasirinkimas saugumui užtikrinti IoT įrenginiuose [58].

Įprastai TLS veikia su patikimu transporto protokolu TCP, kuris, kaip žinia, nėra geriausias pasirinkimas perduodant vaizdo duomenis iš ribotus išteklius turinčių įrenginių. Kaip alternatyva, naudojamas DTLS protokolas, kuris veikia kartu su UDP ir užtikrina tokio pačio lygmens saugos funkcijas, kaip ir tradicinis TLS. DTLS ir TLS protokolai buvo kurti standartiniams Interneto įrenginiams ir nebuvo siekiamybės jų pritaikyti ribotų išteklių įrenginiams, todėl šiais laikais, aktyviai plečiantis daiktų interneto įrenginių infrastruktūrai, ieškoma būdų kaip pritaikyti šiuos patikimus, saugumą užtikrinančius protokolus ribotų išteklių įrenginiams.

„Brangiausias“ dalykas, turint omenyje, įrenginio ir tinklo resursus, yra sertifikatų ir raktų naudojimas. Siūlomi šie būdai, kurie gali padėti sumažinti energijos suvartojimą, atliekant viešojo rakto operacijas [59]:

1. **Įranga kriptografijos operacijų akceleracijai**. Tai įranga, kuri yra atsakinga už visas kriptografines operacijas ir skaičiavimus. Pavyzdys gali būti TPM lustas (angl. *Trusted Platform Module*), kuris generuoja kriptografinius raktus, juos saugo ir atlieka kriptografinius algoritmus. Tokios įrangos naudojimas užtikrina aukštą saugumo lygį su sąlyginai mažu resursų panaudojimu, mažu vėlavimu ir optimaliu duomenų vietos panaudojimu. Deja, toks sprendimas yra labai brangus ir sudėtingiau įgyvendinamas ribotų išteklių įrenginių grupėse;
2. **Naudojamų protokolų modifikavimas**. Siekiant optimizuoti resursų naudojimą, siūlomos įvairios tradicinių saugumo protokolų modifikacijos, kurios yra labiau pritaikytos ribotus išteklius turinčių įrenginių terpei. Rene Hummen ir kitų mokslininkų [60] pasiūlyta modifikacija optimizuoja DTLS pirminio rankos paspaudimo procedūrą, sumažinant reikalingą skaičiuojamąją galią, apdorojimo laiką ir atmintį. Sangramo Rėjaus ir G. P. Biswaso [61] pasiūlytas IKE protokolo variantas naudoja eliptinių kreivių kriptografijos (ECC) viešo rakto sertifikatą ir eliptinių kreivių Diffie-Helmano (ECDH) raktų apsikeitimo mechanizmą, vietoje RSA ir DH protokolų. Ši variacija sumažina skaičiavimo operacijų galios sąnaudas, kadangi jos apribotos tik taško multiplikacinėmis operacijomis, o raktai, užtikrinantys tokią patį saugumo lygį, gali būti mažesnio dydžio nei RSA;
3. **Autentifikavimas alternatyviais būdais**. Atlikti autentifikavimą tarp kliento ir serverio galima nebūtinai naudojant PKI kriptografinius mechanizmus. Šią procedūra galima įgyvendinti panaudojant protokolų struktūros laukus, autentifikavimo funkcionalumams diegti. Įprastai

diegiamam autentifikavimui naudojami laukai, kurie kuriamos sistemos funkcionalumui nėra reikalingi ar jų nebuvimas nedaro pastebimos žalos protokolo veikimui. Dažniausiai, tai būna specialios autentifikavimo žymos, maišos funkcijų reikšmių pavidalu. Tai užtikrina, jog gautas paketas ar signalas yra iš laukiamo ir sutarto siuntėjo, o ne nusikaltėlio. Ši metodika apsaugo nuo susidūrimo viduryje atakos, kurios metu nusikaltėlis perėmęs duomenis galėtų siųsti „savus“ duomenis, prisidengęs siuntėjo tariama identifikacija. Būtent tarp ribotus resursus turinčių įrenginių, autentifikavimas naudojant sertifikatų ir raktų apsikeitimo infrastruktūrą yra svarstytinas ir priimtinas tik išskirtiniais, retais atvejais.

Egzistuojant tiesioginiam sąryšiui tarp įrenginio pajėgumų ir sunaudojamos galios, renkantis saugos protokolus, būtina į tai atsižvelgti. Įprastai, sertifikatų ir raktų apsikeitimas autentifikavimui nėra taikomas išmaniuosiuose namų įrenginiuose, būtent dėl ribotų resursų, kur stinga pajėgumų sunkių kriptografinių operacijų atlikimui [62].

1.6. Analizės išvados

1. Autonominių robotų panaudojimo sričių atsiranda vis daugiau, pradedant namų ūkiu, tęsiant medicina, baigiant gynybos sektoriumi. Dėl plataus spektro robotų atliekamų funkcijų, sąveika tarp žmonių ir robotų tendencingai didės. Atsižvelgus į tyrimus, didžioji dalis gaminamų autonominių robotų yra su silpnais kibernetinio saugumo mechanizmais arba be jų, todėl sukompromitavus roboto veikimą, kyla pavojus žmonių privatumui ir saugumui.
2. Daugelį funkcijų autonominiai robotai atlieka geriau nei žmonės, todėl jiems patikimos daug atsakomybės nešančios užduotys, kurios yra susijusios su privačiais ir jautriais duomenimis. Apžvelgus keletą robotų panaudojimo sričių, kur roboto saugumas užima svarbų vaidmenį, daroma prielaida, jog didžioji dalis atakų įgyvendinamos nuotoliniu būdu, pasinaudojus nesaugaus ryšio kanalo pažeidžiamumo vektoriais.
3. Palyginus standartinius transporto lygmens protokolus TCP ir UDP, nustatyta, jog UDP yra labiau priimtinas vaizdo duomenų transliavimui internete, kadangi jis pasižymi mažesniu vėlavimu, palaiko daugiaabonentes transliacijas, yra greitesnis, tačiau ne toks patikimas.
4. Išanalizavus RTP ir DASH multimedijos duomenų protokolus, nustatyta, jog RTP yra žymiai universalesnis, greitesnis, pasižymintis mažu vėlavimu ir pritaikomas platesnei terpei įrenginių, tačiau naudoja DES šifravimo algoritmą, kuris pripažintas pažeidžiamu. DASH protokolas patikimas, paprastai diegiamas, naudoja panašius saugumo mechanizmus kaip HTTP, tačiau naudojamas galingų išteklių tinkluose ir sunkiai pritaikomas ribotų išteklių įrenginiams.
5. Atsižvelgiant, kur bus naudojamas autonominis aplinkos stebėjimo robotas, kokią užduotį jis atliks ir kokios svarbos duomenys bus transliuojami, projektuojant protokolų architektūrą, svarbu apsibrėžti, kokie vaizdo transliavimo parametrai turi būti užtikrinami, koks duomenų saugumo lygis yra reikalingas. Visa tai turi būti vertinama, atsižvelgus į stipriai ribotus įrenginio resursus, kurie turi būti panaudoti kaip galima efektyviau.
6. Autonominiai aplinkos stebėjimo robotai susiduria su identiškomis problemomis, su kuriomis susiduria ir kiti daiktų interneto įrenginiai. Dėl silpnų skaičiavimo pajėgumų, atminties ir energijos trūkumo, tradiciniai, stiprūs kriptografiniai saugos mechanizmai, kurie naudojami įprastose kompiuterinėse sistemose, negali būti pritaikyti ribotus išteklius turintiems įrenginiams. Dėl to, apsaugotai vaizdo transliacijai įgyvendinti, būtinos reikalingų protokolų ir saugumo mechanizmų modifikacijos, kurios padėtų optimaliai išnaudoti įrenginio pajėgumus. Tokiu atveju, jeigu stiprių kriptografinių operacijų įrenginiui pritaikyti neįmanoma, saugumas turi būti užtikrinamas silpnesniais saugumo mechanizmais.

2. Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodas

Analitinėje darbo dalyje buvo nuodugniai išnagrinėti autonominiai robotai, jų tipai, taikymo sritys, saugumo problemos. Išnagrinėtos dažniausiai pasitaikančios atakos prieš robotus, saugumo spragos. Ypatingai, atsižvelgta į autonominius aplinkos stebėjimo robotus, kadangi šio tipo robotai yra pagrindinis darbo tyrimo objektas. Išnagrinėtos vaizdą transliuojančių robotų charakteristikos, panaudojimo aplinkos, perduodamų duomenų svarba, atliktas detalus, vaizdo transliacijoms naudojamų protokolų, tyrimas ir jų funkcionalumų, saugumo palyginimas.

Po atliktos analizės nustatyta, jog didžioji dalis atakų prieš autonominius aplinkos stebėjimo robotus įgyvendinamos dėl nesaugaus ryšio kanalo. Tai gali būti paaiškinama tuo, jog įprastai robotų saugumo mechanizmai yra pamiršti robotų gamintojų, nes didžioji dalis robotų resursų, kurie yra ir taip stipriai riboti, yra išnaudojami patikėtam funkcionalumui atlikti.

Apžvelgti būdai ir pagrindiniai saugos mechanizmai, kurie gali padėti užtikrinti perduodamų vaizdo duomenų saugumą. Nustatyta, jog dėl ribotų roboto resursų, tai yra atminties, energijos, mažų skaičiavimo pajėgumų, tinklo pralaidumo, didžioji dalis įrenginių, atliekant vaizdo transliaciją, nėra pajėgūs naudoti visaverčių apsaugos mechanizmų ir saugiausių kriptografinių algoritmų.

Pagrindinė darbo užduotis yra sukurti saugaus komunikavimo metodą tarp vaizdą perduodančio roboto ir galinio įrenginio. Šis metodas turi užtikrinti, jog, dažnai, gana privačiose aplinkose veikiančių, aplinkos stebėjimo robotų perduodami jautrūs vaizdo duomenys nebūtų nutekinti pašaliniams asmenims.

Uždavinys įgyvendinamas tinkamai suprojektavus protokolų architektūrą, kuri leistų pasiekti reikalingus vaizdo transliacijos parametrus, atsižvelgus į roboto darbo aplinką, įrenginio resursus, perduodamų duomenų jautrumą, reikalingą vaizdo transliacijos kokybę, galimą vėlavimą ir kitas charakteristikas. Atkreipus dėmesį į minėtus kriterijus, vaizdo transliacijos sauga bus įgyvendinama pasitelkus RTP profilį – SRTP, atlikus keletą modifikacijų jo veikime. Tyrimo eigoje, bus pasitelkiamas DTLS protokolas, kuris galėtų pasiūlyti tokius funkcionalumus, kaip labai stiprus duomenų šifravimas, autentifikavimas, raktų apsikeitimas sesijos metu. Tai leistų užtikrinti perduodamų duomenų vientisumą ir konfidencialumą, tačiau galimai kainuotų daugybę resursų namų aplinkoje, kur duomenys nėra kritinės svarbos.

2.1. Siekiami rezultatai ir pagrindinės saugaus komunikavimo metodo savybės

2.1.1. Autonominio aplinkos stebėjimo roboto panaudojimas

Kaip minėta pirmame skyriuje, projektuojant vaizdo duomenų perdavimą saugiu ryšio kanalu, labai svarbu įvertinti, kokioje aplinkoje robotas dirbs ir kokią užduotį atliks. Nuo to priklausys reikiamas saugumo lygio užtikrinimas ir vaizdo transliacijos kokybės rodikliai.

Šiuo atveju, robotas būtų naudojamas namų aplinkoje, kur fiksuojami duomenys yra privatūs ir jautrūs, todėl jie neturėtų būti prieinami pašaliniams asmenims. Žinoma, tai nėra kritinės infrastruktūros duomenys, kurie gali kainuoti gyvybę ar sukelti milžiniškus finansinius nuostolius. Numatoma, jog robotas atliks namų apsaugos funkciją. Numatomi vaizdo transliavimo kokybės parametrai yra filmavimas 720P raiška ir palaikant 30 kadrų per sekundę (angl. *FPS*). Tokie transliacijos parametrai yra daugiau nei pakankami, potencialiam įsilaužėliui identifikuoti. Kadangi įrenginio resursai yra riboti, autonominis aplinkos stebėjimo robotas pradės vaizdo transliaciją į

numatytą galinį įrenginį tik užfiksavus judesį, pavyzdžiui, įėjimo į namus zonoje. Tokiu būdu būtų išvengta nuolatinio vaizdo transliavimo, kuris gali būti nepriimtinas įrenginiui, kurio skaičiavimo pajėgumai yra maži, o energija – ribota. Taip pat, nuolatinis transliavimas, dažnais atvejais, gali būti nerezultatyvus. Sėkmingai įgyvendinus transliaciją, robotas pereis į ramybės režimą, o po 60 sekundžių grįžta į budėjimo padėtį. Tai reiškia, jog po atliktos riboto laiko vaizdo transliavimo operacijos, pakartotina vaizdo transliacija bus galima tik po mažiausiai 60 sekundžių. Tokiu būdu bus sumažintas klaidingų transliacijų skaičius.

Galima situacija galėtų būti, kai pašalinis asmuo įsilaužia į namus, robotas užfiksuoja judesį ir nedelsdamas pradeda vaizdo transliaciją. Tokios paskirties vaizdo transliacija vyktų trumpą laiko atkarpą, kadangi pagrindinė funkcija yra užfiksuoti galimo nusikaltėlio veido bruožus ir kuo greičiau įspėti, ištransliuoti vaizdą gavėjui, pavyzdžiui, namo šeimininkui ar apsaugos darbuotojui. Galutiniai duomenys būtų išsaugomi stebėtojo įrenginyje, tokiu atveju, jeigu gyvos transliacijos metu nebūtų operatyviai sureaguota į transliuojamą vaizdo medžiagą. Suprantama, jog vaizdo kokybė tokiai užduočiai vykdyti turėtų būti kaip galima geresnė, vaizdo transliacija turėtų netrūkinėti, o dėl realaus laiko transliacijos poreikio, vaizdo transliacijos vėlavimas turėtų būti kuo minimalesnis, kadangi kiekviena sekundė gali daug ką lemti.

Vertėtų nepamiršti, jog robotas fiksuos judesį ir pradės vaizdo transliaciją ne tik įsilaužus pašaliniams asmenims, tačiau ir įeinant / išeinant namų šeimininkui. Perėmus tokios vaizdo transliacijos duomenis, nusikaltėlis sužinotų, kada šeimininkas išvyksta iš namų ir palieka juos tuščius. Taip pat, nesant transliacijos autentifikavimo, nusikaltėlis gali nukreipti vykdomą transliaciją kita linkme, vykdant susidūrimo viduryje (angl. *MITM*) ataką, tokiu būdu namo šeimininkui nebūtų ištransliuotas vaizdas įsilaužimo metu.

2.1.2. Saugumo įgyvendinimas

Kaip minėta 2.1.1. skyrelyje, reikalinga užtikrinti, jog perduodami duomenys būtų nepasiekiami pašaliniams asmenims, tai yra, šifruojami, o transliacija vykdoma saugiu ryšio kanalu autentifikavus duomenis. Visa tai leistų užtikrinti perduodamų vaizdo duomenų vientisumą ir konfidencialumą. Remiantis transliacijos kriterijais ir analizės išvadomis, šiuos funkcionalumus įvykdyti leistų transporto lygmens saugumo protokolai TLS ar DTLS poroje su RTP arba saugesniu RTP protokolo profiliu – SRTP. Konfidencialumas ir vientisumas užtikrinamas naudojant kriptografinius algoritmus, maišos funkcijas. Autentfikavimas gali būti įgyvendinamas naudojant skaitmeninių sertifikatų ir raktų infrastruktūrą, kurią siūlo DTLS protokolas arba kitais metodais, pavyzdžiui, naudojant žinutės autentifikavimo maišos reikšmę paketo antraštėje. Tačiau, kaip jau aptarta, būtent sertifikatų ir raktų naudojimas yra kritinis, kalbant apie įrenginio pajėgumų naudojimą. DTLS operacijas, sesijos inicijavimas (angl. *Handshake*) yra labai brangus, kalbant apie resursų sunaudojimą siunčiant mažus duomenų srautus, tačiau siunčiant didesnius duomenų srautus, šis žymus resursų sunaudojimas amortizuojasi per laiką [63]. AASR sistemos koncepcijoje operacijos laikas yra 15 sekundžių, todėl šis laikas nėra pakankamas kompensuoti DTLS rankos paspaudimo operacijos staigaus ir žymaus resursų sunaudojimo. Prabrėžtina, jog tokia pajėgumų sunaudojimo kaina gali stipriai pabloginti įrenginio veikimą, ypač jeigu sesijos laikas yra trumpas, kas galioja roboto vaizdo transliacijos atveju.

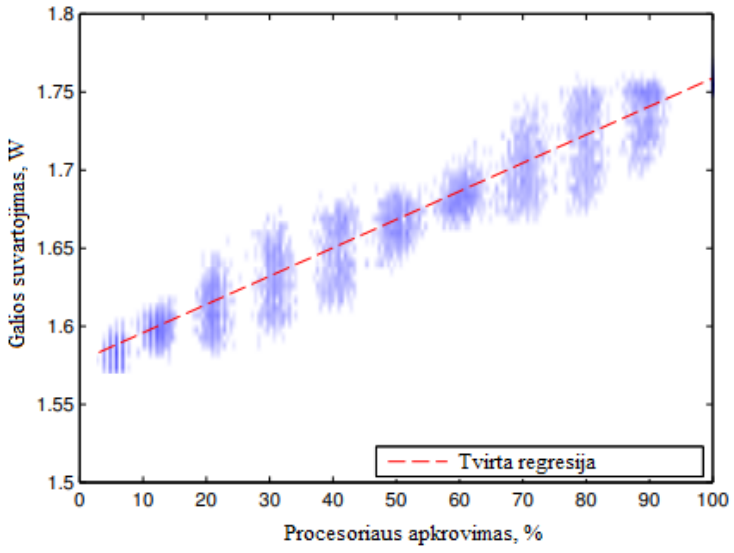
Deja, bet DTLS ir TLS buvo sukurti standartiniams interneto įrenginiams, o ne ribotus išteklius turintiems prietaisams. Kontrolinės komandos, įspėjimai, pranešimai gali būti pilnai užšifruoti net ir ribotų išteklių įrenginiuose. Tačiau, vaizdo duomenų šifravimas reikalauja daug skaičiuojamosios

galios, kuri reikalinga procesams apdoroti, todėl siekiant juos pritaikyti ribotų resursų įrenginiams, būtinas šių protokolų konfigūracijos keitimas ir modifikavimas juos „palengvinant“. Priešingu atveju, turi būti naudojami kiti saugumo mechanizmai, kurie yra pasverti pagal reikalaujamą saugumo lygį, resursų kiekį, tinklo pajėgumą.

Kaip jau minėta 1.5.4. skyrelyje, RTP protokolas turi daug skirtingų profilių ir yra lengvai modifikuojamas, kas leidžia šį protokolą pritaikyti platesniam spektrui įrenginių. Viena iš modifikacijų, kuri „palengvina“ šį protokolą, sumažinant reikalaujamus skaičiavimo išteklius ir suteikia papildomų saugumo funkcijų – SRTP. Tinkamai sukonfigūruotas ir pritaikytas SRTP protokolas puikiai tinka ribotų išteklių įrenginiams ir užtikrina duomenų autentiškumą, vientisumą ir konfidencialumą.

2.1.3. Riboti skaičiavimo ir energijos ištekliai

Multimedijos duomenų kodavimo, suspaudimo, perdavimo operacijos reikalauja daugybės skaičiuojamosios galios, kadangi perduodami didžiuliai duomenų srautai. Tokių duomenų šifravimas stipriai apkrauna procesorių, nes atliekamos sunkios ir kompleksiškos kriptografinės operacijos. Dėl to, jog saugaus komunikavimo metodas taikomas ribotų išteklių įrenginių grupei, būtina atsižvelgti, jog saugus, visavertis, įprastinėse sistemose naudojamas duomenų šifravimas gali būti neįmanomas. Žinoma, jog procesoriaus apkrovimas yra tiesiogiai susijęs su energijos suvartojimu. Fabian Kaup ir kiti mokslininkai atliko tyrimą [64], orientuotą būtent į „Raspberry PI“ procesoriaus apkrovimo ir energijos suvartojimo priklausomybę. Iš priklausomybės grafiko matyti (žr. 2.1 pav.), jog energijos suvartojimo priklausomybė nuo procesoriaus apkrovimo yra praktiškai tiesinė.



2.1 pav. Galios suvartojimo nuo procesoriaus apkrovimo priklausomybė [64]

Numatomas autonominio aplinkos stebėjimo robotas turės ribotus energijos rezervus, kadangi energiją naudos iš baterijos, todėl nuo to priklausys galimų šifravimo ir autentifikavimo operacijų pobūdis.

2.1.4. Perduodamų duomenų vėlavimas

Pagal roboto atliekamą užduotį, apsaugota vaizdo transliacija turi pasižymėti kuo mažesniu vėlavimu, kadangi transliacija yra aktuali realiu laiku. Autonominis aplinkos stebėjimo robotas, galimai, aplinką

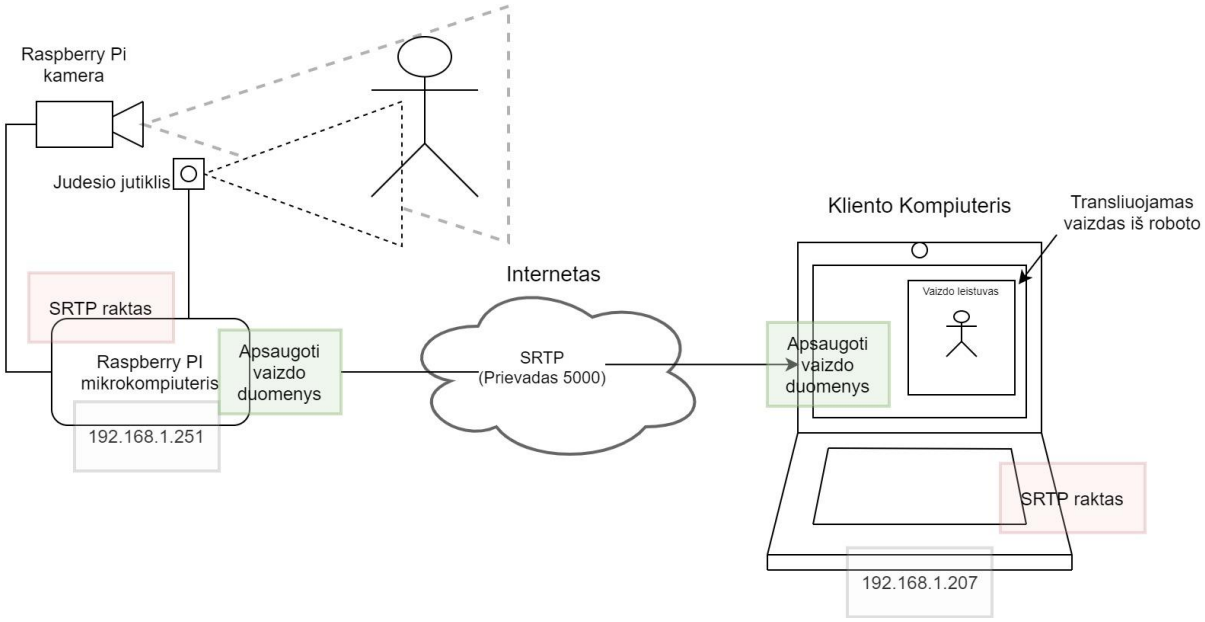
stebės nutolus nuo tinklo maršrutizatoriaus, pavyzdžiui, prie įėjimo į patalpą, tinklo pralaidumas ir ryšio signalas gali būti stipriai riboti, naudojant belaidį ryšį. Norint, jog tiesioginė transliacija būtų atliekama su kuo mažesniu vėlavimu, kadrai turi būti rodomi tam tikru dažniu, be to, siunčiami ir priimami šifruoti paketai turi būti išsiųsti per tam tikrą laiko tarpą, kad būtų išlaikomas priimtinas transliacijos delsimas. Sudėtingos kriptografines operacijos (šifravimas, raktų apsikeitimas) užtrunka nemažai laiko, todėl galimas transliacijos pradžios ir vaizdo vėlavimas didėja. Dėl to, svarbu atsižvelgti, kokius šifravimo algoritmus ar raktų apsikeitimo mechanizmus naudoti. Pavyzdžiui, įrodyta, jog šifruojant būtent multimedijos duomenis, AES gali atlikti šifravimą efektyviau nei RC4 ir XOR algoritmai, dėl to gaunamas mažesnis transliacijos vėlavimas galiniame įrenginyje [65]. Būtent šį šifravimo algoritmą siūlo SRTP protokolas.

2.2. Roboto serverio-kliento architektūra

Atsižvelgiant į 2.1.1. skyrelyje aptartą roboto atliekamą užduotį, transliacijos kriterijus ir jo veikimo terpę, planuojama duomenis perduoti belaidžiu ryšiu galiniam įrenginiui – klientui. Tuo tarpu robotas atliks transliavimo serverio vaidmenį.

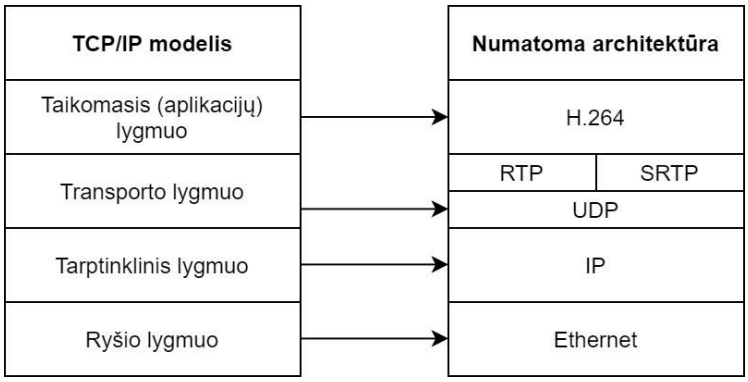
Šiame darbe, vaizdo transliacijai įgyvendinti, geriausiai tiktų saugesnis ir labiau tinkantis ribotų resursų įrenginių terpei RTP profilis – SRTP. Atlikus keletą modifikaijų protokolo veikime ir pritaikius veikimą, atsižvelgiant į ribotus išteklius, šis komunikavimo metodas turėtų būti pakankamai saugus ir efektyvus. Taip pat, analizė parodė, jog RTP ar SRTP įprastai naudojamas su UDP transporto protokolu, kuris pasižymi savybėmis, tinkančiomis vaizdo transliacijoms pagal nustatytas sąlygas.

Bendroji taikymo srities architektūra pateikta 2.2 paveiksle. Čia robotas atliks transliuotojo vaidmenį, tuo tarpu kompiuteris bus stebėtojas (klientas). Apsaugoti vaizdo transliacijos duomenys, saugaus komunikavimo metode, numatomi siųsti naudojant modifikuoto veikimo SRTP per 5000 prievadą. Duomenų šifravimui ir autentifikavimui reikalingas identiškas SRTP raktas patalpinas įrenginiuose iš anksto. Judesio jutikliui aptikus veiksmą, kamera pradeda fiksuoti vaizdą, „Raspberry PI“ mikrokompiuteris apdoroja ir transliuoja duomenis klientui.



2.2 pav. Bendroji taikymo srities architektūra

Atvaizduoti RTP ir SRTP protokolų vietas OSI modelyje netikslinga, kadangi kuriamo prototipo atveju, nenaudojami sesijos ir atvaizdavimo lygmenys, nors literatūroje RTP priskiriamas įvairiems, skirtingiems OSI lygmenims: transporto, sesijos, atvaizdavimo, taikomojo. Nepaisant to, jog transporto lygmenyje naudojamas UDP protokolas, RTP protokolui atvaizduoti patogiau naudoti vadinamąjį TCP/IP protokolų atvaizdavimo būdą. Šiuo atveju RTP protokolas įsiterptų tarp transporto ir taikomojo lygmenų, virš UDP protokolo. RTP protokolo pašonėje įsiterpia ir SRTP protokolo dalis. Toks atvaizdavimas visiškai atitinka numatomą prototipo protokolų architektūrą:



2.3 pav. Numatoma architektūra pagal TCP/IP modelį

Tolimesnėje darbo eigoje bus tiriamos RTP, SRTP, SRTP-DTLS protokolų veikimo savybės, kiekybiniai protokolų parametrai. Projektuojamo komunikavimo metodo pagrindinis tikslas yra parinkti tinkamą protokolų architektūrą, kuri turi būti priimtiniausia, siekiant užtikrinti duomenų konfidencialumą, vientisumą, atsižvelgiant į reikalaujamą saugumo lygį ir ribotus įrenginio resursus.

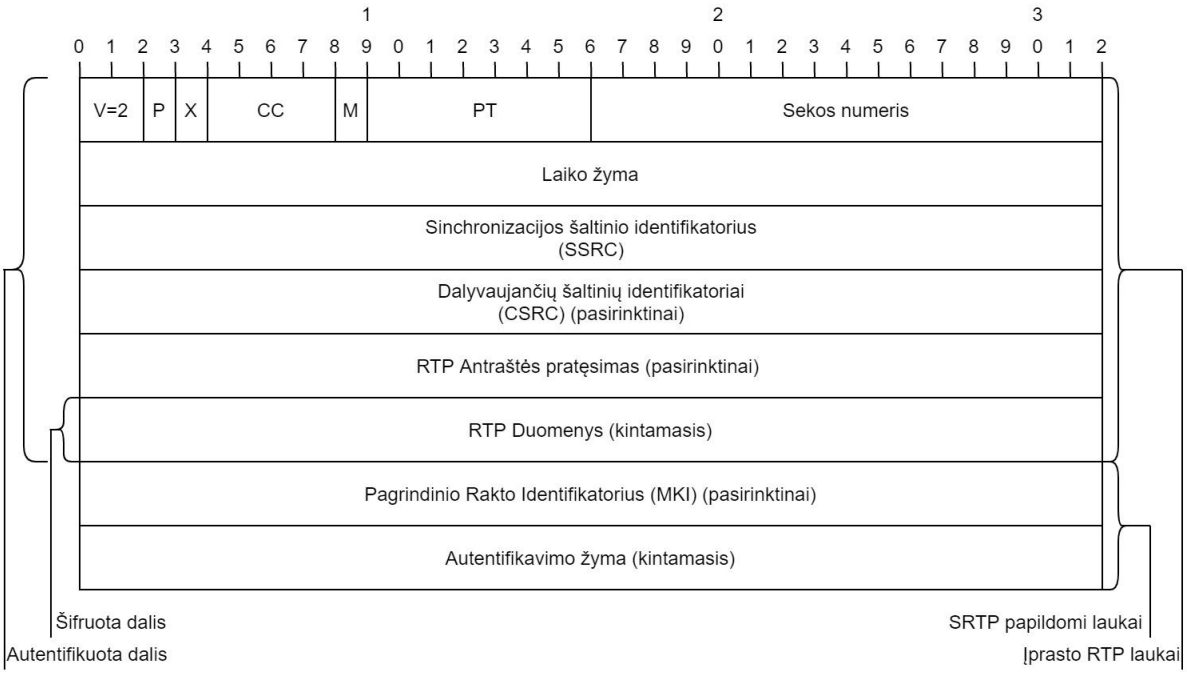
2.3. Pritaikomi protokolai ir jų saugumo mechanizmai

2.3.1. Naudojami protokolai

Tyrime bus naudojami du pagrindiniai protokolai: RTP ir SRTP. Jau išsiaiškinta, jog abu jie skirti medijos duomenim perduoti. Paketo struktūros prasme protokolai nedaug kuo skiriasi. Pats SRTP yra saugesnis RTP profilis su keletu modifikacijų, papildomų algoritmų, kurie leidžia užtikrinti duomenų konfidencialumą, vientisumą ir autentiškumą. Tai apsaugo nuo atkartojimo atakų (angl. *Replay attack*). SRTP paketo struktūra pateikiama 2.4 pav. Iš paketo struktūros matoma, jog RTP nuo SRTP paketo skiriasi tuo, jog prisideda MKI ir autentifikavimo žymos (angl. *Authentication tag*) laukai. Pagrindinio rakto identifikatoriaus (angl. *Master Key Identifier – MKI*) laukas yra pasirinktinis, naudojamas Pagrindinio rakto ID indikacijai, kuris naudojamas SRTP sesijos raktams generuoti. Jis naudingas, kai reikia iš naujo paskirstyti SRTP raktus (pavyzdžiui, pakeisti raktus, kol SRTP sesija vyksta). MKI nukreipia tik į rakto ID, o pats raktas yra išgaunamas kitur, naudojant raktų valdymo infrastruktūrą. Numatomame prototipe pastarojo funkcionalumo atsisakoma, kadangi raktai bus paskirstomi ir patalpinami įrenginiuose iš anksto, be raktų valdymo infrastruktūros, o būtent neprivalomos, papildomos funkcijos ir laukai, pakete lemia mažesnį efektyvumą ir didesnį resursų sunaudojimą. Taip pat, nenaudojamas ir CSRC (angl. *Contributing Source Identifiers*) laukas, kadangi prototipas veiks įrenginys-įrenginiui (angl. *M2M*) principu, o atsižvelgus į tai, jog RTP šaltinis bus tik vienas (robotas), šaltinių žymėti ir skaičiuoti nėra poreikio.

Žymiai svarbesnį vaidmenį projektuojamoje sistemoje atlieka autentifikavimo žyma. Šis laukas, taip pat, nėra būtinas, tačiau jo naudojimas yra stipriai rekomenduojamas SRTP kūrėjų. Apie tai plačiau

kalbama tolimesniame 2.3.3. skyrelyje. Verta paminėti, jog naudojant autentifikavimo žymą, SRTP paketo dydis, lyginant su RTP, padidėja 10 baitų. Tyrimo metu bus išsiaiškinta, ar tai duoda pastebimą poveikį transliacijos parametrams.



2.4 pav. SRTP paketo struktūra

čia: *V* – versija; *P* – užpildas (angl. *padding*); *X* – antraštės pratęsimo žymė; *M* – 1 bito žymeklis, naudojamas įvykiams, tokiems kaip kadro ribos, nustatyti; *PT* – RTP duomenų tipo žymė.

2.3.2. Saugus raktų naudojimas

Būtina dedamoji SRTP duomenų vientisumui, konfidencialumui ir autentifikavimui užtikrinti yra SRTP pagrindinis raktas. Jis susideda iš pagrindinio rakto *„base64“* formatu ir „druskos“ (angl. *Salt*) reikšmės. Rekomenduojamas pagrindinio rakto dydis yra 30 baitų – 16 baitų pats raktas ir 14 baitų „druskos“ reikšmė. Protokolo funkcija (raktų skilimo algoritmas) iš šio rakto sukuria du sesijos raktus, kurie naudojami duomenų šifravimui/iššifravimui ir autentifikavimui. Rakto skilimo proceso algoritmas paremtas AES-CM. SRTP funkcionalumas leidžia nustatyti pagrindinio rakto galiojimo trukmę. Tai reiškia, jog po nustatyto perduotų paketo skaičiaus šis raktas nebegalios. Kaip ir įprastų slaptažodžių keitimas periodiškai, taip ir pagrindinio rakto atnaujinimas yra rekomenduojama papildoma saugumo priemonė. Tam tikrose valdžios institucijose, kur duomenys ypač slapti, šis šifravimo raktų periodinis atnaujinimas yra privalomas [66]. Verta paminėti, jog SRTP funkcionalumas leidžia naudoti ir atsitiktinį pagrindinį raktą kiekvienos sesijos metu, kas reiškia, jog net ir jį pasisavinus, kitos sesijos metu jis nebegalios, nes bus naudojamas kitas sukurtas raktas.

Aplinkos stebėjimo robotas kuriamas ir taikomas naudoti namų aplinkoje. Sistemos koncepcija apibrėžia, jog duomenų apsikeitimas vyksta tarp vaizdo transliuotojo roboto ir specifinio galinio įrenginio, kuriame vaizdas bus stebimas. Įvertinant, jog galinis įrenginys yra vartotojui žinomas ir patikimas prietaisas (asmeninis kompiuteris, nešiojamas kompiuteris ar kt.), diegiant sistemą atsižvelgiama, jog raktų apsikeitimas saugiu kanalu per tinklą, nėra būtinybė, kadangi raktai gali būti sugeneruoti ir įdiegti iš anksto į abiejų šalių įrenginius (kliento ir serverio). Toks sprendinys atsižvelgia, į kuriamos sistemos reikalavimą ir užtikrina, jog bus maksimaliai taupomi autonominio

įrenginio resursai, atsisakant sunkių ir, šiuo atveju, nebūtinų kriptografinių operacijų ar raktų apsikeitimo sesijų.

Taikant iš anksto pasidalinto rakto techniką (angl. *Pre-Shared key*), kaip papildomą saugos priemonę, galima naudoti jo šifravimą ramybės būsenoje (angl. *At-rest encryption*). Autonominiame aplinkos stebėjimo robote šis raktas gali būti talpinamas vaizdo transliavimo programoje arba atskirame faile. Dėl ribotų išteklių programos kodas yra apsaugomas, pavyzdžiui, identifikuojantis vartotojo vardu ir slaptažodžiu. Kliento pusėje šis raktas naudojamas gautų duomenų iššifravimui ir autentifikavimui. Jis talpinamas apsaugotame programiniame kode arba atskirame faile, kuris, taip pat, gali būti papildomai apsaugotas. Kliento pusėje kompiuterio pajėgumai yra pakankami naudoti stiprų šifravimo algoritmą rakto failui šifruoti. Nepaisant to, pagrindinis apsaugos mechanizmas, nuo neautorizuotos prieigos prie rakto, abiejuose įrenginiuose, yra autentifikavimasis slaptažodžiu ar biometriniais autentifikavimo būdais.

2.3.3. Autentifikavimas

SRTP protokolas yra saugesnė RTP versija, kuri užtikrina konfidencialumą, vientisumą ir autentifikavimą medijos duomenų transliacijose. Be duomenų šifravimo, SRTP standartas palaiko pranešimų autentifikavimą ir RTP paketo vientisumą.

Apskaičiuojant viso RTP pranešimo maišos funkciją, įskaitant RTP antraštę ir užšifruotą duomenų dalį, gaunama vertė – pranešimo autentifikavimo kodas (angl. *MAC*). Šis kodas įdedamas į SRTP paketo sudėtį ir paketo struktūroje pažymimas, kaip autentifikavimo žyma. Pasirenkama ir naudojama HMAC-SHA1 maišos funkcija. Rekomenduojama naudoti 32 ilgio maišos funkciją garso transliacijoms, kitoms – 80. Būtent 80 bitų dydžio maišos funkcija yra rekomenduojama ir laikoma kaip saugi riba [67]. Tokio dydžio maišos funkcija sukuria minėtą žymą, kuri įeina į paketo struktūrą.

Procesas, kaip vyksta autentifikavimas, aprašomas toliau. Siuntėjas apskaičiuoja M1 žymą ir prideda ją prie paketo. SRTP gavėjas, naudodamas sutartą algoritmą ir abiems šalims žinomą raktą, apskaičiuoja naują žymą M2 ir atlieka lyginimą su gauta žyma M1. Jeigu abi žymų reikšmės yra lygios (M1 = M2), reiškia, jog duomenys siunčiami tarp autentifikuotų įrenginių, kurie turi tą patį, abiems žinomą SRTP pagrindinį raktą. Priešingu atveju, jeigu žyma M2 ≠ M1, tampa žinoma, jog siuntėjo raktas skiriasi nuo gavėjo rakto, todėl paketai nepriimami ir atmetami, siunčiant audito klaidos pranešimą *„Autentifikavimas nesėkmingas“*. Taip identifikuojamas apsišaukėlis, neautentifikuotas siuntėjas.

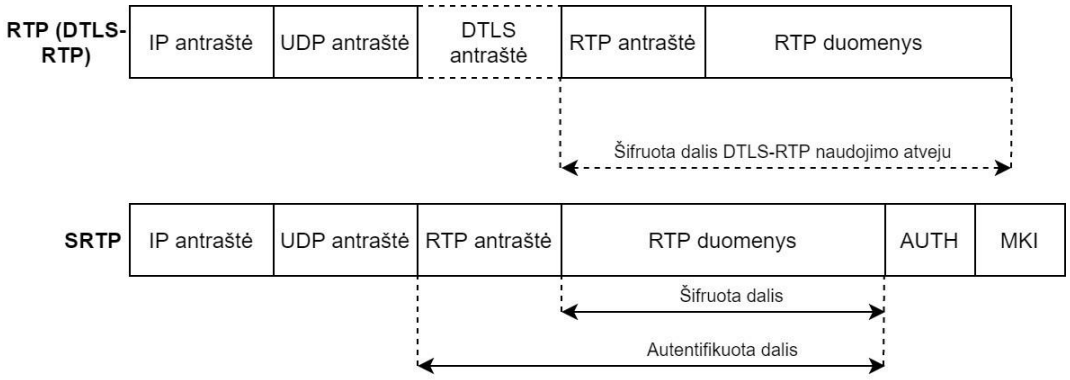
2.3.4. Duomenų šifravimas

Komunikavimo metodo saugumui įgyvendinti pritaikomas SRTP protokolas naudoja AES šifravimo algoritmą su 128 bitų ilgio raktu, kuris, pasak NIST, yra saugus bent jau iki 2030 metų [68]. Šifravimui naudojamas raktas, gautas iš pagrindinio SRTP rakto. Įprastai šifravimo raktai yra signalizuojami SDP deskriptoriuje, panaudojant sesijos inicijavimo protokolus SIP ar SAP. Dažniausiai, tokia perdavimo schema naudojama Pasiūlymas/Atsakymas (angl. *Offer/Answer*) duomenų apsikeitimo algoritme. Minėtas algoritmas naudojamas VoIP pokalbiuose ir kitose konferencijos tipo taikomosiose programose, kuriose duomenys apsikeičiami abiejų krypčių srautais [69]. Siunčiant SDP deskriptorių SIP protokolu, neapsaugotu kanalu, raktas gali būti lengvai išgaunamas. Tokiu atveju, rekomenduojama naudoti SIPS protokolą, kurį sudaro SIP ir TLS. Tačiau, kaip minėta, projektuojamos sistemos atveju, dėl stipriai ribotų resursų, stengiamasi išvengti daug

skaičiavimo išteklių ir energijos naudojančių sunkių kriptografinių operacijų, tokių kaip TLS sesijos inicijavimas. Verta paminėti, jog SIP protokolo naudojimas IoT įrenginių terpėje daugelyje scenarijų sukelia tik papildomą resursų naudojimą ir yra dažniausiai nereikalingas [32].

Dažniausiai, SDP deskriptoriuje aprašomas SRTP pagrindinis raktas, naudojami šifravimo ir autentifikavimo algoritmai, transliacijos pobūdis, parametrai. Projektuojamoje sistemoje duomenų perdavimas nėra abipusis (konferencijos tipo) ir yra atliekamas tarp žinomų ir patikimų įrenginių, todėl ši informacija patalpinama iš anksto, netransportuojant jos ryšiu. Siekiama, jog transliavimo sistema būtų kiek įmanoma kompaktiškesnė, todėl visa reikalinga informacija, kuri įprastai pateikiama SDP deskriptoriuje, talpinama programiniame kode-scenarijuje (išskyrus SRTP pagrindinį raktą). Ši informacija reikalinga transliacijos parametrų perdavimui, sėkmingam autentifikavimui ir duomenų šifravimui/iššifravimui. Minėti duomenys turi sutapti siuntėjo ir gavėjo pusėse, priešingu atveju, transliacija ir vaizdo priėmimas gali būti nesėkmingi.

Būtina pabrėžti, jog DTLS-RTP protokolo poros veikimas iš esmės skiriasi nuo DTLS-SRTP veikimo. Raktų apsikeitimo mechanizmas veikia taip pat abiejose protokolų porose, tačiau šifravimo algoritmai skirtingai. DTLS-RTP protokolų poroje naudojamas standartinis DTLS duomenų srauto šifravimas, kas reiškia, jog RTP paketai yra pilnai užšifruojami, nepaliekant antraštės atviru tekstu. SRTP-DTLS ar SRTP naudojimo atvejais šifruojama tik duomenų dalis (angl. *Payload*), paliekant RTP antraštę atvirą.



2.5 pav. Funkcinė paketų struktūros schema

Tokiu būdu taupomi skaičiavimo pajėgumai atsisakant nereikalingo pilno paketo šifravimo, kas stipriai apkrauna įrenginio procesorių. Vertinant tinklo parametrus, įgaunamas pranašumas atsižvelgus į tinklo pralaidumą, kadangi siunčiamas, dalinai šifruotas paketas yra mažesnio dydžio. Kitas pranašumas nešifruojant pilno paketo ir išsaugant antraštes yra, jog neprarandama informacija, kuri gali pasitarnauti šalinant paketų praradimo, vėlavimo ir transliavimo problemas. Verta paminėti, jog paliekant RTP antraštę nešifruotą atsiranda potenciali saugos spraga. RTP paketo antraštėje gali būti nusikaltėliui naudinga informacija, iš kurios nusikaltėlis gali nustatyti, pavyzdžiui, kokiuose tinkluose yra apsikeičiami medijos duomenys tarp dviejų dalyvių. Projektuojamo roboto perduodamo paketo antraštėje esanti informacija, nėra kritinės svarbos, o kadangi saugos komunikavimo metodo kūrimo vienas iš iškeltų tikslų yra kaip įmanoma labiau taupyti autonominio roboto ribotus resursus, šis šifravimo metodas puikiai tinka.

Lyginant SRTP-DTLS ir RTP-DTLS veikimą, buvo atliktas tyrimas [70], kurio rezultatų išvados parodė, jog RTP-DTLS yra labiau pritaikomas skirtingo pobūdžio programose ir platformose, diegimas yra lengvesnis ir ši architektūra yra labiau tinkama aukšto efektyvumo tinkluose. Visgi,

SRTP-DTLS siūlo aukštesnį saugumo lygį ir efektyviau naudoja įrenginio skaičiuojamąją galią, kadangi šifravimui naudoja SRTP, pilnai nešifruojant viso paketo.

2.3.5. Saugos mechanizmų modifikavimas ir naudojimas

Kaip aprašyta 2.3.1. skyrelyje, SRTP protokolas, veikiantis individualiai ar kartu su raktų valdymo pagalbiniu protokolu, šifravimui ir autentifikavimui naudojamą pagrindinį raktą (angl. *Master key*), įprastinėje protokolo konfigūracijoje generuoja atsitiktiniu būdu, suteikiant galiojimo laiką, kuris apibrėžiamas fiksuotu siunčiamų paketų skaičiumi. Standartinis paketų skaičius, kol raktas galioja yra iki 2^{48}. Tokiu principu sumažinima potenciali rakto pasisavinimo rizika. Šis sugeneruotas raktas transportuojamas tarp serverio ir kliento, naudojant išorinius raktų apsikeitimo protokolus. Saugaus vaizdo transliavimo metode, tarp autonominio roboto ir galinio įrenginio, numatoma naudoti tą patį, iš anksto, o ne atsitiktiniu būdu sugeneruotą pagrindinį SRTP raktą. Raktas numatytas be galiojimo laiko ir žinomas abiems įrenginiams, prieš pradedant vaizdo transliaciją. Naudojant tokį algoritmą pavyks išvengti raktų apsikeitimo sesijos inicijavimo proceso, kuris vykdant vaizdo duomenų transliaciją gana trumpą laiką (15 sekundžių), sunaudotų didelę dalį įrenginio resursų, vertinant resursus, sunaudotus bendros operacijos metu.

Šis raktas pagal protokolo funkcionalumą, skyla į du raktus, kurie naudojami siuntėjo/gavėjo autentifikavimui bei duomenų šifravimui, todėl užtikrinamas duomenų vientisumas ir konfidencialumas. Net ir perėmus šiuos vaizdo duomenis jie būtų beverčiai, kurių neįmanoma atkurti pašaliniam asmeniui – nusikaltėliui.

Lyginant paketo struktūrą, atsisakoma keletos papildomų, nereikalingų laukų. Šie pakeitimai padės „išlengvinti“ SRTP protokolą ir pritaikyti jį autonominio aplinkos stebėjimo roboto saugos komunikavimo metode. Paketo struktūros pokyčiai aprašyti 2.3.1. skyrelyje, kur nagrinėjami naudojami protokolai ir bendra SRTP paketo struktūra.

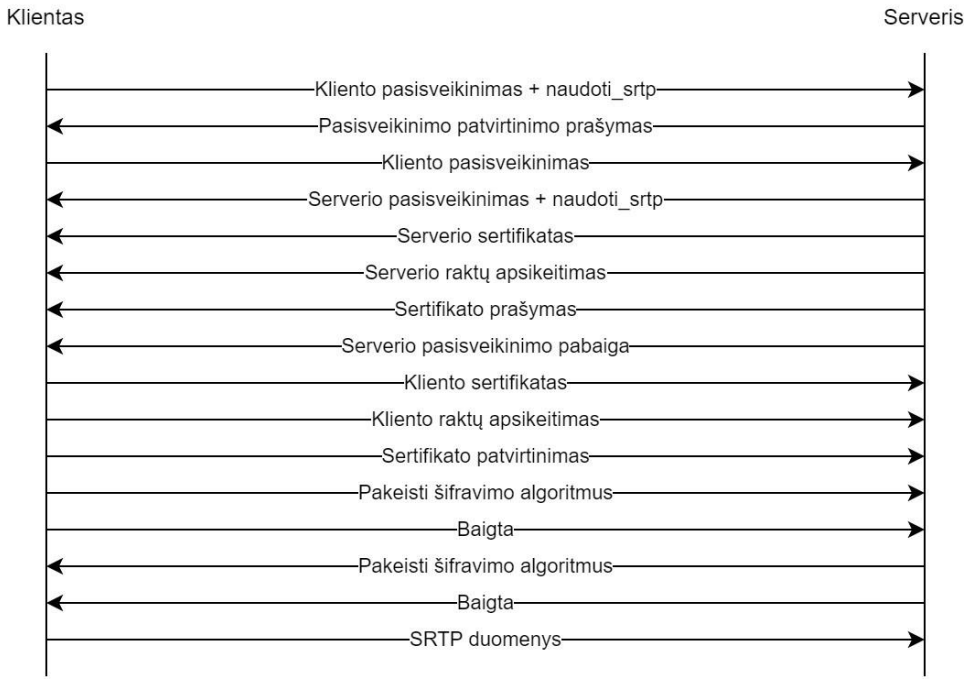
SRTP-DTLS protokolų panaudojimo atveju, įprastai SRTP raktas apsikeičiamas saugiu kanalu, kuris inicijuojamas užmezgus DTLS sesiją ir įgyvendinus „rankos paspaudimo“ algoritmą. Sėkmingam „rankos paspaudimo“ mechanizmo įgyvendinimui apskaičiuojami vieši sertifikatai, raktai, privatūs raktai ir apsikeičiami tarp kliento ir serverio, siekiant patvirtinti siuntėjo ir gavėjo tapatybes (autentifikavimas). Sėkmingai užmezgus sesiją, pagal apsikeistą informaciją, generuojamas ir pasidalijamas atsitiktinis SRTP raktas, kuris naudojamas tolimesnėje komunikacijoje, šifruojant duomenis. Toks ilgas procesas, sudarytas iš daug operacijų, trumpoje duomenų siuntimo sesijoje yra labai nuostolingas pajėgumų prasme. Siunčiant duomenis SRTP protokolu ir naudojant DTLS, kaip raktų apsikeitimo mechanizmą, operacijos atsakomybės, tarp minėtų protokolų, pasiskirsto taip:

- Duomenys šifruojami naudojant SRTP;
- DTLS naudojamas SRTP reikalingų raktų, parametrų apsikeitimui;
- DTLS naudojamas SRTP algoritmų susitarimui (angl. *Negotiation*) ir apsikeitimui.

Norint sutarti, jog būtų naudojamas SRTP kartu su DTLS, klientas pasisveikinimo pranešime (angl. *ClientHello*) turi įtraukti papildymą *„naudoti_srtp“* (angl. *use_srtp*). Šiame papildyme nurodoma, jog naudojamas SRTP profilis, tai yra, šifravimo algoritmas, rakto dydis ir autentifikavimo maišos funkcijos tipas bei dydis. Serveris, gavęs pasisveikinimą su šiuo papildymu, gali sutarti naudoti SRTP, taip pat, įtraukiant *„naudoti_srtp“* į savo pasisveikinimo pranešimą. Viską sutarus ir įgyvendinus visus DTLS sesijos užmezgimo standartinius žingsnius, duomenys apsikeičiami

naudojant SRTP-DTLS protokolų porą. Šios sesijos metu apsikeičiami apsaugoti duomenys transportuojant juos per vieną UDP protokolo siuntėjo ir gavėjo prievadų porą.

Naudojant SRTP-DTLS protokolus kartu, DTLS architektūroje, priešingai nei bendroje sistemos koncepcijoje, kliento vaidmuo atitenka autonominiam aplinkos stebėjimo robotui, kadangi užfiksavus judėjimą stebėjimo zonoje, pradedama vaizdo transliacija, to pasekoje inicijuojama DTLS pasisveikinimo sesija. Pranešimų tipas ir pranešimų apsikeitimo eiliškumas tarp serverio ir kliento SRTP-DTLS sesijos užmezgimo metu pavaizduotas 2.6 paveiksle.



2.6 pav. SRTP-DTLS sesijos užmezgimas

2.4. Rizikos transliuojant vaizdą ir apsaugos mechanizmai

Informacija, naudingojoje paketo dalyje (vaizdo duomenys), yra konfidenciali ir turi būti skirta tik nurodytam gavėjui. Antraštės ir naudingojo krovinio dalies vientisumas yra būtinas prieš pasyvų šnipinėjimą ir aktyvius apsimetinėjimo išpuolius. SRTP pranešimų autentifikavimas ir šifravimas apsaugo nuo šių atakų. Jeigu yra reikalingas ir paketo antraštės konfidencialumas, galima naudoti papildomus apsaugos mechanizmus, tačiau, kaip minėta 2.3.4. skyrelyje, kuriamo komunikavimo metodo atveju, kritinės svarbos informacijos antraštėje nėra.

RTP paketų šifravimui, iššifravimui ir vientisumo užtikrinimui, naudojamas SRTP pagrindinis raktas turi būti apsaugotas nuo išorinio pasisavinimo. Norint apsisaugoti nuo pasyvių ir aktyvių atakų, reikalingos tinklo ir galinių sistemos prieigų kontrolės (angl. *End system access control*). Kuriamo komunikavimo metodo atveju, raktai nėra siunčiami tinklu, todėl raktų apsaugą užtikrina galinės sistemos prieigos kontrolė. Numatoma naudoti prisijungimą vartotojo vardu ir slaptažodžiu arba biometriniais duomenimis, užtikrinant, jog neautorizuotas asmuo negalės prieiti prie atskiro rakto failo ar programinio kodo su raktu. Kaip papildomas saugumo mechanizmas gali būti programinio kodo ar atskiro rakto failo pasyvus šifravimas.

Signalizuojantys pranešimai, kuriuose nurodomi sesijos parametrai ar net siunčiami raktai, turi būti apsaugoti. Vienas iš tyrimo objektų yra raktų pasidalinimas tinklu. Norint, jog tai įvyktų saugiai,

išvengiant raktų šniukštinėjimo ir galimo jų pasisavinimo, būtina naudoti papildomus saugumo mechanizmus. Pavyzdžiai gali būti DTLS, IPSEC, S/ MIME ar kiti duomenų saugumo protokolai, skirti apsaugoti signalizavimo pranešimus. Kuriamo komunikavimo metodo prototipo tyrimuose, raktų pasidalijimui ir transliacijos parametrų siuntimui tinklu, bus naudojamas DTLS protokolas.

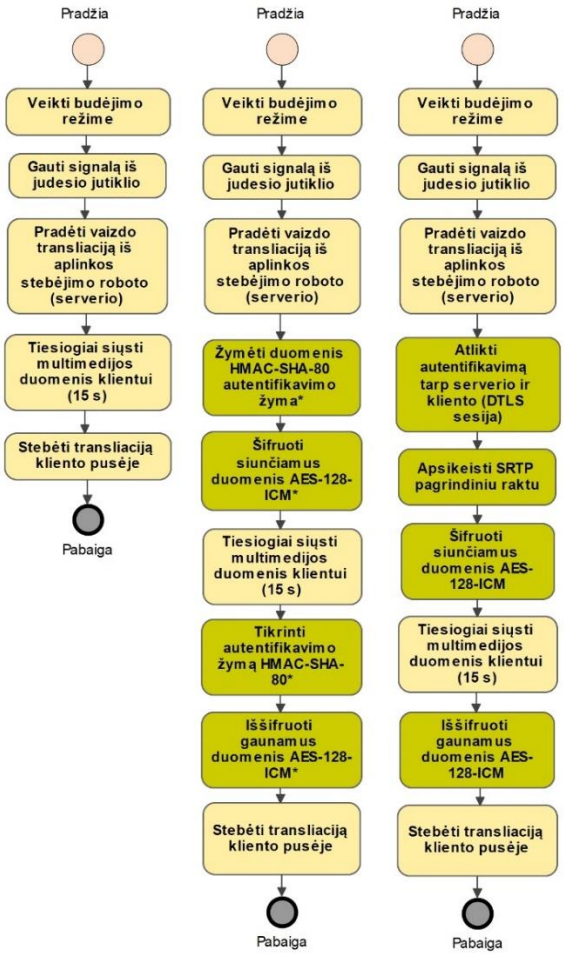
Tik autorizuoti RTP sesijos dalyviai turėtų siųsti paketus RTP sesijos adresu (tinklo adresas ir prievadas). Norint apsisaugoti nuo paskirstytos atsisakymo aptarnauti atakos (angl. *DDoS*) ir kitų išpuolių, neleistini, neautorizuoti paketai turi būti atmetami sunaudojant kuo mažiau skaičiuojamųjų resursų ir atminties. Apsauga nuo atkartojimo atakų ir SRTP paketo vientisumo užtikrinimas, gali apsaugoti nuo minėtos atakos, atmetant neautentifikuotus paketus.

2.1 lentelė. Rizikų ir grėsmių sumažinimo apsaugos funkcijos

Rizika ir grėsmė	**Saugos funkcijos**
Duomenų konfidencialumas ir vientisumas	SRTP naudingosios paketo dalies šifravimas, pranešimo autentifikavimas maišos funkcija
Raktų pasisavinimas	Prieigos prie įrenginio kontrolė, autentifikavimasis vartotojo vardu ir slaptažodžiu, biometriniais duomenimis
Tinklu signalizuojamų raktų pasisavinimas	Papildomo saugos mechanizmo naudojimas, pavyzdžiui, DTLS
Atsisakymo aptarnauti ataka	SRTP pranešimo autentifikavimas, naudojant maišos funkciją

2.5. Roboto atliekamos operacijos scenarijus

Skyrelyje 2.1.1. aptartas numatomo autonominio aplinkos stebėjimo roboto panaudojimo scenarijus, įrenginio aplinka, veikimo principas. Remiantis numatoma koncepcija, 2.7 paveiksle pateikiamos funkcinės *„as-is“* ir *„to-be“* diagramos, kuriose supaprastintai atvaizduota, kokie papildomi žingsniai atsiranda transliuojant multimedijos duomenis saugiai. Pateikti neskaidyti žingsniai, atliekant roboto operaciją, tiesioginį vaizdo transliavimą, naudojant protokolus RTP, SRTP ir SRTP-DTLS.



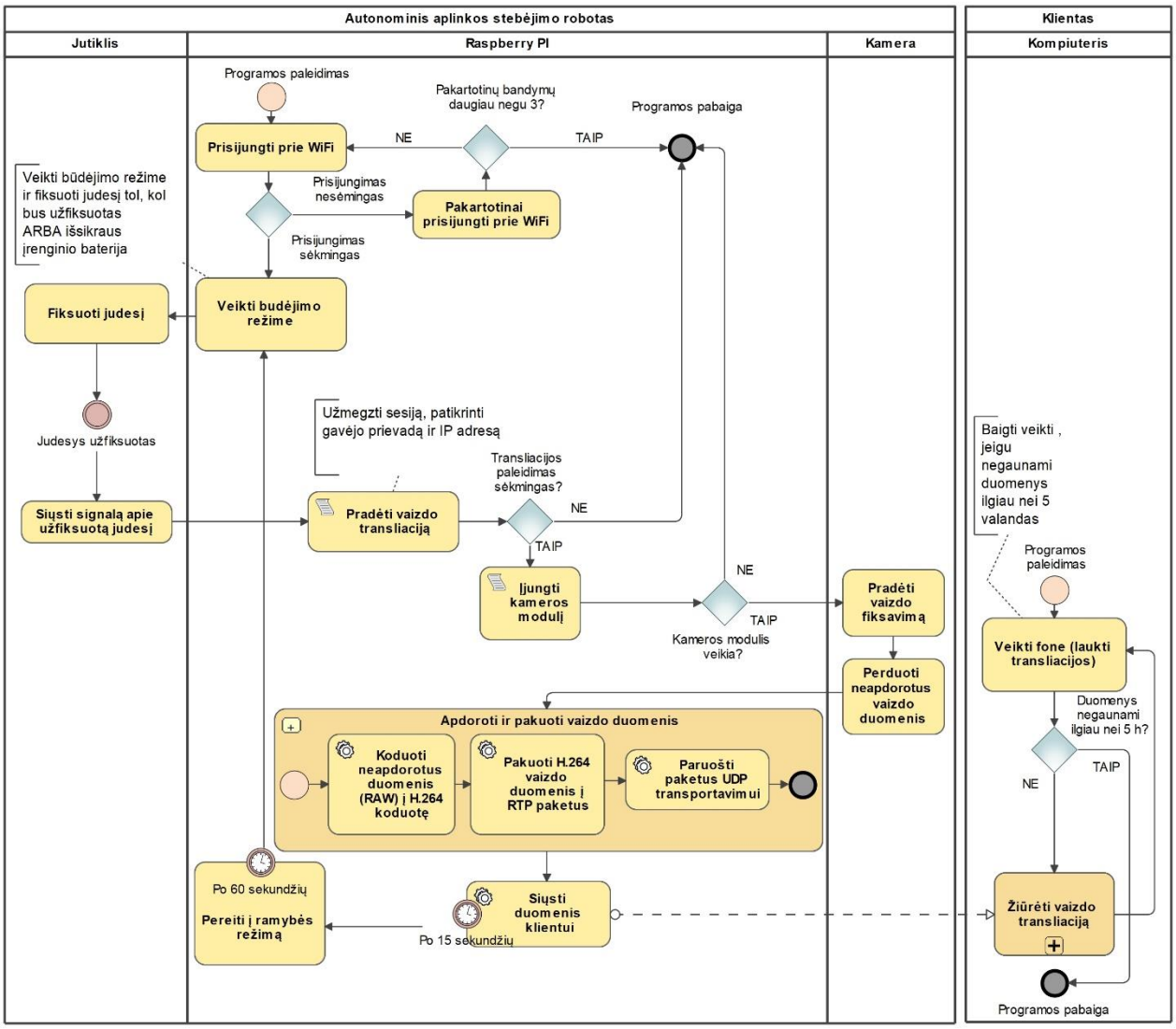
2.7 pav. Funkcinės RTP, SRTP ir SRTP-DTLS veiklos proceso schemos

čia: * - Naudojamas iš anksto pasidalintas SRTP pagrindinis raktas.

Vaizdo transliacijos įgyvendinimas naudojant RTP ir reikiamo saugumo užtikrinimas, naudojant modifikuotą SRTP su iš anksto pasidalinto rakto technika, pavaizduoti 2.8 ir 2.10 paveiksluose pateikiamose BPMN veiklos procesų diagramose. Detaliai pavaizduojami komunikavimo žingsniai ir iš serverio klientui keliaujantys duomenų srautai.

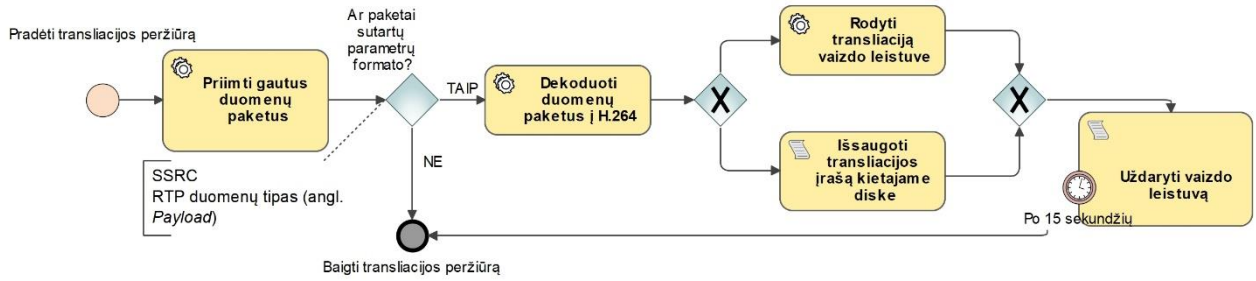
Roboto veikimas yra autonomiškas, tai reiškia, jog jis turi veikti, kaip įmanoma su mažesniu, arba be žmogaus įsikišimo. Įjungiant programą robote, nurodomas gavėjo IP adresas ir prievadas. Operacija, naudojant RTP protokolą multimedijos duomenų transliavimui, prasideda nuo prisijungimo prie WiFi (žr. 2.8 pav.). Komunikavimas neįmanomas, jeigu nėra ryšio. AASR veikimas numatytas konkrečiame tinkle (namų ūkio tinklas). Po trijų nesėkmingų bandymų prisijungti prie tinklo, programa stabdoma. Sėkmingai prisijungus, robotas veikia budėjimo režime ir laukia judesio fiksavimo jutiklio signalo. Kai judesys užfiksuojamas, judesio signalo būsenos pasikeitimas reiškia startavimo signalą vaizdo transliacijos pradžiai. Pirmiausia, patikrinama ar transliacijos paleidimas sėkmingas (tikrinami gavėjo prievadas ir IP adresas). Jeigu kliento įrenginys, su nurodytais parametrais yra prieinamas, įjungiamas kameros modulis ir pradedamas vaizdo fiksavimas. Neapdoroti duomenys (angl. *Raw*) perduodami į vaizdo transliavimo programos karkasą, kur atliekami kodavimai į reikiamą koduotę, pakavimas į RTP paketus ir paruošimas transportavimui UDP protokolu. Tai pavaizduota išskleistame sub-procese „*Apdoroti ir pakuoti vaizdo duomenis*“. Kai duomenys paruošiami, jie siunčiami klientui, o po 15 sekundžių transliavimo, programa robote

pereina į ramybės režimą (angl. *Cool-down time*). Po 60 sekundžių šiame režime, robotas vėl veikia budėjimo režime ir laukia galimo judesio užfiksavimo. Budėjimo režime robotas veikia tol, kol judesys bus užfiksuotas arba išsikraus įrenginio baterija.



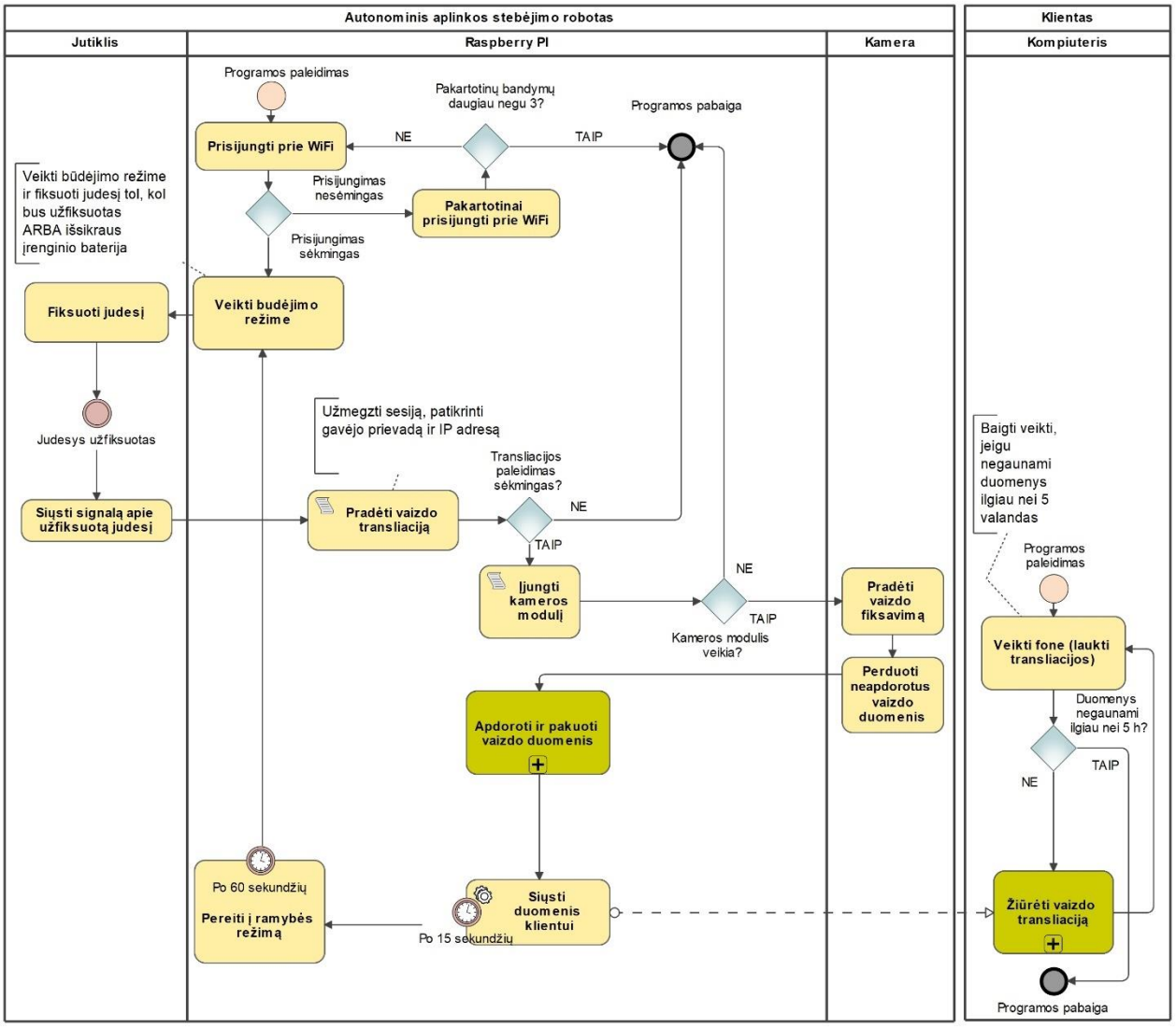
2.8 pav. Operacijos, naudojant RTP, veiklos proceso modelis

Tuo tarpu kliento kompiuteryje, programa paleista ir veikia fone. Pradėjus duomenims keliauti nurodytu prievadu, vaizdo duomenys priimami seka, kuri pavaizduota išplestiniame sub-procese „*Žiūrėti vaizdo transliaciją*“ (žr. 2.9 pav.). Čia priimami gauti duomenų paketai, patikrinami ar yra sutarto formato. Tikrinamas SSRC ir RTP duomenų tipas. Jeigu parametrai nesutampa su parametrais, sutartais iš anksto su robotu, transliacija nesuderinta ir vaizdo transliacijos priėmimas baigiamas. Toliau duomenų paketai dekoduojami į H.264 vaizdo kodeko formatą. Šis formatas yra tinkamas atlikti sekančius žingsnius, tai yra, lygiagrečiai „*Rodyti transliaciją vaizdo leistuve*“ ir „*Išsaugoti transliacijos įrašą kietajame diske*“. Sėkmingai įvykus vienai iš šių užduočių, po 15 sekundžių (transliacijos laikas) uždaromas vaizdo leistuvas. Transliacijos peržiūrai pasibaigus (žr. 2.8 pav.), vėl pereinama į „*Veikti fone*“ režimą, kur toliau laukiama galima vaizdo transliacija iš roboto. Jeigu transliacija neįvyksta ilgiau nei 5 valandas, programa baigiama.



2.9 pav. Sub-procesas „Žiūrėti vaizdo transliaciją“ (RTP)

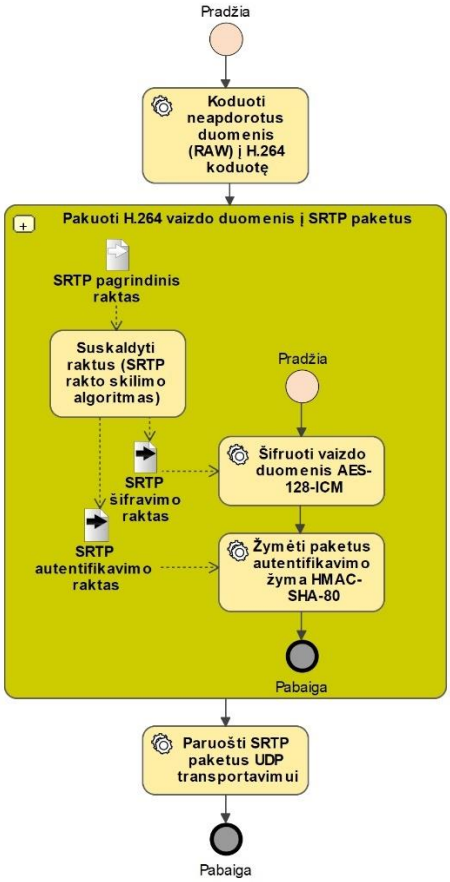
Modifikuoto SRTP naudojimo atveju, su iš anksto pasidalinto rakto technika (žr. 2.10 pav.), pagrindinis sekos algoritmas išlieka tas pats, tačiau prisideda keletas papildomų žingsnių, kurie užtikrina duomenų saugumą.



2.10 pav. Operacijos, naudojant pritaikomą SRTP, veiklos proceso modelis

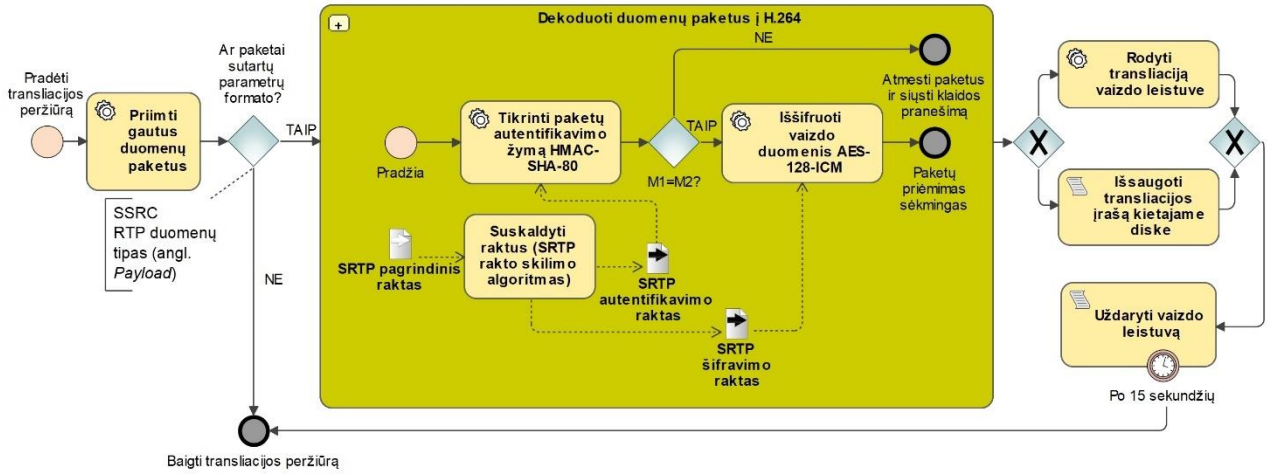
Sub-procesas „*Apdoroti ir pakuoti vaizdo duomenis*“ pateikiamas kaip išplėstinis sub-procesas (žr. 2.11 pav.). Duomenų šifravimui ir autentifikavimui naudojamas iš anksto pasidalintas SRTP

pagrindinis raktas, be galiojimo laiko, kuris yra vienodas ir patalpintas abiejuose įrenginiuose. Šis raktas skyla į šifravimo ir autentifikavimo raktus, kurie naudojami šioms funkcijoms atlikti.



2.11 pav. Sub-procesas „Apdoroti ir pakuoti vaizdo duomenis“

Kliento įrenginyje, keičiasi sub-proceso *„Žiūrėti vaizdo transliaciją“* sandara (žr. 2.12 pav.). Naudojant metodui pritaikomą SRTP, paketų dekodavimas į vaizdo formatą atliekamas naudojant tą patį, iš anksto pasidalintą SRTP pagrindinį raktą, iš kurio, taip pat, skyla atskiri šifravimo ir autentifikavimo raktai. Šiuo atveju autentifikavimo raktas naudojamas HMAC-SHA-80 maišos funkcijos kontrolinei sumai apskaičiuoti ir patikrinti ar prie paketo esanti autentifikavimo žyma sutampa (M1=M2). Jeigu šios žymos nesutampa, paketai atmetami ir siunčiamas klaidos pranešimas, kadangi tai reiškia, jog paketai gauti ne iš rakto bendrasavininko, o apsišaukėlio.



2.12 pav. Sub-procesas „Žiūrėti vaizdo transliaciją“ (modifikuotas SRTP)

2.6. Metodo išvados

1. Projektuojamas komunikavimo metodas, tarp AASR ir kliento galinio įrenginio, turi garantuoti saugų vaizdo duomenų perdavimą, užtikrinant jų vientisumą ir konfidencialumą, pasižymėti priimtinu vaizdo transliacijos vėlavimu ir optimaliai naudoti įrenginio skaičiavimo pajėgumus, atsižvelgiant į ribotus roboto energijos išteklius.
2. Įvertinti vaizdo transliacijai tinkamų protokolų ir jų architektūrų saugumo mechanizmai, pritaikomos modifikacijos. SRTP ir SRTP-DTLS užtikrina duomenų konfidencialumą, vientisumą, tačiau, autentifikavimas yra saugesnis, naudojant SRTP poroje su DTLS.
3. Sistemoje numatytoje riboto laiko vaizdo transliacijoje siekiama išvengti sudėtingų kriptografinių operacijų, tokių kaip DTLS sesijos užmezgimas, kadangi trumpos transliacijos atveju, ši dalis sunaudoja didelę dalį įrenginio skaičiavimo resursų, kas tiesiogiai susiję su energijos naudojimu, taip pat, iššaukia transliacijos vėlavimą.
4. Autonominio aplinkos stebėjimo roboto atliekamos operacijos žingsniai atvaizduoti BPMN veiklos procesų diagramose. Įdiegus reikiamą saugą, operacijos pagrindiniai algoritmo žingsniai išlieka tokie patys, tačiau keletas procesų yra išplėčiami, papildomi užduotimis, kurios atlieka šifravimo ir autentifikavimo funkcijas.

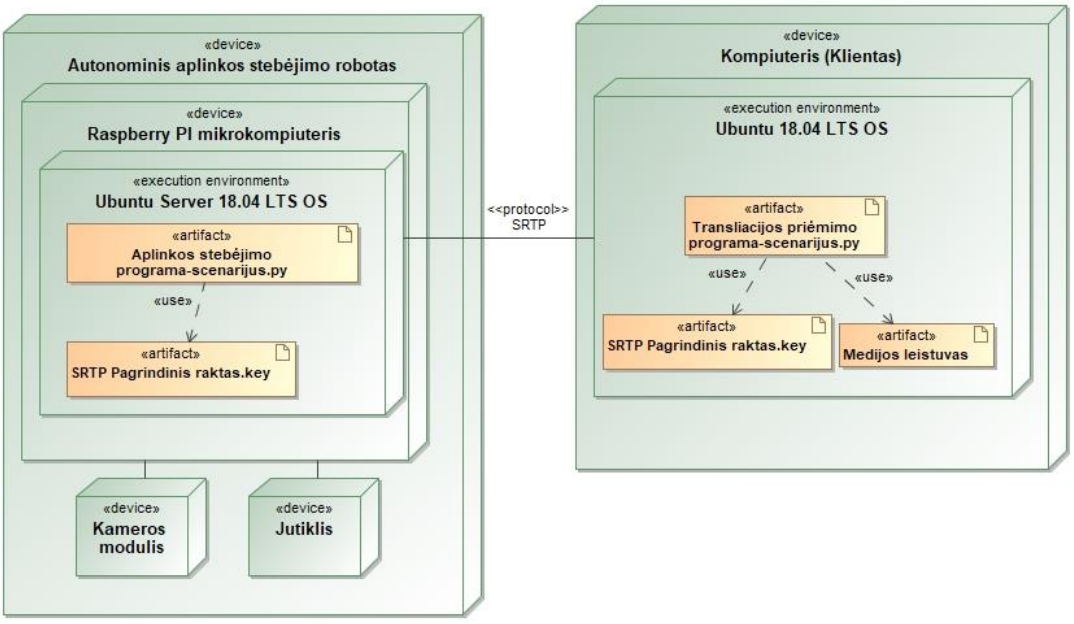
3. Autonominio aplinkos stebėjimo roboto komunikavimo metodo prototipas

Komunikavimo metodo prototipas bus realizuotas „Raspberry PI 3 B+“ įrenginyje, kuriame veiks „Raspbian“ OS. Šis mikrokompiuteris buvo pasirinktas dėl tyrimui palankių savybių: universalumo, suderinamumo, geros dokumentacijos, elementų gausos ir patogaus konfigūravimo. Visa tai leis tyrimus atlikti efektyviai, lanksčiai ir patogiai. Nepriklausomai nuo gana gerų įrenginio parametrų, tyrimas bus orientuotas į ribotų energijos išteklių, skaičiavimo pajėgumų, tinklo parametrų įrenginių grupę.

Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo prototipo tyrimams, bus naudojami SRTP, SRTP-DTLS protokolai, taip pat, nesaugus RTP protokolas, kurio atžvilgiu bus lyginama, kiek „kainuoja“ saugumas vaizdo duomenų transliavime. Vaizdo transliacija įgyvendinama metode, pritaikant minėtus protokolus ir jų konfigūracijas. Eksperimentinėje dalyje, fiksuojant kiekybinius parametrus, atsižvelgus į transliacijos kriterijus, bus apibendrinami ir palyginami gauti rezultatai, konstatuojamos išvados.

3.1. Sistemos diegimo diagrama

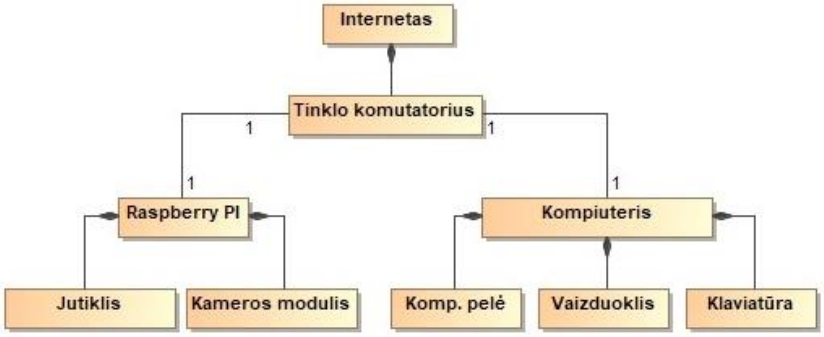
Komunikavimo metodo sistemos prototipo sudėtis pavaizduota 3.1 paveiksle. Atvaizduotas SRTP protokolo architekūros transliavimo metodas, kuris bus pagrindinis darbo tyrimo objektas. Vaizdo transliavimas vyks iš autonominio aplinkos stebėjimo roboto į kliento namų kompiuterį. Kaip minėta, metode taikoma iš anksto pasidalinto rakto technika. Raktas talpinamas programiniame kode-scenarijuje arba atskirame faile. Šis, iš anksto sugeneruotas raktas, diegiant sistemą patalpinamas abiejų įrenginių atmintyje (robote ir kliento kompiuteryje). Dėl patogesnio eksploatavimo, įrenginiuose naudojamos „Ubuntu“ operacinės sistemos, kuriose galima naudoti atvirojo kodo programinę įrangą (angl. *Open-Source Software*) ir įvairias bibliotekas, karkasus. Tai leis lengviau įgyvendinti norimas modifikacijas protokoluose ar programose, pritaikyti jau egzistuojančius, išdirbtus, esamus produktus ir įrankius.



3.1 pav. Sistemos diegimo diagrama (modifikuotas SRTP)

3.2. Pagrindiniai autonominio aplinkos stebėjimo roboto techninės įrangos elementai

Fizinė autonominio aplinkos stebėjimo roboto koncepcija, apibrėžta kaip įrenginių visuma, kurią sudaro mikrokompiuteris, kameros modulis, PIR judesio jutiklis ir jų sąveikai reikalingi montažiniai, jungiamieji elementai. Kliento, šiuo atveju vaizdo stebėtojo (gavėjo), pusėje bus naudojamas nešiojamas kompiuteris Intel Core i7-4700HQ CPU @ 2,40 GHz, 8GB RAM, kuris priims vaizdą iš roboto. Kokių parametrų įrenginys priims vaizdą, tyrimo atžvilgiu, nėra svarbu, kadangi tyrimas orientuotas į vaizdo trasnliuotojo (roboto) kiekybinių parametrų tyrimą, atliekant saugią vaizdo transliaciją. Patogesnei vartotojo sąsajai naudojami papildomi elementai – pelė, klaviatūra, vaizduoklis. Duomenų apsikeitimui WiFi bevieliu ryšiu būtinas tinklo komutatorius ir internetas. Pateikiama bendra techninės įrangos UML klasių diagrama 3.2 paveiksle.



3.2 pav. Techninės įrangos struktūra UML klasių diagramoje

3.2.1. „Raspberry PI 3B+“ mikrokompiuteris

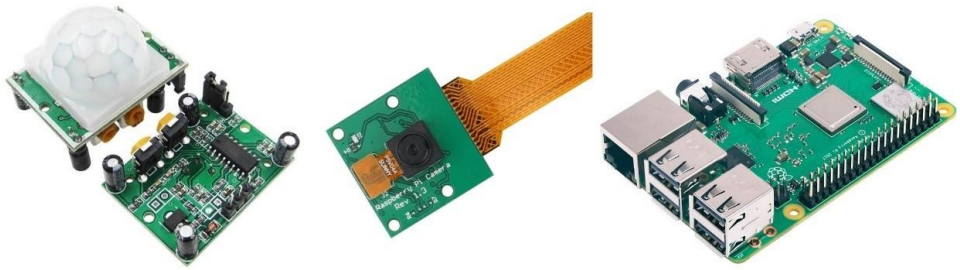
Pagrindinis darbo tyrimų įrankis „Raspberry PI 3B+“ kredito kortelės dydžio mikrokompiuteris, plačiai naudojamas įgyvendinant įvairaus pobūdžio IoT projektus, tiek namų aplinkoje, tiek pramonėje pramonėje (žr. 3.3 pav.). Kompiuterio parametrai yra pakankami apdoroti ir siųsti vaizdo duomenis realiu laiku dedikuotam galiniui įrenginiui. Įdiegta „Ubuntu Server“ operacinė sistema paremta „Unix“ OS pagrindu, todėl įrenginys gali naudoti didžiąją dalį standartinių vaizdo transliavimo bibliotekų, karkasų, jeigu įrenginio parametrai leidžia jas palaikyti. Tyrimas nukreiptas į ribotus skaičiavimo, energijos ir tinklo išteklius turinčius įrenginius, todėl prototipas ir eksperimentai orientuoti į protokolų modifikavimą, konfigūracijų pakeitimus, kas leistų standartinius vaizdo transliavimo ir saugos protokolus naudoti kuo efektyviau, suvartojant kaip galima mažiau energijos.

3.2.2. Kameros modulis

Prototipe pasirinktas mažų matmenų, kompaktiškas kameros modulis „RPI Cam OV5647“, kuris pasižymi visomis transliacijai reikalingomis savybėmis: H.264 kodavimu, aukštos rezoliucijos palaikymu (2592 x 1944), 5 mega pikselių lęšiu, 1080P raiškos ir 30 kadrų per sekundę (angl. *FPS*) filmavimo galimybe (žr. 3.3 pav.). Kameros modulio ir mikrokompiuterio komunikavimas, įgyvendinamas sujungus elementus MIPI CSI jungtimi. Palaikomi parametrai yra pakankami, norint atlikti kokybiško vaizdo ištransliavimą, pagal antrame skyriuje nustatytus kriterijus.

3.2.3. PIR judesio daviklis

Pagrindinės sistemos operacijos inicijavimui, numatytas „HC-SR501“ PIR judesio daviklis, kuriam užfiksavus judesį, pradedama vaizdo transliacija į galinį įrenginį (žr. 3.3 pav.). Jutiklis su mikrokompiuteriu sujungiamas per GPIO kontaktus, maitinamas 5 VDC įtampa. Jutiklyje gali būti konfigūruojama signalo siuntimo vėlavimo trukmė, trigerio būsenos, judesio fiksavimo jautrumas.



3.3 pav. Autonominio aplinkos stebėjimo roboto komponentai [71], [72], [73]

3.3. Įrenginio maitinimas ir suvartojama energija

Mikrokompiuteriui reikalingas 5VDC įtampos ir 2,5A srovės maitinimas. Toks maitinimas gali būti užtikrinamas maitinant per micro USB jungtį arba per 5V GPIO kontaktus. Elektros energija įrenginiui gali būti teikiama iš galvaninių elementų, elektros bankų (angl. *Power banks*), elektros lizdo, naudojant maitinimo bloką (keitiklį), ar tiesiog maitinantis nuo galingesnio įrenginio. Tikslumui išgauti, tyrimo eigoje atliekami bandymai ir matavimai bus atliekami tam tikrą skaičių iteracijų, todėl tyrime numatytas maitinimas iš galingesnio įrenginio ar elektros tinklo, per USB magistralę. Roboto prototipo koncepcijoje numatytas veikimas iš išorinės baterijos, tačiau siekiant greičiau atlikti bandymus ir norint išvengti pakartotino baterijos įkrovimo, po daugybės bandymų, įrenginys bus maitinamas patogesniu būdu.

Vertėtų nepamiršti pagrindinio, kuriamo saugos komunikavimo metodo prototipo, vertinimo kriterijaus – energijos suvartojimo. Saugus vaizdo transliavimas turi būti įgyvendinamas su kuo mažesnėmis energijos sąnaudomis, kadangi prototipas yra orientuotas į ribotų išteklių, autonominius įrenginius. Dėl neišvengiamo balansavimo tarp transliacijos saugumo funkcijų ir energijos suvartojimo, svarbu pasirinkti efektyvius saugumo mechanizmus ir protokolus, kurie užtikrintų pakankamą, tačiau neperteklinį duomenų saugumo lygį, atsižvelgiant į roboto savybes, atliekamą užduotį, duomenų pobūdį ir veikimo aplinką.

Įrenginį maitinant iš baterijos yra žinoma ir fiksuota jos talpa, kurios pokyčio matavimas leidžia apskaičiuoti suvartojamą elektros energiją. Būtina paminėti ir minusus. Maitinimo tikslumas iš baterijos gali būti su didele paklaida, kas priklauso nuo daugybės išorinių ir vidinių aplinkybių: baterijos kokybės, ciklų skaičiaus, aplinkos temperatūros, gilaus iškrovimo efekto, vėdinimo, įkrovimo įtampos kokybės.

Maitinant iš USB magistralės, tiekiama energija yra beribė. Dėl šios priežasties, būtinas papildomas apskaitos matavimo prietaisas, kuris leistų išmatuoti, kiek energijos iš maitinimo magistralės tiekiama į robotą. Tokį funkcionalumą gali suteikti srovės ir įtampos matuoklis, jungiamas į maitinimo šaltinio USB magistralę, kaip „tiltas“ tarp elektros energijos šaltinio ir apkrovos.

3.3.1. Įtampos ir srovės matuoklis

Autonominio roboto energijos suvartojimo matavimams naudojamas „KCX-017“ srovės ir įtampos matuoklis. Matuoklis jungiamas kaip tiltas, tarp elektros energijos šaltinio ir roboto, ir gali būti maitinamas iš išorinių baterijų, elektros tinklo, naudojant AC/DC galios keitiklį ar kito įrenginio, su stabiliu elektros maitinimu, pavyzdžiui, nuo stalinio kompiuterio.

3.4 pav. „KCX-017“ įtampos ir srovės matuoklis [74]

Mažo biudžeto įrenginys pasižymi itin geru tikslumu ir leidžia matuoti įtampos ir srovės reikšmes, atitinkamai su < +/- 1% ir < +/- 2% paklaidomis. Įėjimo ir išėjimo įtampų diapazonas yra nuo 3VDC iki 7VDC. Autonominis aplinkos stebėjimo robotas, idealiausiu atveju, turi būti maitinamas stabilia, 5V nuolatine įtampa. Kalbant apie įrenginio parametrus, būtina paminėti, jog matuoklio LCD ekrano duomenų atsnaujinimo dažnis yra 500 milisekundžių, tai reiškia, jog matuoti elektros parametrų kitimą galima tik kas 500 milisekundžių. Toks parametrų atnaujinimo dažnis ekrane yra daugiau nei pakankamas, norint gana tiksliai išmatuoti roboto suvartotą energiją.

3.3.2. Matavimo metodika

Vaizdo transliacijos metu išmatavus srovę, įtampą ir žinant operacijos laiką, galima suskaičiuoti, kiek energijos operacija sunaudojo. Toks skaičiavimo metodas yra tinkamas, jeigu visos operacijos metu, srovė yra pastovi ir įrenginio apkrovimas yra vienodas. Deja, bet AASR apibrėžtos operacijos metu, resursai naudojami netolygiai, todėl operacijos metu būtina fiksuoti kintančios srovės reikšmes laike. Toks matavimo metodas leis ypatingai tiksliai išmatuoti energijos suvartojimą. Srovės kitimą per laiką matuoja gana profesionalūs, specialūs įtaisai, pavyzdžiui, oscilografai. Apmaudu, tačiau tokie įtaisai ir kartu veikianti programinė įranga yra stebėtinai brangūs, todėl matavimams naudojamas minėtas USB magistralės įtampos, srovės matuoklis. Kameros ir chronometro pagalba, fiksuojamos skirtingos srovės reikšmės laiko intervaluose. Skaičiavimams atlikti naudojama galios formulė per laiką:

$$P_t = (U_{t_1} \cdot I_{t_1} \cdot t_1) + (U_{t_2} \cdot I_{t_2} \cdot t_2) + \ldots + (U_{t_n} \cdot I_{t_n} \cdot t_n); \quad (1)$$

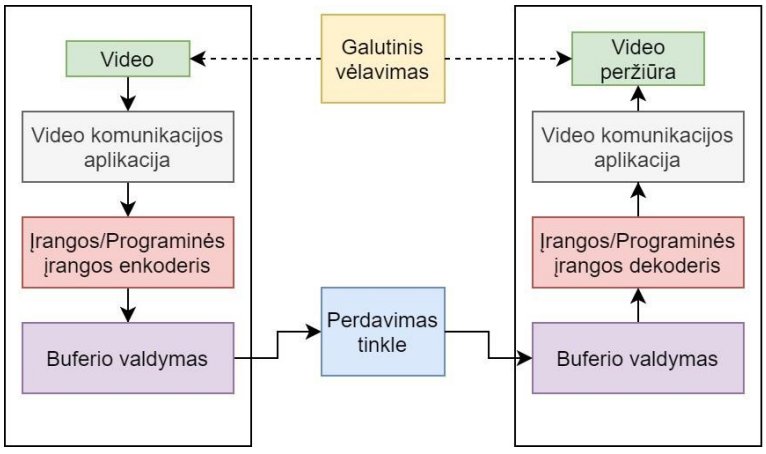
čia: P_t – galia, W s; U – įtampa V; I – srovė, A; t – laikas, s.

Elektros energijos suvartojimo matavimo bandymo pavyzdys pateikiamas 3.5 paveiksle.

3.5 pav. Elektros energijos suvartojimo matavimas operacijos metu

3.4. Vaizdo transliacijos vėlavimas

Komunikavimo metodui tirti, bus matuojamas galutinis vėlavimas, tai yra, laiko skirtumas, nuo realaus vaizdo fiksavimo iki vaizdo galiniame įrenginyje (angl. *Capture-to-Display Delay*). Galutinis vaizdo vėlavimas susidaro iš visų grandinės mazgų vėlavimo [75]. Tai galima atvaizduoti schematinėje diagramoje, kuri pateikta 3.6 paveiksle.



3.6 pav. Galutinio vėlavimo funkcinė schema

Vartotojų vaizdo vėlavimo toleravimas buvo tirtas daugelyje multimedijos programų. ITU-T G.114 standartas apibrėžia, jog vienos krypties (angl. *One-way*) „end-to-end“ transliacijos vėlavimas, planuojant bet kokias multimedijos programas ar sistemas, negali viršyti 400 ms [76]. AASR komunikavimo metodo transliacijoje, bus siekiama neviršyti šios rekomenduojamos gairės.

Siekiamas galutinis vaizdo vėlavimas, įskaitant visus, vėlavimą sukeliančius mazgus, negali būti didesnis nei 300 ms.

Tiriant transliacijos vėlavimą, naudojami identiški vaizdo parametrai, koderiai (angl. *Encoder*) ir kodavimo algoritmai, todėl transliacijos vėlavimas skirsis tik dėl skirtingų medijos protokolų, jų modifikacijų ir saugumo mechanizmų.

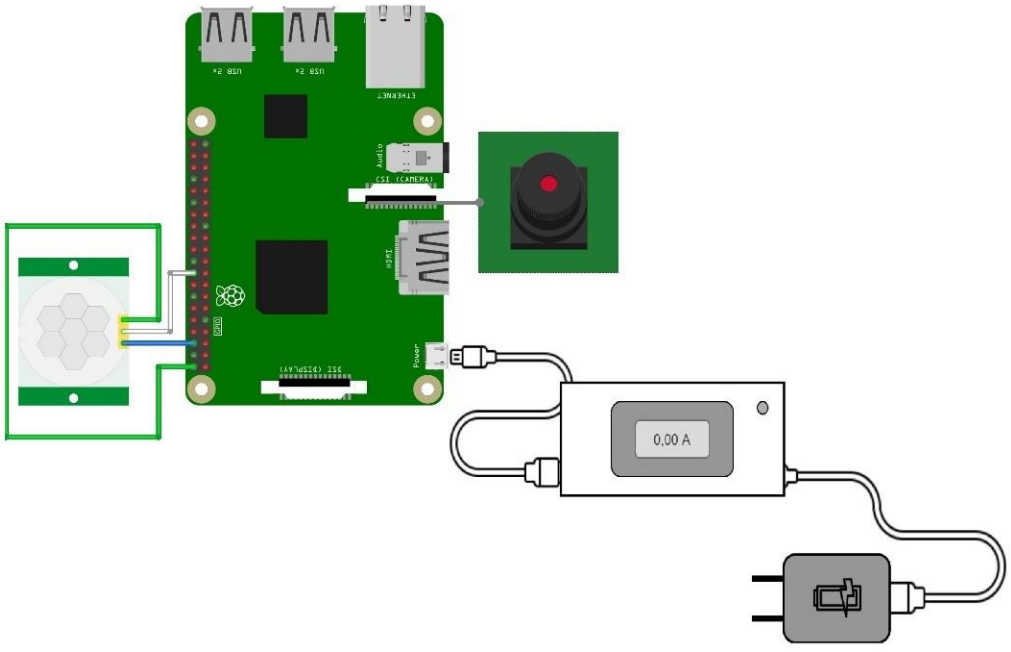
3.5. Protokolų greitaveika

Galutinis vaizdo vėlavimas, kuris matomas kliento įrenginyje, negali visiškai atskleisti, kurie protokolai veikia greičiau, o kurie lėčiau. Siekiant detaliau išnagrinėti protokolų greitaveiką, svarbu išmatuoti ir išsiaiškinti, kiek užtrunka transliacijos inicijavimas ir jos paleidimas, naudojant skirtingus protokolų rinkinius. Tai padės išsiaiškinti, pavyzdžiui, kokį vėlavimą sukelia DTLS rankos paspaudimo operacijos, kurios vyksta prieš pradedant fiksuoti vaizdą.

Norint išmatuoti, kada pradedami siųsti vaizdo duomenys ir kaip greitai suveikė protokolai, reikia išmatuoti laiką nuo judesio užfiksavimo (operacijos pradžios) robote ir fiksuoti laiką, kada atkeliavo pirmasis vaizdo duomenų paketas kliento kompiuteryje. Visa tai bus atliekama eksperimentinėje dalyje.

3.6. Autonominio aplinkos stebėjimo roboto sujungimų schema

Norint lengviau pavaizduoti visų roboto fizinių elementų sąryšį, kartu su papildoma įranga matavimams, kaip pagalbinę priemonę, galima pasitelkti funkcinę įrenginių sujungimų schemą. Poskyryje 3.2. jau paminėti pagrindiniai elementai yra jungiami tarpusavyje, užmaitinami per įtampos ir srovės matuoklį „KCX-017“, nuo buitinio, kintamosios srovės 230V elektros tinklo. PIR judesio jutiklis jungiamas per GPIO 2, 6 ir 24 kontaktus, atitinkamai, tai būtų DC 5V maitinimas, įžeminimas, PIR signalo išėjimas. Kameros modulis jungiamas per CSI kameros jungtį, naudojant lankstų CSI kabelį. Darbui paruoštos sistemos funkcinė sujungimų schema pateikiama 3.7 paveiksle.



3.7 pav. Funkcinė sujungimų schema

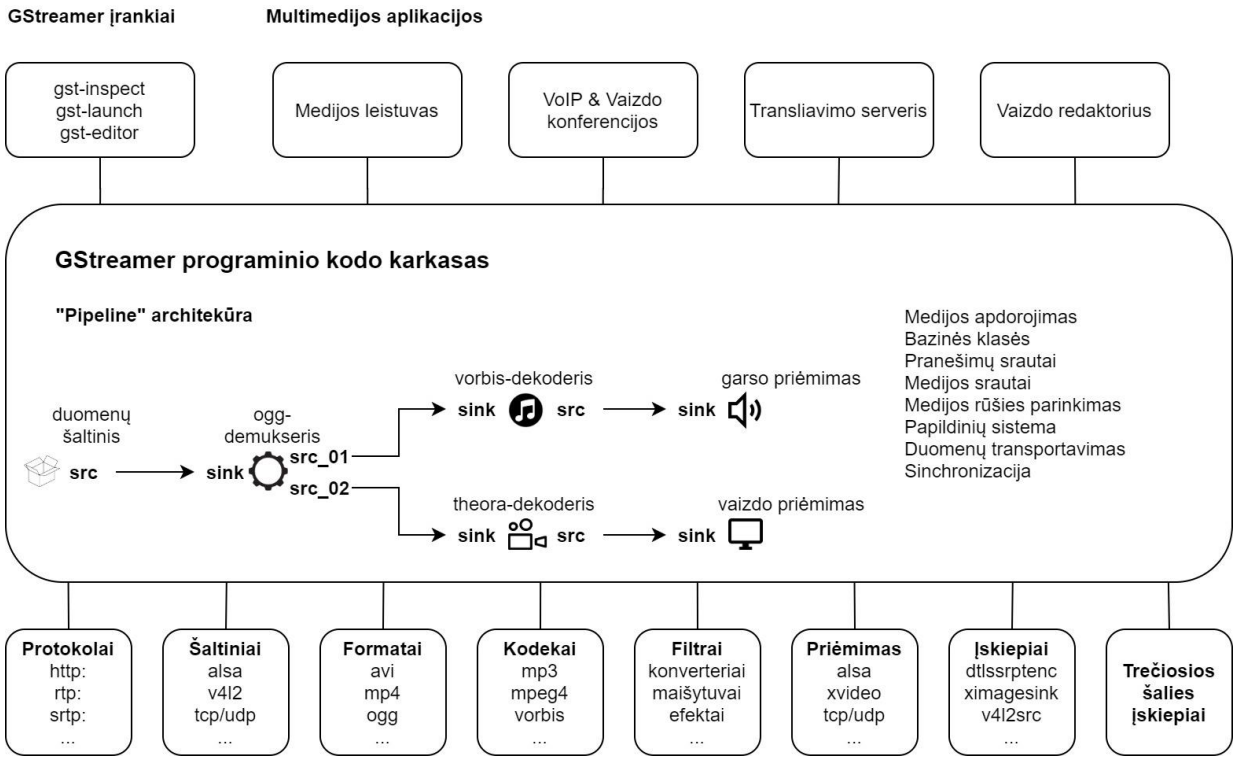
3.7. Programinė įranga, bibliotekos, karkasai

Saugaus komunikavimo metodo įgyvendinimui, vien fizinės įrangos nepakanka. Kuriamas prototipas yra multimedijos duomenų perdavimo sistema, o tai reiškia, jog šios sistemos paleidimui, konfigūravimui, testavimui, įgyvendinimui naudojama nemažai programinės įrangos, bibliotekų ir karkasų. Ypatingai, kai ši sistema projektuojama kaip saugaus komunikavimo sistema. Pats vaizdo apdorojimas ir transliavimas susideda iš daugelio etapų ir žingsnių, todėl sistemoje naudojamas karkasas turi palaikyti įvairius medijos duomenų tvarkymo ir apdorojimo metodus. Komunikavimo metodo prototipo įgyvendinime, vaizdo transliavimo karkasas yra vienas iš pagrindinių komponentų.

3.7.1. Medijos duomenų apdorojimo ir transliavimo karkasas

Prototipo įgyvendinimui pasirinktas lankstus, greitaveikis, daugialypis multimedijos karkasas „GStreamer“ (angl. *Framework*) [77]. Ypač galingas ir universalus, atvirojo kodo medijos programų kūrimo karkasas, paremtas „pipeline“ vykdymo architektūra, leidžiantis pasiekti minimalų skaičiavimo, energijos suvartojimo, atminties resursų naudojimą. Karkasas yra pilnai tinkamas kuriant aukščiausios klasės multimedijos transliavimo programas, keliančias aukštus vėlavimo reikalavimus ir užtikrinant paslaugų serviso kokybę. Veikimo architektūra paremta įskiepiais, kurie suteikia įvarias multimedijos duomenų kodavimo, pakavimo, modifikavimo funkcijas. Be ypatingai našių multimedijos programų kūrimo funkcijų, „GStreamer“ siūlo įvairius, konsolės pagrindu sukurtus įrankius, leidžiančius efektyviai kurti ir testuoti programos prototipą.

„GStreamer“ paremtas jau egzistuojančių ir laiko patikrintų multimedijos bibliotekų integravimu į „pipeline“ principu paremtą savo veikimo technologiją. Karkaso architektūros veikimo pavyzdys pateikiamas 3.8 paveiksle.



3.8 pav. „GStreamer“ atvirojo kodo karkaso architektūros pavyzdys [77]

čia: *src* – šaltinis; *sink* – priimantis elementas.

3.7.2. Papildomi įrankiai, programinė įranga

Šalia pagrindinio, vaizdo duomenų apdorojimo ir transliavimo karkaso, rikiuojasi visa begalė kitų programų ir įrankių. Norint atlikti visus reikalingus testavimus, matavimus, konfigūravimus ir pakeitimus, projektuojamame saugiame komunikavimo metode tarp autonominio roboto ir galinio įrenginio, pasitelkiamas nemažas skaičius programų ir papildomų įrankių. Pagrindinės programos ir įrankiai paminėti ir aprašyti 3.1 lentelėje.

3.1 lentelė. Pagrindinė papildoma programinė įranga, įrankiai

Programinė įranga, įrankis	**Paskirtis**	**Panaudojimo atvejis**	**Dalis**
„Wireshark“	Tinklo srauto analizatorius	Sistemos veikimo patikrinimas, protokolų greitaveikos, vėlavimo matavimai	Prototipo projektavimo, eksperimentinė
„Seconds Count“	Vizualus laikmatis Android telefone	Energijos suvartojimo kitimo laike matavimas	Eksperimentinė
„Videosnarf“	Medijos išgavimas iš tinklo srauto įvesties failo	Protokolų saugos mechanizmų veikimo patikrinimas	Prototipo veikimo, eksperimentinė
„Stopwatch“	Laikmatis Unix tipo operacinėms sistemoms	Vaizdo transliacijos vėlavimo matavimas	Eksperimentinė
„NTP“	NTP serverio diegimas, sistemos laiko sinchronizavimui tarp kompiuterinių sistemų tinkle	Laiko sinchronizavimas tarp įrenginių su tikslu išgauti tikslius matavimus	Eksperimentinė
„Docker“	Taikomųjų programų paleidimas, naudojant konteinerių technologiją	Videosnarf paleidimas iš įdiegtos 32 bitų Ubuntu 12.04 operacinės sistemos	Eksperimentinė

3.8. Metodo prototipo išvados

1. Metodo prototipo įgyvendinimui serverio ir kliento įrenginiuose naudojamos lanksčios, atvirojo kodo operacinės sistemos, sukurtos „UNIX“ pagrindu. Šių OS naudojimas suteikia galimybę modifikuoti esamas bibliotekas, karkasus, programas, ir visa tai pritaikyti, atliekant saugų multimedijos duomenų transliavimą.
2. Darbo tyrimams pasirinktas „Raspberry PI 3B+“ mikrokompiuteris, plačiai naudojamas įgyvendinant įvairaus pobūdžio IoT projektus. Judesio užfiksavimui ir vaizdo filmavimui pasirinkti „RPI Cam OV5647“ ir „HC-SR501“ komponentai. Suvartojamos elektros energijos matavimams atlikti, naudojamas „KCX-017“ įtampos ir srovės matuoklis.
3. Roboto operacijos metu, energijos suvartojimas kinta laike. Norint išgauti tikslius duomenis, atliekant elektros energijos suvartojimo matavimus, fiksuojamos srovės ir įtampos pokyčių reikšmės laikui bėgant.
4. Tyrimo metu bus matuojamas galutinis vaizdo vėlavimas ir laikas, nuo vaizdo užfiksavimo iki pirmojo multimedijos paketo kliento kompiuteryje. Šie matavimai leis palyginti vėlavimą ir protokolų greitaveiką, naudojant skirtingus protokolus ar jų rinkinius.
5. Prototipo įgyvendinimui pasirinktas kokybiškas, atvirojo kodo, multimedijos duomenų apdorojimo ir transliavimo karkasas „GStreamer“, kuris yra ypač galingas ir universalus multimedijos duomenų karkasas, leidžiantis kurti aukštos kokybės transliavimo programas.

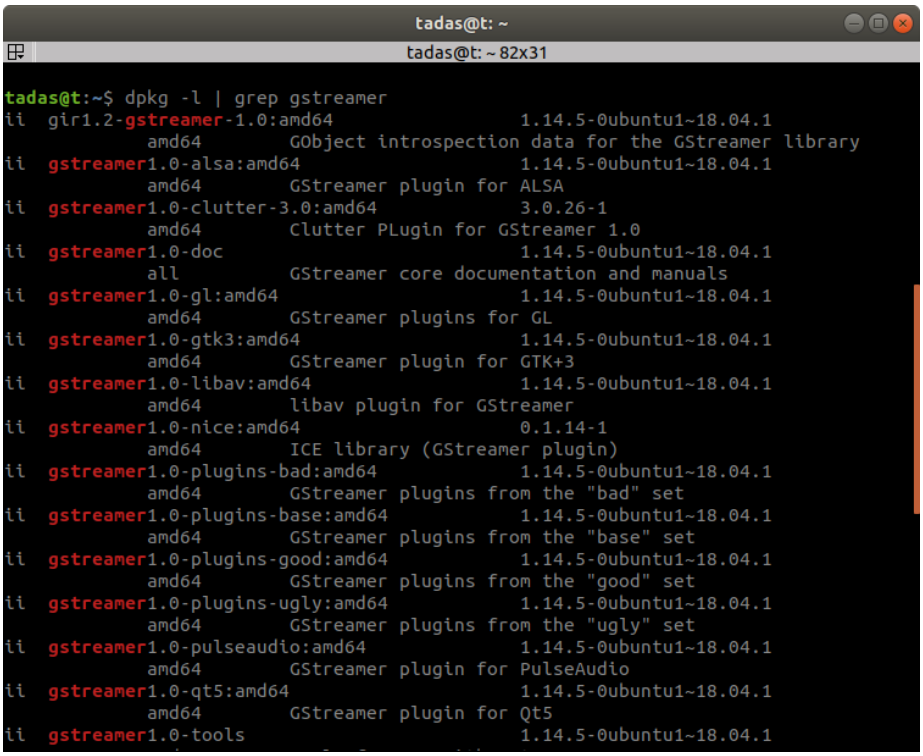
4. Autonominio aplinkos stebėjimo roboto saugaus komunikavimo metodo eksperimentiniai tyrimai

4.1. Įrenginių paruošimas darbui

Dėl praktiškai neribotų galimybių naudoti atvirojo kodo programas, bibliotekas ir kitus resursus, robote ir kliento kompiuteriuose diegiamos UNIX tipo operacinės sistemos „Ubuntu 18.04 LTS“ ir „Ubuntu Server 18.04 LTS“. Standartiškai, „Raspberry“ mikrokompiuteryje diegiama „Raspbian“ operacinė sistema, tačiau dėl suderinamumo ir norint turėti galimybę įdiegti visas reikalingas bibliotekas, naudojama „Ubuntu Server“ operacinė sistema, pritaikyta „Raspberry“ mikrokompiuteriams [78].

Prieš pradedant vaizdo duomenų transliavimą tarp autonominio aplinkos stebėjimo roboto ir kliento kompiuterio, abejose pusėse būtina įdiegti multimedijos duomenų karkasą „GStreamer“, kuris jau aprašytas 3.7.1. skyrelyje. „GStreamer“ iškart diegiamas su reikalingomis pagrindinėmis bibliotekomis ir įskiepių rinkiniais:

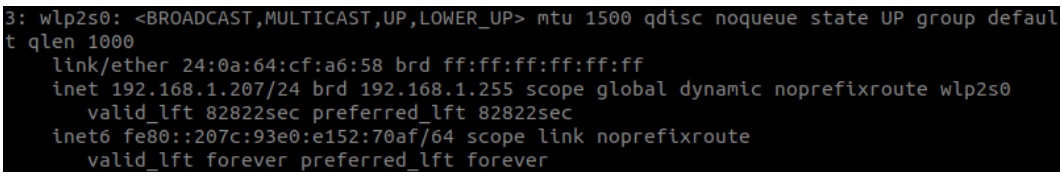
apt-get install libgstreamer1.0-0 gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly gstreamer1.0-libav gstreamer1.0-doc gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-qt5 gstreamer1.0-pulseaudio



4.1 pav. Dalis GStreamer įdiegtų bibliotekų, įskiepių, pagalbinių įrankių

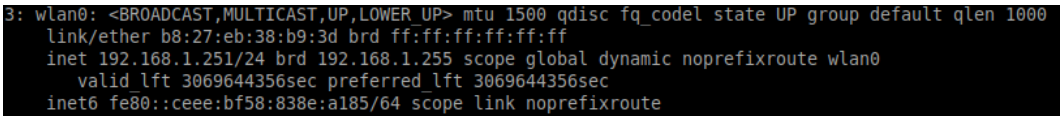
Papildomos bibliotekos, reikalingos komunikavimo metodo įgyvendinimui, įdiegiamos naudojimo metu. Vienos iš tokių bibliotekų pavyzdys yra CISCO kūrėjų *libsrtp-dev*.

Duomenų siuntimas iš roboto į kliento kompiuterį atliekamas tame pačiame, namų tinkle. Kliento įrenginio IP adresas *192.168.1.207*. Detalesnė įrenginio tinklo informacija pateikiama naudojant komandą *ip a* konsolėje:



4.2 pav. Kliento įrenginio informacija

Autonominio aplinkos stebėjimo roboto IP adresas *192.168.1.251*.



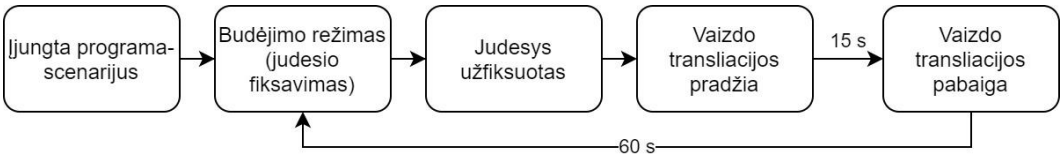
4.3 pav. Autonominio aplinkos stebėjimo roboto informacija

Duomenys siunčiami ir priimami per sutartą *5000* prievadą. Roboto programoje-scenarijuje nurodomas prievadas ir įrenginio IP, į kurį transliuojami vaizdo duomenys, o kliento pusėje nurodomas prievadas, per kurį priimami vaizdo duomenys.

4.2. Operacijos programa-scenarijus

Sukurto komunikavimo metodo prototipo operacijos įgyvendinimui rašoma programa-scenarijus, kuris leis sujungti ir suderinti visus operacijos mazgus, pradedant judesio užfiksavimu ir baigiant vaizdo transliavimu klientui. Programa-scenarijus rašomas Python kalba, dėl universalumo, bibliotekų gausos ir suderinamumo su įvairiais įrenginiais. Būtina paminėti, jog didelė dalis „Raspberry“ programų ir bibliotekų parašytos Python programavimo kalba, todėl ši kalba pasirinkta komunikavimo metodo kūrimui ir įgyvendinimui. Naudojama ir BASH scenarijų kalba. Roboto atliekamos operacijos aprašymas pateiktas 2.1.1. skyrelyje, o detalus algoritmas pateiktas BPMN veiklos procesų diagramoje 2.5. poskyryje.

Sėkmingam AASR operacijos ir saugaus vaizdo transliavimo įgyvendimui, programa-scenarijus turi atlikti algoritmą, tai yra, veiksmus nurodyta seka. Reikalingi žingsniai ir jų seka atvaizduota supaprastintoje sekos diagramoje, pateiktoje 4.4 paveiksle. Programos baigimo sąlygos yra, jeigu vaizdo duomenys nėra gaunami daugiau nei 5 valandas (klientas) arba įrenginys išsikrauna (robotas).



4.4 pav. Supaprastinta programos-scenarijaus sekos diagrama

Programos sisteminiuose argumentuose nurodomas gavėjo IP adresas ir prievadas, šiuo atveju *192.168.1.207* ir prievadas *5000*. Šie *sys.argv[1]* ir *sys.argv[2]* argumentai priskiriami kintamąjam *udpsink host* ir *port.*

Roboto sėkmingai atliktos operacijos pavyzdys, iš roboto perspektyvos, pateiktas 4.5 paveiksle. Robotas prijungtas prie išorinio vaizduoklio, siekiant užfiksuoti ir atvaizduoti programos veikimą.

```
ubuntu@ubuntu:~/Desktop/matavimas_r$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether b8:27:eb:6d:ec:68 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether b8:27:eb:38:b9:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.251/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 3069644356sec preferred_lft 3069644356sec
    inet6 fe80::ceee:bf58:838e:a185/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
ubuntu@ubuntu:~/Desktop/matavimas_r$ python testSRTP_pir.py 192.168.1.207 5000
2020-03-21 17:42:23 Programa veikia budejimo rezime.. Fiksuojamas judesys..
2020-03-21 17:42:29 Judesys uzfiksuotas. Pradedame tiesiogine vaizdo transliacija
2020-03-21 17:42:44 Transliacija baigta. Po 60 sekundziu griztame i budejimo rezima.
2020-03-21 17:43:44 Programa grizo i budejimo rezima.. Fiksuojamas judesys..
2020-03-21 17:43:50 Judesys uzfiksuotas. Pradedame tiesiogine vaizdo transliacija
2020-03-21 17:44:05 Transliacija baigta. Po 60 sekundziu griztame i budejimo rezima.
^CIsjungiama..
ubuntu@ubuntu:~/Desktop/matavimas_r$
```

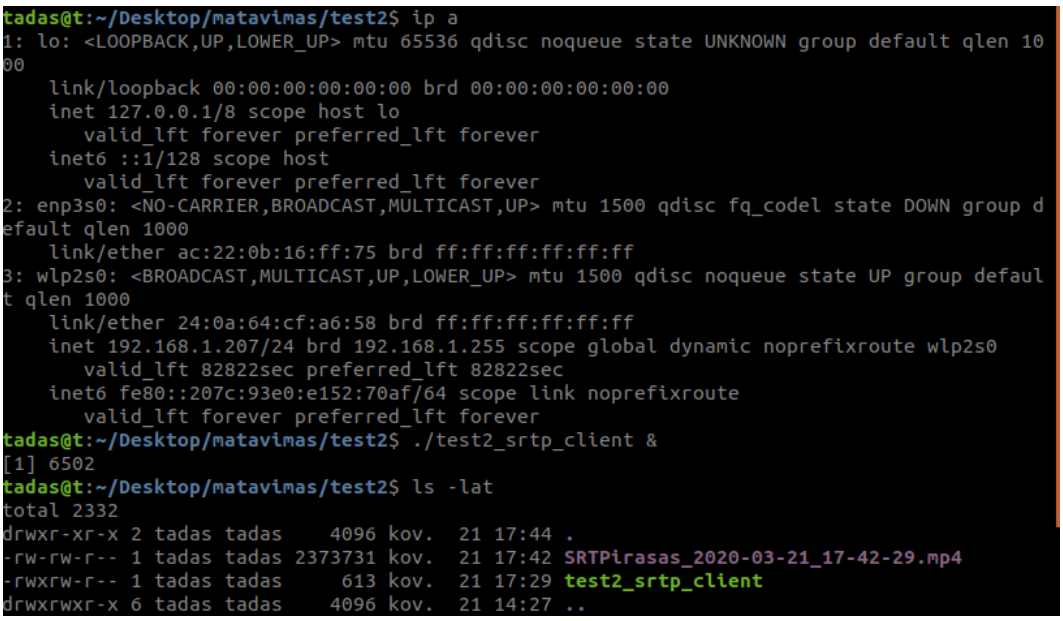
4.5 pav. Sėkminga vaizdo transliavimo operacija

4.5 paveiksle matyti, jog po keletos sekundžių veikimo budėjimo režime, buvo užfiksuotas judesys. Nedelsiant, pradėta tiesioginė vaizdo transliacija į kliento įrenginį. Po 15 sekundžių pasibaigus transliacijai, robotas pereina į „atvėsimo“, ramybės laikotarpį (angl. *Cooldown time*), o po 60 sekundžių pereinama į budėjimo režimą, kurio metu toliau fiksuojamas judesys.

Kliento įrenginyje, veikiant programai ir robotui užfiksavus judesį, iškyla atskiras langas, kuriame matomas tiesioginis vaizdas iš autonominio aplinkos stebėjimo roboto.

4.6 pav. Tiesioginis vaizdas iš autonominio aplinkos stebėjimo roboto

Pasibaigus vaizdo transliacijai, kuri trunka 15 sekundžių, papildomai įrašomas tiesioginis vaizdas *.mp4* formatu. Šis išsaugomas pagal nurodytą pavadinimo formatą, kuris bus aprašomas tolimesniame skyriuje. Jame bus detaliau nagrinėjamos funkcijos, naudojamos kliento įrenginyje.



4.7 pav. Vaizdo įrašas kliento kompiuterio kietajame diske

4.2.1. Vaizdo apdorojimas ir perdavimas

Kaip jau aptarta antrame ir trečiame skyriuose, vaizdo apdorojimui, pakavimui, transliavimui ir priėmimui naudojamas „GStreamer“ karkasas, bibliotekos ir įskiepiai. Visi reikalingi įrankiai ir parametrai intregruojami pagrindinėje, Python programavimo kalba parašytoje programoje, kuri apjungs visus operacijos elementus, įrankius ir inicijuos vaizdo transliavimą iš autonominio aplinkos stebėjimo roboto, užfiksavus judesį. Kliento kompiuteryje naudojama BASH scenarijų kalba.

4.2.2. Programos veikimas budėjimo režime ir judesio užfiksavimas

Kadangi vienas pagrindinių uždavinių, kuriant komunikavimo metodą, yra kaip įmanoma mažesnis energijos suvartojimas, būtina užtikrinti, jog visuose AASR operacijos fazėse šis tikslas būtų įgyvendinamas. Įrenginys, prieš pradedant vaizdo transliaciją, daug laiko veiks budėjimo režime, todėl labai svarbu, jog ir šioje dalyje energijos suvartojimas būtų, taip pat, kaip įmanoma mažesnis, nors tai ir nėra pagrindinė dalis, į kurią orientuotas darbo tyrimas.

Dažnais atvejais, eksploatuojant panašią sistemą, esant budėjimo režime, naudojamas nuolatinis GPIO gnybtų būsenos tikrinimas (angl. *GPIO polling*). Tai reiškia, jog visą laiką yra vykdomas begalinis ciklas, kuris nuolatos tikrina ar gnybtuose yra gautas aukštas signalas, indikuojantis, jog judesys užfiksuotas. Kaip žinoma, bet koks nuolatinis ciklas naudoja daugybę skaičiuojamųjų resursų, todėl programai veikiant tokiu būdu, įrenginys jau gali būti išsikrovęs taip ir nesulaukęs judesio užfiksavimo. Siekiant išvengti tokio scenarijaus, naudojama „RPI GPIO“ bibliotekos funkcija *GPIO.add_event_detect()*. Naudojant šią funkciją, vietoje nuolatos tikrinamo GPIO gnybto, programa lauks, kol gnybtas gaus trukdį (angl. *Interrupt*). Trukdis gali būti signalo pasikeitimas iš aukšto į žemą, arba, šiuo atveju, iš žemo į aukštą, kas reikš įvykį – judesio užfiksavimą. Šiuo metodu, programai veikiant budėjimo režime, naudojama tiek pat energijos, kiek robotui veikiant ramybės būsenoje (angl. *Idle*).

```
GPIO.setmode(GPIO.BCM)
PIR_PIN = 24
GPIO.setup(PIR_PIN,GPIO.IN)

def Motion(PIR_PIN):
    GPIO.remove_event_detect(PIR_PIN)
    print datetime.now().strftime('%Y-%m-%d %H:%M:%S'), "Judesys uzfiksuotas. Pradedame tiesiogine vaizdo transliacija"
    os.system(CreateCommand(applicationName, parameters, values, pipes))
    print datetime.now().strftime('%Y-%m-%d %H:%M:%S'), "Transliacija baigta. Po 60 sekundziu griztame i budejimo rezima."
    time.sleep(60)
    GPIO.add_event_detect(PIR_PIN, GPIO.RISING, callback=Motion)
    print datetime.now().strftime('%Y-%m-%d %H:%M:%S'), "Programa grizo i budejimo rezima.. Fiksuojamas judesys.."
print datetime.now().strftime('%Y-%m-%d %H:%M:%S'), "Programa veikia budejimo rezime.. Fiksuojamas judesys.."

try:
    GPIO.add_event_detect(PIR_PIN, GPIO.RISING, callback=Motion)
    while 1:
        time.sleep(100)
except KeyboardInterrupt:
    print "Isjungiama.."
    GPIO.cleanup()
```

4.8 pav. Operacijos programos judesio užfiksavimo dalies fragmentas

4.3. Vaizdo duomenų transliavimas

4.3.1. Transliacijos parametrai

Viso tyrimo metu, išbandant skirtingus protokolus ir jų konfigūracijas, vaizdo transliacijos parametrai naudojami tokie patys. Naudojant vienodus vaizdo transliacijos fiksavimo parametrus, eksperimentai, orientuoti į protokolų greitaveiką, vaizdo vėlavimą, energijos suvartojimą bus lengviau palyginami ir tikslesni. Vaizdui gauti naudojama v4l2 (angl. *Video4Linux2*) įrenginio tvarkyklė, kuri padeda fiksuoti vaizdą ir išvesti pirminius, neapdorotus (angl. *Raw*) duomenis į tolimesnius vaizdo apdorojimo karkaso mazgus. Pirminiam vaizdo duomenų kodavimui naudojamas *x264enc()* karkaso elementas, kuris gautus, neapdorotus duomenis koduoja į H.264 formato duomenis, o šie, konteineriuojami į plačiau žinomo MPEG-4 AVC formato duomenis. H.264 formato kodavimas buvo sukurtas siekiant kokybiškesnio vaizdo pakavimo skirtingoms programoms, įskaitant vaizdo transliavimą internetu. Pagrindinis šio kuriamo standarto tikslas buvo gebėti paruošti aukštos kokybės vaizdo įrašą su kuo mažesnėmis sąnaudomis, lyginant su anksčiau naudojamais standartais, nepadidinant įdiegimo sunkumo lygio [79]. Šis vaizdo kodavimo standartas pasižymi puikiu našumu ir resursų naudojimu, leidžiant H.264 subalansuotai naudoti galią kodavimui ir dekodavimui. Verta paminėti, jog viena iš H.264 naudojimo priežasčių yra tai, jog šis standartas yra lengvai pritaikomas naudojimui skirtinguose įrenginiuose, įskaitant ir „Raspberry PI“, kuris turi galimybę koduoti ir dekoduoti minėto standarto vaizdo įrašus. Tyrimo metu naudojami pagrindiniai vaizdo transliacijos parametrai aprašyti 4.1 lentelėje.

4.1 lentelė. Transliacijos parametrai

Parametras	**Reikšmė**
Plotis, pikseliai	1280
Aukštis, pikseliai	720
Kadrai per sekundę	30
Įvestis	/dev/video0
Įvesties tvarkyklė	v4l2
Kodavimas	H.264
Procesoriaus branduoliai	4

Prototipo realizavimo metu, atliekant aplinkos stebėjimo operaciją, naudojami visi keturi procesoriaus branduoliai (angl. *Quad-core*). Transliavimo karkase konfigūruojama *„multi-thread“* technologija vaizdo perdavimui. Šiuo atveju, mikrokompiuterio apkrovimas išlieka optimalus ir nei vienas iš procesoriaus branduolių neviršija 50 % apkrovimo. Naudojant visus procesoriaus branduolius, efektyviai naudojami visi prieinami įrenginio pajėgumai. Naudojant tik vieną iš keturių branduolių, procesoriaus apkrovimas operacijos metu siekia apie 98 %, todėl atsiranda potenciali rizika, jog mikrokompiuterio darbas gali sutrikti, o transliavimo programa sustoti. Tokiu atveju, be papildomo aušinimo ir vėdinimo įrenginys gali fiziškai sugesti.

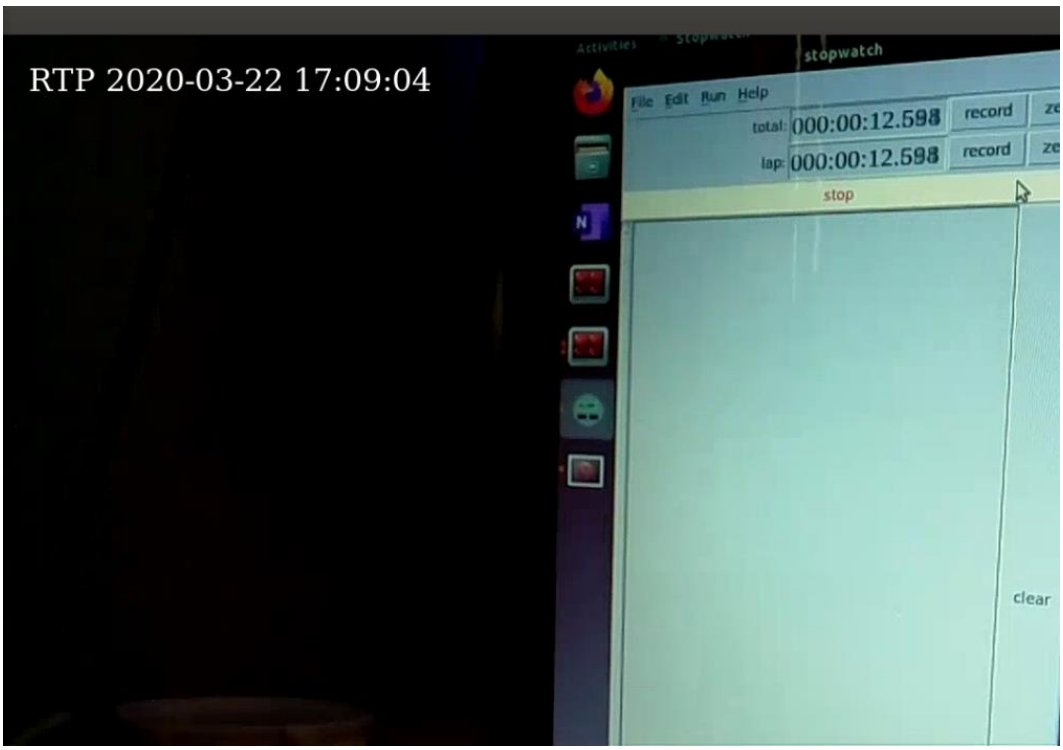
```
values = {"device": "/dev/video0",
          "width": "1280",
          "height": "720",
          "application": "x-rtp",
          "host": host,
          "port": port,
          "framerate": "30/1",
#H264 enc options
          "speed-preset": "ultrafast",
          "tune": "zerolatency",
          "byte-stream": "true",
          "threads": "4",
          "key-int-max": "15",
          "intra-refresh": "true"
          }

pipes = ["v4l2src device={device}", "video/x-raw,width={width},height={height},framerate={framerate}",
         "x264enc speed-preset={speed-preset} tune={tune} byte-stream={byte-stream} threads={threads} key-$
         "rtph264pay",
         "\"application/{application}, payload=(int)96, ssrc=(uint)1356955624\"",
         "udpsink host={host} port={port}"]
```

4.9 pav. Vaizdo transliacijos parametrai

4.3.2. RTP medijos duomenų transliavimo įgyvendinimas

Nusprendus, kokie bus naudojami vaizdo transliacijos parametrai ir įdiegus „GStreamer“ karkasą kliento, serverio įrenginiuose, konfigūruojamas RTP vaizdo duomenų perdavimas realiu laiku. Naudojama *rtph264pay()* funkcija, kuri pakuoja H264 koduotės vaizdą į RTP paketus (RFC 3984). Nurodomi būtini parametrai vaizdo transliavimui, naudojant RTP protokolą – sinchronizacijos šaltinio identifikatorius (angl. *SSRC*) ir duomenų tipas. Pabrėžtina, jog šie parametrai nurodomi kliento ir serverio programose. Duomenų tipo konfigūracijos nesutapimo atveju, abejose pusėse, duomenys nėra priimami (nesuderinamumas).



4.10 pav. RTP vaizdo transliacija tyrimo metu

4.3.3. SRTP medijos duomenų transliavimo įgyvendinimas

Vaizdo duomenų tiesioginis transliavimas, naudojant modifikuotą SRTP protokolą yra pagrindinis tyrimo objektas. Reikalingos protokolo modifikacijos ir konfigūravimas, atliekamas „GStreamer“ karkase. Komunikavimo metodo prototipas suprojektuotas ir įgyvendinamas, naudojant iš anksto pasidalinto rakto techniką, atsisakant kelių papildomų paketo antraštės laukų. Šis raktas talpinamas programiniame kode arba atskirame faile. Abiems šalims (klientui ir serveriui) turint identišką SRTP pagrindinį raktą, atliekama vaizdo transliacija iš roboto į kliento kompiuterį. Pavyzdinė rakto reikšmė, kuri galėtų būti naudojama SRTP komunikavimo metode:

98765432100123456789012345678901234567890123456789012345678888

Priklausomai nuo naudojamų bibliotekų ir karkasų, SRTP raktas priimamas skirtingais formatais: HEX, DEC, BASE64. GStreamer rakto įvestį priima tekstiniu ASCII formatu.

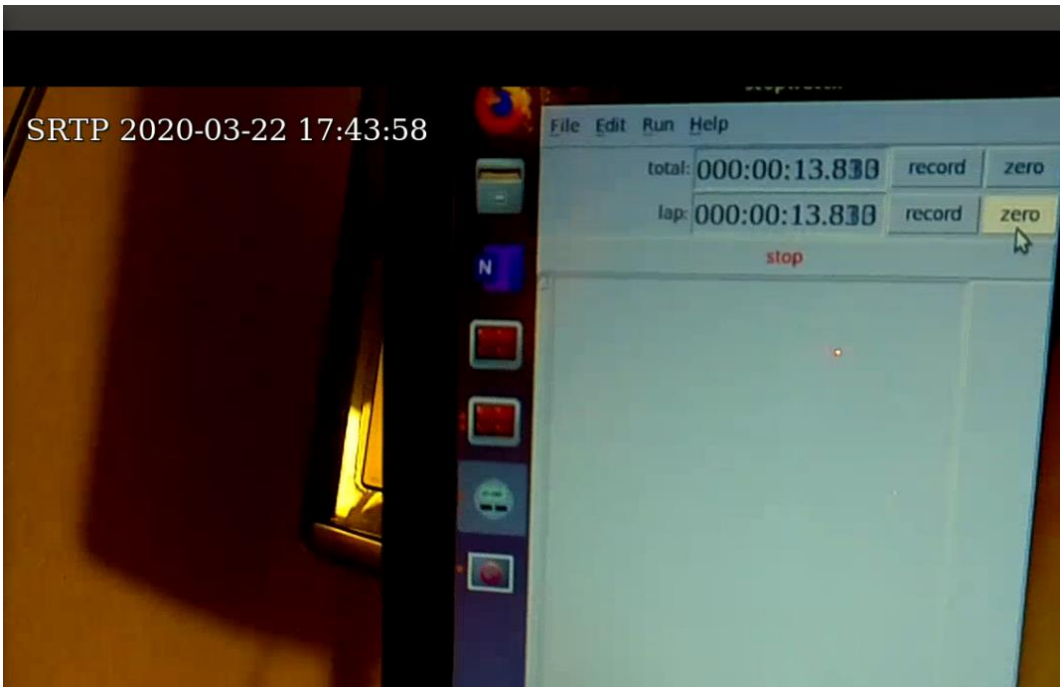
```
values = {"device": "/dev/video0",
          "width": "1280",
          "height": "720",
          "application": "x-rtp",
          "key": GetKey("/home/ubuntu/Desktop/matavimas_r/raktas.key"),
          "host": host,
          "port": port,
          "framerate": "30/1",
```

4.11 pav. Rakto įvestis iš išorinio failo

Transliuojant vaizdą, naudojant modifikuotą SRTP protokolą, be minėtų SSRC ir duomenų tipo nustatymų, apsaugos funkcijų parametrų aprašymui, naudojama pagrindinė funkcija *srtpenc()*. Ši funkcija priima parametrus, kurie aprašo pagrindinį raktą, autentifikavimo tipą, šifravimo algoritmą.

Autentifikavimo tipas ir naudojamas šifravimo algoritmas, apibrėžia apsaugos lygio stiprumą. Konfigūruojami ir pasirenkami šie nustatymai:

- *key=98765432100123456789012345678901234567890123456789012345888;*
- *rtp-cipher=aes-128-icm;*
- *rtp-auth=hmac-sha1-80.*



4.12 pav. Modifikuoto veikimo SRTP vaizdo transliacija tyrimo metu

Pasirinktas 128 bitų rakto dydis, naudojant AES šifravimo algoritmą. Šis pasirinkimas argumentuojamas tuo, jog robotas veiks namų aplinkoje ir toks šifravimo stiprumas yra daugiau nei pakankamas, taip pat, atsižvelgiama ir į vieną iš tyrimo objektų – taupų energijos vartojimą. Vidutinis procesoriaus apkrovimas šifravimo metu, naudojant 128 bitų raktą, yra apie 10 % mažesnis, nei naudojant 256 bitų raktą [80]. Tarp procesoriaus apkrovimo ir energijos suvartojimo yra tiesioginė proporcija, todėl kuo mažiau apkraunamas procesorius, tuo daugiau energijos sutaupo robotas. Šis žymus energijos sutaupymas yra labai svarbus ribotų išteklių, autonominių įrenginių grupės prietaisams.

4.3.4. SRTP-DTLS medijos duomenų transliavimo įgyvendinimas

Naudojant SRTP-DTLS SRTP-DTLS metodo architektūros prototipo įgyvendinimui naudojamos šios funkcijos:

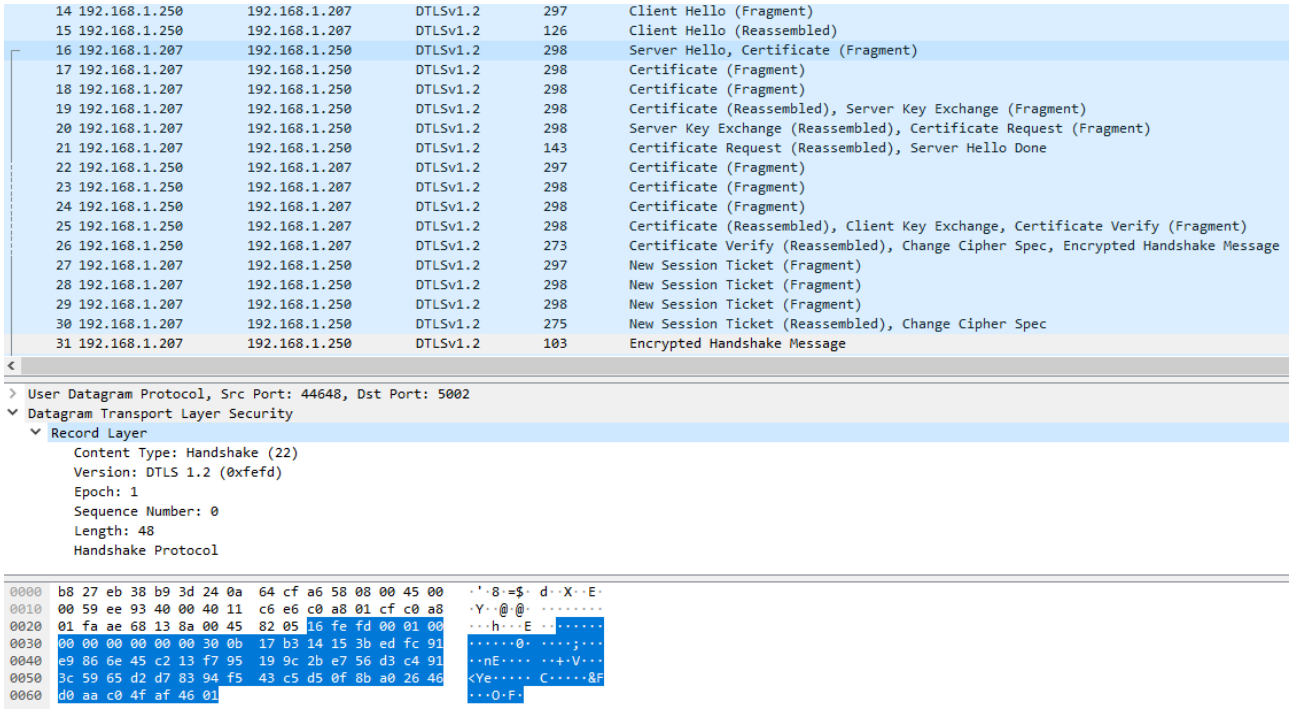
- *dtlsrtpenc()* – koduoja SRTP paketus su raktu, gautu DTLS protokolu;
- *dtlssrtpdec()* – dekoduoja SRTP paketus su raktu, gautu DTLS protokolu.

Šioje architektūroje SRTP raktas generuojamas atsitiktinai ir kiekvieną kartą yra skirtingos reikšmės.

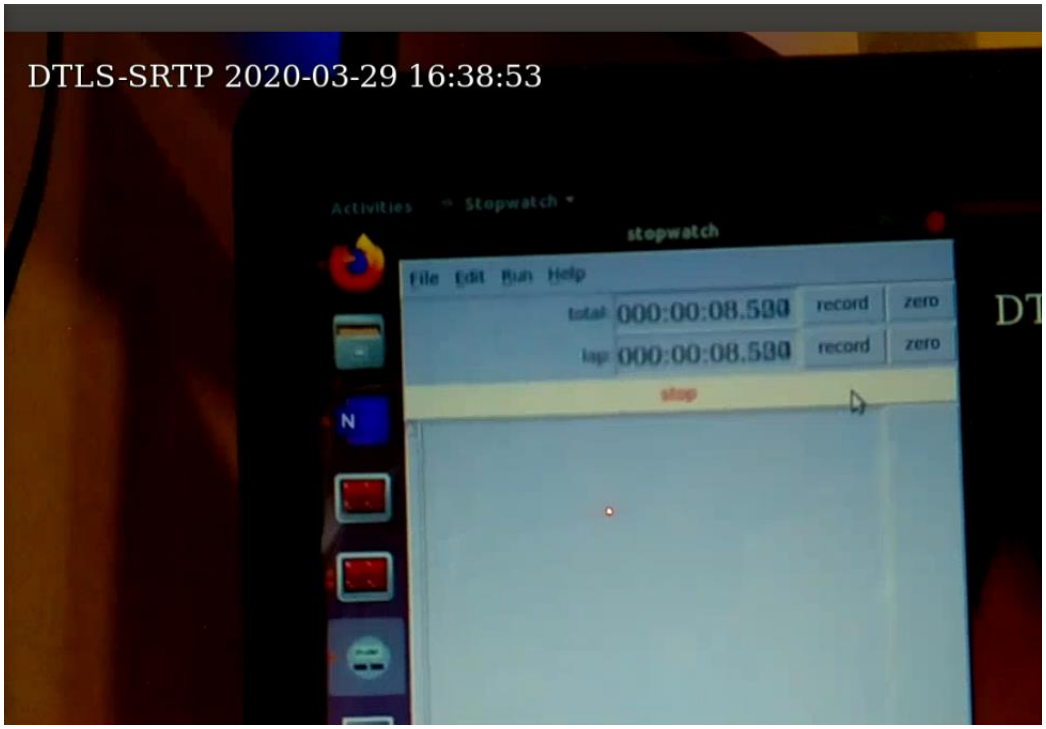
Nors vaizdo transliavimas yra vienakryptis, iš roboto į kliento kompiuterį, SRTP-DTLS architektūroje, naudojamos *dtlssrtpenc(), dtlssrtpdec()* funkcijos ir kliento, ir roboto pusėse. Tai galima paaiškinti tuo, jog ši funkcija yra sudaryta iš keleto elementų, įskaitant ir jau minėtą *srtpenc()*

(šioje architektūroje šifravimą atlieka SRTP). Nepaisant to, jog vaizdas keliauja viena kryptimi, šios funkcijos būtinos DTLS sesijos inicijavimo algoritmams aprašyti, kurie vyksta abejomis kryptimis.

DTLS abipusis bendravimas ir duomenų apsikeitimas tarp įrenginių vyksta per *5000* ir *5002* prievadus. Sėkmingai atlikus DTLS rankos paspaudimo algoritmą, SRTP duomenys siunčiami per tą patį 5000 prievadą. DTLS sesijos užmezgimo žingsniai, matomi „WireShark“ programos užfiksuotame tinklo duomenų sraute, pavaizduoti paveiksle 4.13. Šis algoritmas detaliau išnagrinėtas 2.3.5. skyrelyje.



4.13 pav. Užfiksuotas DTLS rankos paspaudimo algoritmas WireShark programoje



4.14 pav. SRTP-DTLS vaizdo transliacija tyrimo metu

4.3.5. Vaizdo priėmimas kliento įrenginyje

Kliento įrenginyje priimami duomenys, atliekamas jų dekodavimas ir konvertavimas į reikiamą formatą. Priėmimui svarbiausia nurodyti prievadą *udpsrc port=5000*, kuriuo klausomasi atkėliaujančių duomenų ir duomenų tipą, šiuo atveju *payload=(int)96*. Naudojamos šios pagrindinės funkcijos duomenų priėmimui ir dekodavimui:

- *rtpjitterbuffer()* – elementų sekos organizavimas ir dublikuotų RTP paketų panaikinimas;
- *rtph264depay()* – H.264 vaizdo išgavimas iš RTP paketų;
- *avdec_h264()* – H.264 kodavimo formato vaizdo dekodavimas;
- *videoconvert()* – vaizdo konvertavimas į kadrus, kurie gali būti leidžiami per leistuvą;
- *vaapisink()* – vaizdo kadrų apdorojimas ir paleidimas per leistuvą.

SRTP protokolo panaudojimo atveju, naudojama papildoma funkcija:

- *srtpdec()* – dekoderis, kuris naudodamas identišką raktą, iššifruoja, autentifikuoja SRTP paketus ir paverčia juos RTP paketais.

Sėkmingam SRTP-DTLS metodo architektūros prototipo įgyvendinimui naudojamos komandos:

- *dtlsrtpenc()* – sesijos užmegimo funkcija, inicijuojanti ir atliekanti DTLS operacijas. Taip pat, šifruoja SRTP paketus su raktu, gautu DTLS protokolu;
- *dtlssrtpdec()* – sesijos užmegimo funkcija, inicijuojanti ir atliekanti DTLS operacijas. Taip pat, iššifruoja SRTP paketus su raktu, gautu DTLS protokolu.

4.3.4. skyrelyje paaiškinta, jog dėl dvipusio DTLS duomenų apsikeitimo reikalingos šifravimo ir iššifravimo komandos kartu, nepaisant to, jog vaizdo duomenys siunčiami viena kryptimi.

Kol vaizdas transliuojamas tiesiogiai ir matomas ekrane realiu laiku, įrašas yra saugomas į įrenginio kietąją atmintį. Naudojamos sekančios funkcijos:

- *tee()* – duomenų priėmimas per šaltinio elementą (angl. *src*) ir išskaidymas į skirtingus priėmimo elementus (angl. *sink*). Ši funkcija naudojama, kai, pavyzdžiui, yra stebimas vaizdas medijos leistuve ir tuo pačiu įrašinėjamas į įrenginio kietojo disko atmintį;
- *h264parse()* – H.264 duomenų srauto analizavimas (reikalingas nustatyti formatui);
- *mp4mux()* – H.264 kodavimo formatą konvertuoja į daugeliui leistuvų priimtiną formatą MPEG-4 (.mp4);
- *filesinklocation()* – nurodoma vieta kietajame diske, kur vaizdo įrašas bus išsaugomas. Dėl patogesnio failų tvarkymo ir atsekamumo, galima papildomai formatuoti failo pavadinimą, pavyzdžiui, *„SRTPirasas_`date '+%Y-%m-%d_%H-%m'`.mp4"*.

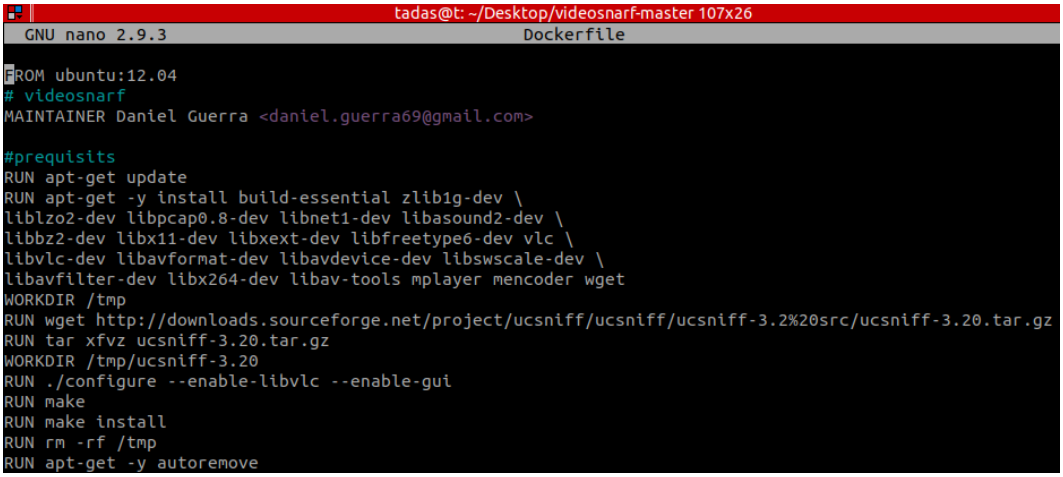
4.4. Apsaugotas, tiesioginis vaizdo duomenų perdavimas

Siekiant patikrinti, jog transliacijos metu duomenys siunčiami šifruoti, įprastai naudojami tinklo duomenų srauto analizatoriai, tokie kaip "WireShark", "tcpdump", "NetworkMiner" ar kt. Išgavus duomenų srautą, analizuojami paketai ir jų turinys. Dažniausiai, siunčiant tekstinius duomenis, nesunku atskirti, kurie paketai yra šifruoti, o kurie atviri, kadangi paketo duomenų dalyje galime matyti aiškų tekstą. Transliuojant medijos duomenis, paketo naudingieji duomenys sudaryti iš daugybės neaiškių simbolių, nepriklausomai ar duomenys šifruoti, ar ne. Daugelis, geriausių tinklo srauto paketų analizatorių, neturi funkcionalumo, kuris leistų atskirti RTP paketą nuo SRTP, todėl

abiejų protokolų naudojimo atveju, paketai žymimi, kaip paprasti RTP paketai. Kadangi paketo duomenų dalis sudaryta iš tokio pačio tipo simbolių, atskirti, kurie duomenys yra šifruoti, o kurie ne, yra sunku, tačiau įmanoma.

Tam pasitelkiama ir naudojama "VideoSnarf" taikomoji programa. Ši programa yra saugumo įvertinimo įrankis, kuris iš interneto duomenų srauto .pcap įvesties failo užfiksuoja medijos duomenų srautą ir konvertuoja jį į formatą (H.264), kurį galima peržiūrėti. Įrankis tinkamas aptikti RTP protokolo duomenų srautą, kuris koduotas H.264 vaizdo kodeku. Būtent šis kodavimo metodas naudojamas prototipo sistemoje. Peržiūrai rekomenduojamas programos kūrėjų siūlomas "mplayer" vaizdo leistuvas.

Kadangi „VideoSnarf“ taikomoji programa nėra vystoma jau keletą metų, jos diegimas įmanomas tik senesnėse „Ubuntu“ *OS* versijose. Dėl šios priežasties programa diegiama pasinaudojus „Docker“ platformą, kurios pagalba įsidiegiamas virtualus „Ubuntu 12.04“ OS konteineris, o jame instaliuojama reikalinga taikomoji programa.

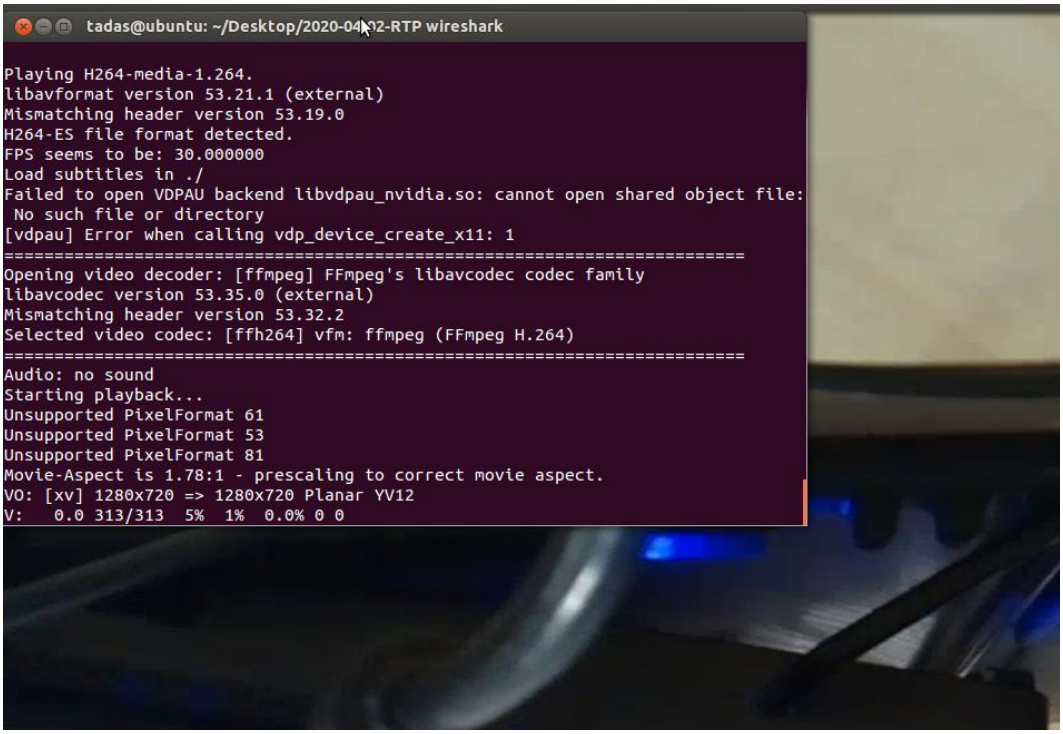


4.15 pav. „VideoSnarf“ diegimas

Duomenų paketų srauto fiksavimui, naudojama minėta programa "Wireshark". Transliacijos metu, fiksuojamas siunčiamų, į kliento kompiuterį, duomenų srautas ir išsaugomas .pcap formatu. Būtent šio formato failas naudojamas kaip įvestis. Medijos srauto išgavimui naudojama komanda:

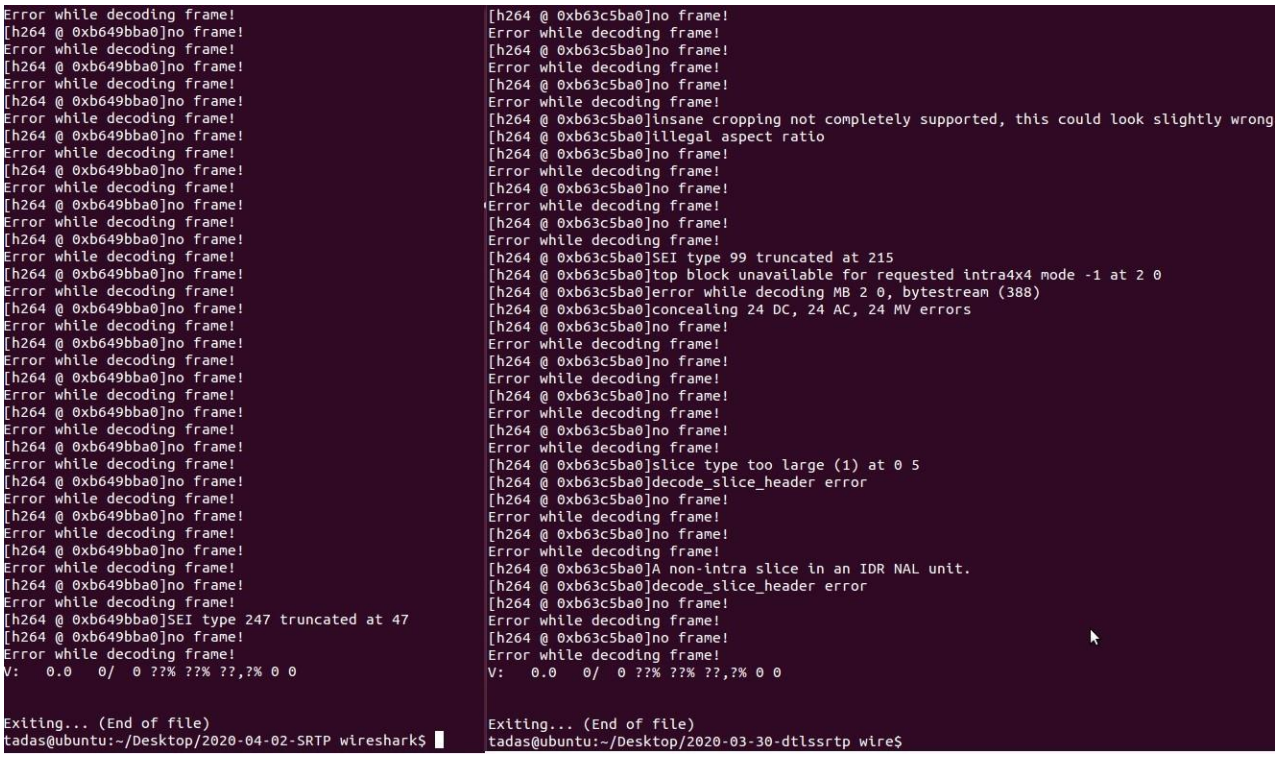
videosnarf -i duom_sraut.pcap

Gaunamas H.264 formato išvesties failas, kuris naudojamas išgautos vaizdo transliacijos peržiūrai. Atkūrus vaizdą iš RTP paketų srauto, išgaunama matoma transliacija:



4.16 pav. RTP vaizdo duomenų atkūrimas

Bandant išgauti matomą vaizdą iš SRTP ir SRTP-DTLS protokolų tinklo duomenų srauto, gaunami klaidos pranešimai, kadangi duomenys yra šifruoti:



4.17 pav. Nesėkmingas modifikuoto SRTP, SRTP-DTLS duomenų atkūrimas

4.5. Energijos suvartojimas

Saugus komunikavimo metodas yra kuriamas ir pritaikomas ribotų skaičiuojamųjų ir energijos išteklių įrenginių grupei. Numatyta, jog atliekant operacijas, robotas veiks iš išorinės baterijos, todėl

labai svarbu energiją taupyti visuose operacijos etapuose. Atliekamo tyrimo matavimo metodika aprašyta 3.3.2. skyrelyje.

Elektros energijos suvartojimas pradedamas matuoti nuo operacijos pradžios – judesio užfiksavimo. Iki judesio užfiksavimo robotas veikia budėjimo režime. Programai veikiant budėjimo režime, fiksuojant energijos matavimo prietaiso rodmenis, apskaičiuojamas energijos suvartojimas, naudojant galios formulę:

$$P = U \cdot I \cdot t = 5{,}18 \cdot 0{,}37 \cdot 1 = 1{,}92\ W \cdot s \qquad (2)$$

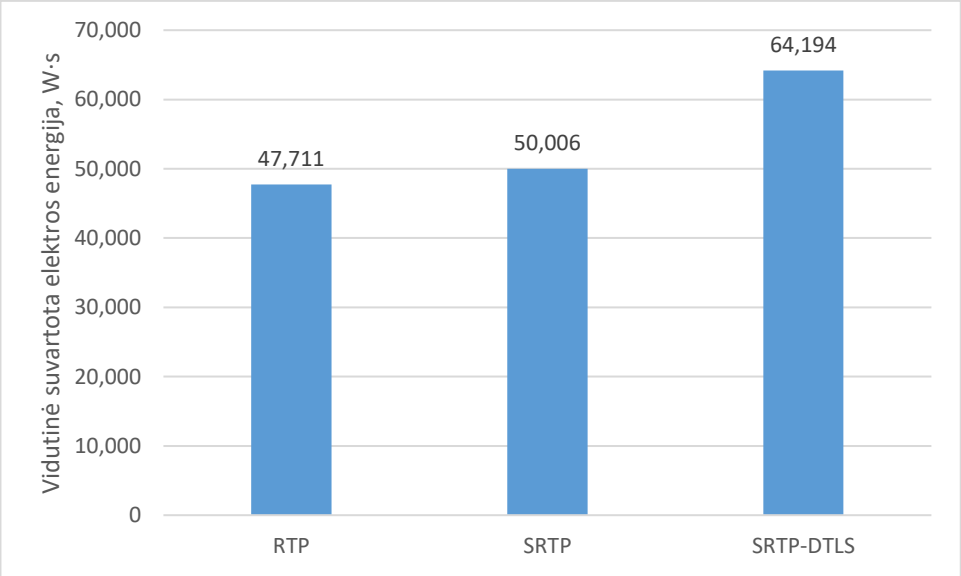
čia: P – galia, Ws; U – įtampa, V; I – srovė, A, t – laikas, s.

Kadangi nėra iš anksto aišku, kiek robotas veiks budėjimo režime iki užfiksuojant judesį, galios suvartojimas apskaičiuojamas priimant, jog laiko kintamasis lygus 1 s. Programai veikiant budėjimo režime sunaudojama tiek pat energijos, kiek įrenginiui veikiant ramybės būsenoje (nepaleidus programos).

Autonominio aplinkos stebėjimo roboto energijos suvartojimas, naudojant skirtingus protokolus ir jų rinkinius, buvo matuojamas nuo judesio užfiksavimo, iki transliacijos pabaigos. Sutarta, jog vaizdo transliavimas vyksta 15 sekundžių. Būtina pabrėžti, jog prie vaizdo transliacijos laiko prisideda ir laikas, kuris reikalingas sesijos inicijavimui, todėl bendra operacija, nuo pradžios iki galo vyksta šiek tiek daugiau nei 15 sekundžių. Vieno, iš daugelio bandymų, rezultatų pavyzdys pateiktas Priede Nr. 1. Matavimams naudojamas USB srovės ir įtampos matuoklis. Kiekvienu tiriamuoju atveju (RTP, SRTP, SRTP-DTLS), atlikta po 100 pilnų operacijų. Rezultatų suvestinė pateikiama 4.2 lentelėje.

4.2 lentelė. Vidutinė suvartota elektros energija operacijos metu

Protokolų architektūra	RTP	SRTP	SRTP-DTLS
Vidutinė suvartota elektros energija operacijos metu, W·s	47,711	50,006	64,194



4.18 pav. Vidutinė suvartota elektros energija operacijos metu

Modifikuoto SRTP protokolo naudojimas vaizdo transliacijoje, su iš anksto pasidalinto rakto technika, suteikia gana stiprų saugumo lygį, suvartodamas tik apie 5 % daugiau elektros energijos, nei atliekant identišką operaciją naudojant nesaugų RTP protokolą.

Hermann Hellwagneris ir kiti mokslininkai [81] ištyrė H.264/SVC formato vaizdo transliacijų, skirtingų šifravimo algoritmų savybes ir įtaką skaičiuojamiems pajėgumams. Tyrimo rezultatai parodė, jog SRTP vaizdo transliacijos metu procesoriaus apkrovimas buvo apie 26 %, tuo tarpu siunčiant duomenis neapsaugotu RTP, kompiuteris apkraunamas apie 13 %. Verta paminėti, jog šiame tyrime duomenų pobūdis SVC tipo, o patys duomenys buvo siučiami medijos priėmimo elementui (angl. *Media-aware network element*) ir tik po to galutiniam klientui. Tai sukelia papildomą apkrovimą transliavimo serveriui.

Hoseb M. Dermanilian ir Imad H. Elhajj [82] atliko tyrimą, kurio tikslas buvo nustatyti, kiek papildomų skaičiuojamųjų resursų ir laiko reikia, norint duomenis apsaugoti, naudojant SRTP protokolą ribotų išteklių įrenginiuose. Viename iš bandymų buvo skaičiuojami procesoriaus ciklai, atliekant SRTP operacijas. Atsižvelgiant, jog procesoriaus ciklų skaičius turi tiesioginę priklausomybę suvartojamai energijai, prieita prie išvados, jog SRTP suteikiamas gana stiprus saugumas, suvartoja stebėtinai mažai energijos. Mokslininkų tyrime, taip pat įrodyta, jog daugiau skaičiuojamųjų resursų reikalauja, būtent, autentifikavimas, o ne duomenų šifravimas, todėl duomenų autentifikavimas yra „brangesnis“ energijos suvartojimo prasme.

SRTP-DTLS naudojimo atveju, elektros energijos suvartojimas kur kas didesnis, nei transliuojant vaizdą nesaugiu RTP. Šis, apie 25 % skirtumas, gali būti paaiškintas tuo, jog šifravimui naudojamas SRTP prokotolas, kaip matyti iš bandymų, suvartoja tik apie 5 % daugiau elektros energijos nei RTP, o šiuo atveju, kartu naudojamas DTLS suvartoja didžiąją dalį energijos, atlikdamas sesijos inicijavimo algoritmus ir skaičiavimus. Šios DTLS procedūros, tam tikrą operacijos dalį, intensyviai naudoja skaičuojamuosius resursus, kas sukelia akivaizdų skirtumą energijos suvartojimo palyginime.

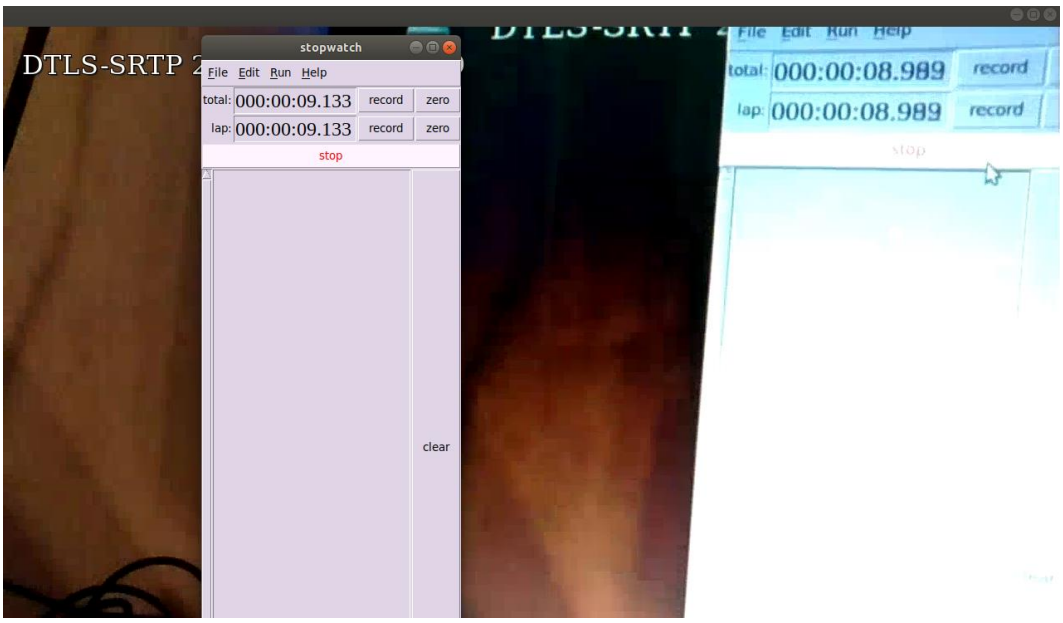
4.6. Tiesioginės transliacijos vėlavimas

Vienas iš sėkmingai įgyvendinto saugaus komunikavimo metodo kriterijų yra galutinis vaizdo vėlavimas, perduodant vaizdą realiu laiku. Siekiama, jog vėlavimas neviršytų 300 ms. Tokia užsibrėžta vėlavimo gairė praktiškai nėra matoma plika akimi ir vėlavimas vartotojui nejuntamas.

Matavimams naudojamas Ubuntu OS prieinamas „smartwatch“ virtualus chronometras. Vėlavimui išmatuoti, vartotojo įrenginyje įjungiamas chronometras, kuris yra filmuojamas roboto kamera. Tokiu būdu vartotojo ekrane matoma reali laiko žyma ir roboto filmuojama (transliuojama) laiko žyma. Operacijos metu, pakartotinai fiksuojant darbalaukio ekranvaizdes (angl. *Screenshot*), matomos dvi skirtingos laiko žymos. Šių laiko žymų skirtumas vadinamas galutiniu vaizdo vėlavimu.

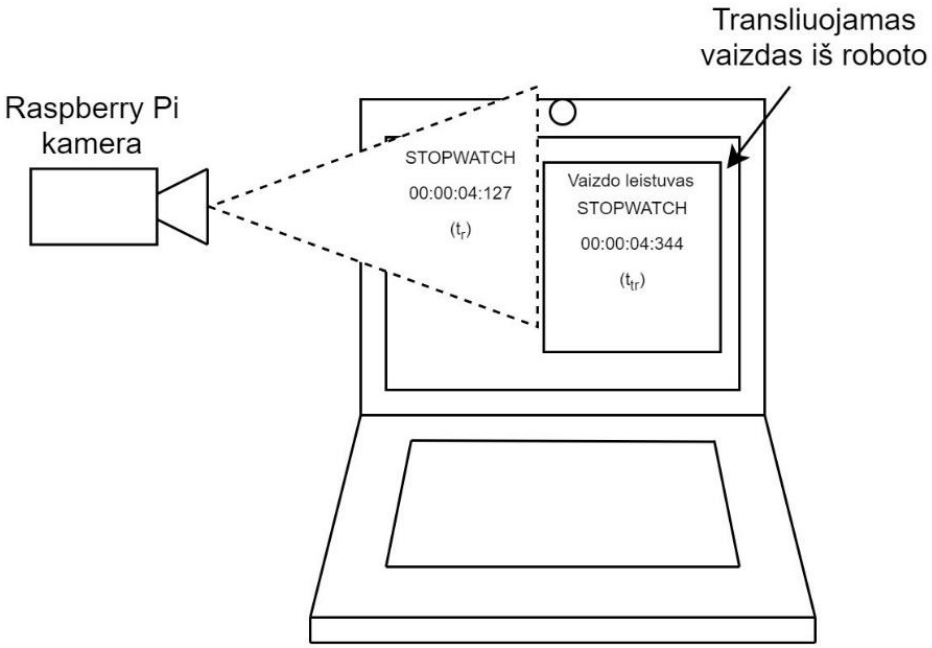
$$CDD = t_r - t_{tr}; \tag{3}$$

čia: *CDD* – galutinis vaizdo vėlavimas; t_r – reali laiko žyma; t_{tr} – transliuojama laiko žyma.



4.19 pav. Vėlavimo fiksavimo bandymas

Paveiksle 4.19 matoma, jog bandymo metu, vėlavimas yra 144 milisekundės. Matavimo bandymo paaiškinamoji schema pateikiama 4.20 paveiksle.

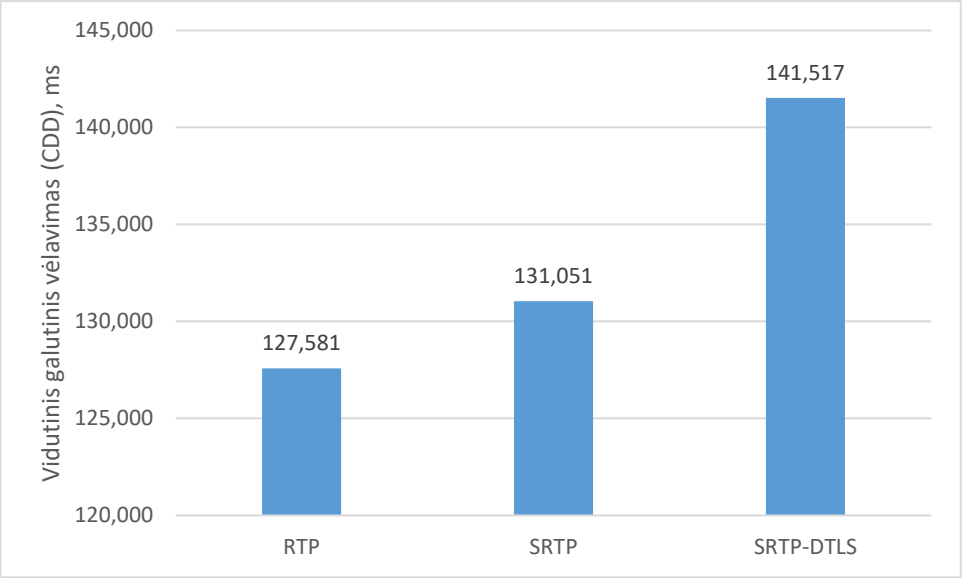


4.20 pav. Vėlavimo matavimo bandymo metodika

Tyrimo metu, vėlavimo bandymai, su visomis protokolų architektūroms atlikti po 100 kartų (15 sekundžių vaizdo transliavimas), vidutiniškai užfiksuojant apie 15 laiko reikšmių, iš kurių išskaičiuojama *CDD* reikšmė. Išvedamas bandymo vidutinio vėlavimo vidurkis, užfiksuojamos maksimalios ir minimalios transliacijos vėlavimo reikšmės. Rezultatai pateikiami 4.3 lentelėje.

4.3 lentelė. Vėlavimo bandymų rezultatų suvestinė

Protokolų architektūra	RTP	SRTP	SRTP-DTLS
Vidutinis galutinis vėlavimas (CDD), ms	127,581	131,051	141,517
Minimali galutinio vėlavimo reikšmė, ms	86,000	87,000	128,000
Maksimali galutinio vėlavimo reikšmė, ms	162,000	240,000	189,000



4.21 pav. Vidutinis galutinis vėlavimas (CDD)

Andre L. Alexanderio ir kitų mokslininkų [83] atliktas tyrimas parodė, jog VoIP programų vėlavimas, naudojant SRTP padidėja nežymiais 2 %, lyginant su RTP. Šį nežymų vėlavimą sukelia keli papildomi baitai, kurie atsiranda prie paketo pridedant autentifikacijos žymą. Žinoma, šis tyrimas orientuotas į garso perdavimą VoIP, tačiau panašūs rezultatai gaunami ir atlikus bandymą transliuojant vaizdą.

4.7. Protokolų greitaveika

Tyrimas užbaigiamas protokolų greitaveikos nustatymo bandymais. Siekiant išsiaiškinti protokolų ar jų rinkinių greitaveiką, atliekant roboto užduotį, fiksuojamas laikas, kada užfiksuotas judesys, ir laikas, kada atvyko pirmasis multimedijos duomenų paketas į kliento kompiuterį. Judesio užfiksavimo momentas fiksuojamas roboto įrenginyje, o pirmojo vaizdo duomenų paketo atvykimo laikas nustatomas kliento įrenginyje, naudojantis „WireShark“ programa.

„WireShark“ programoje pakeičiama standartinė konfigūracija nustatant, jog prie paketų būtų rodomas sistemos laikas, o ne laikas, nuo srauto fiksavimo pradžios. Roboto programoje nurodoma išspausdinti sistemos laiką, kai judesys užfiksuojamas ir transliacija pradedama. Naudojamas *date +%H:%M:%S:%N* laiko formatas. Bandymų metu, apskaičiuotas šių laiko žymių skirtumas leis nustatyti, kiek laiko protokolai trunka atlikdami visas reikiamas operacijas, sėkmingam komunikavimo užmezgimui.

$$G = t_{wr} - t_p; \quad (4)$$

čia: G – protokolų greitaveikos kintamasis; t_{wr} – pirmojo medijos paketo atvykimo laikas kliento įrenginyje; t_p – operacijos pradžios (judesio užfiksavimo) laikas.

Naudojant šią metodiką, norint gauti tikslius bandymų rezultatus, abiejuose įrenginiuose privalomas laiko sinchronizavimas. Kadangi įrenginiai veikia tame pačiame namų tinkle, sistemos laikų sinchronizavimui diegiamas ir konfigūruojamas NTP serveris. Pagal lokaciją pasirenkamos tikslesnės *lt.pool.ntp.org* ir *europe.pool.ntp.org* laiko baseinų zonos (angl. *Pools*). Įdiegus ir paleidus NTP serverį kliento įrenginyje, roboto mikrokompiuteryje nurodomas šio serverio IP adresas */etc/hosts/* faile. Konfigūracijos faile */etc/ntp.conf* nurodomas serverio pavadinimas *NTP-server-host* ir papildoma, jog standartiškai šis laiko sinchronizavimo serveris būtų naudojamas kaip prioritetinis.

Įdiegus NTP laiko sinchronizavimo priemones robote ir kliento įrenginnyje, komanda *ntpdate* patikrinama laiko paklaida tarp sistemų (žr. pav. 4.22). Sistemos laiko sinchronizavimo paklaida tarp įrenginių 0,000351 sekundės, tai yra, viso labo 0,351 milisekundės. Tokia paklaida leidžia daryti prielaidą, jog sistemų laikai yra sinchronizuoti.

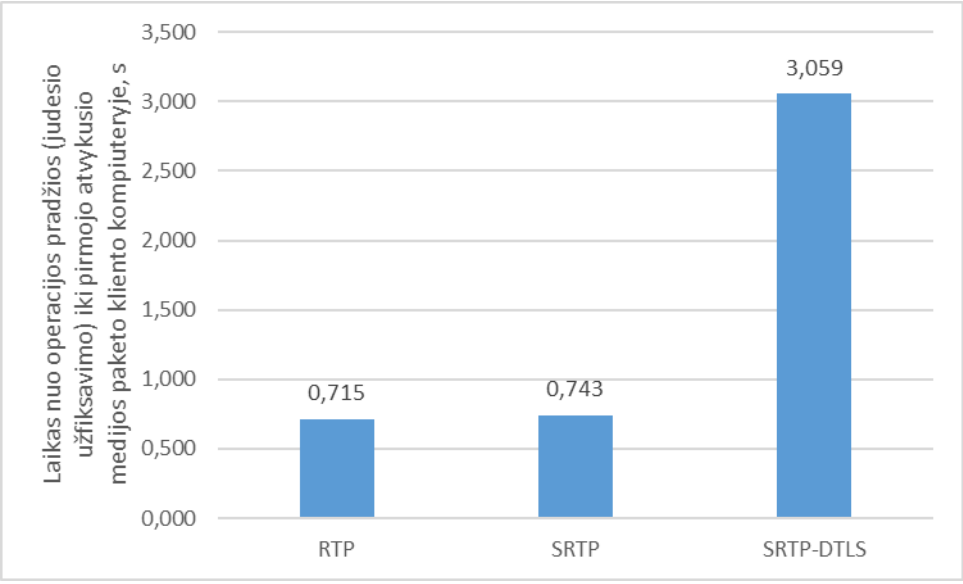
```
ubuntu@ubuntu:~$ sudo ntpdate NTP-server-host
 2 Apr 19:30:30 ntpdate[5282]: adjust time server 192.168.1.207 offset 0.000351
sec
```

4.22 pav. Sistemų laiko sinchronizavimo paklaida

Kiekvienu tyrimo atveju, atlikta po 100 vienodų bandymų. Gauti rezultatai pateikiami 4.4 lentelėje.

4.4 lentelė. Protokolų greitaveikos bandymų rezultatai

Protokolų architektūra	RTP	SRTP	SRTP-DTLS
Laikas nuo operacijos pradžios (judesio užfiksavimo) iki pirmojo atvykusio medijos paketo kliento kompiuteryje, s	0,715	0,743	3,059



4.23 pav. Protokolų greitaveikos palyginimas

Tiesiogiai transliuojant vaizdo duomenis, naudojant RTP ir modifikuotą SRTP protokolus, gaunami panašūs greitaveikos rezultatai (SRTP užtrunka apie 1,04 karto ilgiau). Tuo tarpu, SRTP-DTLS protokolų rinkinys, daugiau nei 4 kartais, lėčiau pasiruošia ir užmezga sesiją. Rezultatai leidžia daryti prielaidą, jog naudojant šį protokolų rinkinį, vartotojo įrenginį multimedijos paketai pasieks tik vidutiniškai po 3,059 sekundžių. Tai reiškia, jog vaizdas vartotojo įrenginyje vidutiniškai bus matomas tik po daugiau nei 3,059 sekundės, kadangi dar prisidės laikas, kurio metu atliekamas paketų išpakavimas, vaizdo duomenų dekodavimas (ši laiko konstanta priklauso nuo vartotojo įrenginio parametrų).

Sureshkumar V. Subramanian, Rudra Dutta [84] atliko panašų tyrimą, o jo rezultatai parodė, jog SIP sesijos inicijavimas duomenų perdavimui, naudojant saugų TLS kartu su SRTP, palyginus su nesaugiu RTP sesijos inicijavimu, užtrunka 30-40 % ilgiau. Pabrėžtina, jog šių mokslininkų tyrimas orientuotas į VoIP technologijos duomenų perdavimą. Atlikus tyrimą, nukreiptą į vaizdo duomenų perdavimą, matoma, jog inicijuojant DTLS sesiją iš ribotų pajėgumų įrenginio, užtrunkama 70-80% ilgiau nei duomenis perduodant nesaugiu RTP protokolu.

Verta paminėti, jog lyginant vidutines minimalias ir maksimalias šio bandymo reikšmes, matyti, jog naudojant:

- RTP protokolą, išlaikytas laiko reikšmių intervalas tarp [0,689; 0,763];
- SRTP protokolą, išlaikytas laiko reikšmių intervalas tarp [0,714; 0,775];
- SRTP-DTLS protokolus, išlaikytas laiko reikšmių intervalas tarp [1,004; 6,179].

Iš šių rezultatų daroma prielaida, jog naudojant DTLS poroje su SRTP, susijungimo tarp įrenginių laikas stipriai kinta, identiškos sesijos inicijavimo operacijos atliekamos per nevienodą laiką, o tai signalizuoja apie gana nestabilų veikimą.

4.8. Saugos komunikavimo metodo eksperimentinio tyrimo išvados

1. Eksperimentinio tyrimo metu, įrenginiai buvo paruošti darbui į juos įdiegus operacines sistemas, reikiamus vaizdo transliavimo karkasus, bibliotekas ir kitus komponentus (pagalbinės programos, įrankiai).
2. Praktiškai įgyvendinti multimedijos duomenų transliavimo metodai: naudojant nesaugų RTP, modifikuoto veikimo SRTP ir SRTP-DTLS protokolus. Apibrėžtos identiškos tyrimo sąlygos (vaizdo parametrai, bandymai tame pačiame tinkle, kodavimo algoritmai), siekiant gauti tikslesnius tyrimo rezultatus.
3. Įdiegta vaizdo priėmimo programa kliento kompiuteryje. Imituota duomenų perėmimo ataka, klausantis tinklo duomenų srauto „Wireshark“ programa. Tiriant duomenų srautą, įrodyta, jog SRTP ir SRTP-DTLS protokolų naudojimo atvejais, multimedijos duomenų išgauti nepavyksta, kadangi protokolų saugos mechanizmai užtikrina jų apsaugą ir autentifikavimą.
4. Ištirtas saugaus komunikavimo metodų elektros energijos suvartojimas, vykdant roboto operaciją. Suvartodamas apie 5 % daugiau energijos nei RTP, modifikuoto veikimo SRTP protokolas užtikrina pakankamą apsaugą: duomenų vientisumą, konfidencialumą ir autentifikavimą. Nors SRTP-DTLS protokolų rinkinys duomenų šifravimui naudoja SRTP protokolo šifravimo mechanizmus, operacijos metu suvartoja apie 25 % daugiau elektros energijos nei RTP. Didžiąją dalį energijos suvartoja sudėtingos DTLS kriptografinės ir sesijos inicijavimo operacijos.
5. Atliekant vaizdo transliavimo operaciją, fiksuojant realaus ir transliuojamo laiko žymas, ištirtas galutinis vaizdo vėlavimas kliento įrenginyje. Duomenų šifravimo ir autentifikavimo

mechanizmai žymaus transliacijos vėlavimo nesukelia. Naudojant SRTP-DTLS protokolų rinkinį, vėlavimas 1,1 karto didesnis nei multimedijos duomenis siunčiant nesaugiu RTP protokolu. Toks skirtumas nėra jaučiamas galutiniam vartotojui.

6. Protokolų greitaveikos tyrimas įrodė, jog DTLS naudojant SRTP pagrindinio rakto apsikeitimui ir stipraus lygio autentifikavimui užtikrinti, sukelia žymų transliacijos pradžios vėlavimą. Palyginimui, DTLS kriptografinės operacijos ir sesijos inicijavimas tarp įrenginių užtrunka vidutiniškai apie 3 sekundes, tuo tarpu, naudojant RTP ar SRTP, pirmasis medijos paketas pristatomas po maždaug 700 milisekundžių. Būtina paminėti, jog SRTP-DTLS protokolų rinkinio greitaveikos bandymų rezultatai stipriai kinta, intervale [1,004; 6,179], kas gali reikšti nestabilų veikimą.

Išvados

1. Išanalizavus gaminamų robotų rinką, nustatyta, jog didžioji dalis robotų yra su silpnais arba be kibernetinio saugumo mechanizmų. Apžvelgus grėsmes, rizikas ir atakų rūšis, nustatyta, jog didžioji dalis kibernetinių atakų prieš robotus įgyvendinamos nuotoliniu būdu, dėl nesaugaus ryšio kanalo pažeidžiamumų.
2. Nustatyta, jog stiprūs kriptografiniai saugos mechanizmai ir daugelis multimedijos duomenų protokolų, standartiškai, nėra tinkami ribotų išteklių įrenginiams. Būtinos protokolų ir saugumo mechanizmų modifikacijos, leidžiančios efektyviai išnaudoti įrenginio pajėgumus.
3. Pasiūlyti du saugos komunikavimo metodai. Saugiam aplinkos stebėjimui, numatytas modifikuoto veikimo SRTP, su iš anksto pasidalinto rakto technika. Metodo veikimas lyginamas su SRTP-DTLS protokolų poros ir nesaugaus RTP protokolo komunikavimo metodais.
4. Atsižvelgiant į autonominio aplinkos stebėjimo roboto atliekamą užduotį, siunčiamų duomenų tipą, apibrėžta operacijos sandara ir transliacijos pagrindiniai kriterijai (kokybė, saugumas, vėlavimas ir greitaveika).
5. Sprendimo įgyvendinimui pasirinktas „Raspberry PI 3B+“ mikrokompiuteris ir komponentai. Programinė komunikavimo metodo dalis atliekama, naudojant „Gsteamer“ atvirojo kodo multimedijos duomenų karkasą, parašytą C kalba. Bendra operacijos programa apjungiama naudojant Python programavimo ir BASH scenarijų kalbas.
6. Praktiškai įrodyta, jog modifikuoto veikimo SRTP ir SRTP-DTLS naudojimo atvejais, siunčiami duomenys yra šifruoti ir apsaugoti, o jų išgauti, tinklo srauto duomenų pasiklausymo atakos metu, nepavyko (priešingai nei RTP). Lyginant protokolų kiekybinius parametrus, matoma, jog modifikuotas SRTP, su iš anksto pasidalinto rakto technika, suvartodamas tik 5 % daugiau elektros energijos nei RTP, užtikrina duomenų vientisumą, konfidencialumą ir autentiškumą. Dėl sudėtingo sesijos inicijavimo, SRTP-DTLS protokolų poros veikimas suvartoja apie 25 % daugiau elektros energijos nei naudojant nesaugų RTP.
7. Atliktas galutinio vėlavimo tyrimas parodė, jog šifravimo ir autentifikavimo saugumo mechanizmai žymaus vėlavimo nesukelia. Naudojant SRTP ir RTP vėlavimo reikšmės praktiškai nesiskiria, o SRTP-DTLS sukeliamas vėlavimas vos 1,1 karto didesnis nei RTP. Protokolų greitaveikos tyrimas įrodė, jog DTLS naudojimas, SRTP pagrindinio rakto apsikeitimui saugiu kanalu, sukelia žymų pradžios transliacijos vėlavimą (vidutiniškai 3 sekundės). Naudojant SRTP-DTLS protokolų rinkinį, greitaveikos bandymų rezultatai stipriai kinta, kas gali reikšti nestabilų veikimą.
8. Komunikavimo metodo architektūrą vertėtų rinktis, priklausomai, nuo atliekamos užduoties, aplinkos, perduodamų duomenų pobūdžio. Robotui siunčiant duomenis kitam galiniam įrenginiui, modifikuotas SRTP, su iš anksto pasidalintais raktais, puikiai tinka, kadangi yra užtikrinamas efektyvus veikimas, duomenų šifravimas ir autentifikavimas. Ši metodo architektūra sunkiai pritaikoma, jeigu vaizdo duomenys transliuojami keliems įrenginiams (angl. *Multicast*), todėl ateityje numatoma ieškoti šio metodo patobulinimų arba pasiūlyti kitą saugos komunikavimo metodą, kuris tiktų daugiaabonentiniui transliavimui.

Literatūros sąrašas

1. RODRIGUEZ LERA, F.J. ir kt. Cybersecurity in Autonomous Systems: Evaluating the performance of hardening ROS. *Workshop on physical agents* [interaktyvus]. Malaga, 2016 [žiūrėta 2019-01-19]. Prieiga per: ResearchGate.
2. RODRIGUEZ LERA, F.J. ir kt. Cybersecurity of Robotics and Autonomous Systems: Privacy and Safety. *Robotics – Legal, Ethical and Socioeconomic Impacts*. 2017, pp. 76-88. Prieiga per: doi: http://dx.doi.org/10.5772/intechopen.69796
3. PRIYADARSHINI, I. Detecting and Mitigating Robotic Cyber Security Risks. *Cyber Security Risks in Robotics* [interaktyvus]. 2017 [žiūrėta 2019-02-20]. Prieiga per: ResearchGate.
4. CLARKE, R. Asimov's laws of robotics: Implications for information technology. 1994, vol. 27, no. 1. Prieiga per: doi: https://doi.org/10.1109/2.248881
5. ZLOTOWSKI, J., YOGEESWARA, K., BARTNECK, C. Can we control it? Autonomous robots threaten human identity, uniqueness, safety, and resources. *International Journal of Human-Computer Studies* [interaktyvus]. 2017, pp. 48-54 [žiūrėta 2019-02-20]. Prieiga per: ResearchGate.
6. CALO, M.R. Robots and Privacy. *ROBOT ETHICS: THE ETHICAL AND SOCIAL IMPLICATIONS OF ROBOTICS* [interaktyvus]. Kembridžas: MIT Press, 2010 [žiūrėta 2019-03-25]. Prieiga per: SSRN.
7. MILLER, K.L. A New Framework for Robot Privacy [interaktyvus]. 2017 [žiūrėta 2019-03-26]. Prieiga per: Semantic Scholar.
8. BEKEY, G.B. Autonomous robots – from biological inspiration to implementation and control. *Intelligent robotics and autonomous agents.* Kembridžas: MIT Press, 2006 vol. 24, pp. 271. Prieiga per: doi: https://doi.org/10.1017/s026357470622280x
9. LITMAN, T.A. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning [interaktyvus]. Viktorija, 2013 [žiūrėta 2019-03-29]. Prieiga per: Semantic Scholar.
10. Driverless bus, *CITE MAGAZINE* [interaktyvus]. 2017 [žiūrėta 2019-03-29]. Prieiga per: https://news.curtin.edu.au/cite/in-brief/driverless-bus/
11. BAGLOEE, S.A., TAVANA, M. ir ASADI, M. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. *Journal of Modern Transportation* [interaktyvus]. 2016, vol. 24, pp. 284-303. Prieiga per: Springer Link.
12. WYGLINSKI, A.M. ir kt. Security of Autonomous Systems Employing Embedded Computing and Sensors. *IEEE Micro* [interaktyvus]. 2013, vol. 33, pp. 80-86 [žiūrėta 2019-03-30]. Prieiga per: IEEE.
13. PETIT, J. ir SHLADOVER, S.E. Potential Cyberattacks on Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems* [interaktyvus]. 2014, vol. 16, pp. 546-556 [žiūrėta 2019-03-30]. Prieiga per: IEEE.
14. TAEIHAGH, A. ir LIM, H.S.M. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport Reviews. Long Term Implications of Automated Vehicles* [interaktyvus]. 2018, vol. 39, pp. 103-128 [žiūrėta 2019-03-30]. Prieiga per: Taylor and Francis Online.
15. MILLER, J., WILLIAMS, A.B. ir PEROULI, D. A Case Study on the Cybersecurity of Social Robots. *HRI'18 Companion* [interaktyvus]. Čikaga, 2018, pp. 195-196 [žiūrėta 2019-03-30]. Prieiga per: ACM DL.

16. MUCCHIANI, C. ir kt. Evaluating older adults' interaction with a mobile assistive robot. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* [interaktyvus]. Vankuveris, 2017 [žiūrėta 2019-04-01]. Prieiga per: IEEE.
17. CERRUDO, C. ir APA, L. Hacking Robots Before Skynet. *Cybersecurity Insight* [interaktyvus]. 2017 [žiūrėta 2019-04-02]. Prieiga per: IOActive.
18. DENNING, T. ir kt. A spotlight on security and privacy risks with future household robots: attacks and lessons. *UbiComp'09: Proceedings of the 11th international conference on Ubiquitous computing* [interaktyvus]. 2009, pp. 105-114 [žiūrėta 2019-04-03]. Prieiga per: ACM DL.
19. STUBBLEFIELD, A., IOANNIDIS, J. ir RUBIN, A.D. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). *ACM Transactions on Information and System Security* [interaktyvus]. 2004, pp. 319-332 [žiūrėta 2019-04-04]. Prieiga per: ResearchGate.
20. JOHNSON, A.M. ir AXINN, S. The Morality of Autonomous Robots. *Journal of Military Ethics* [interaktyvus]. 2013, vol. 12, pp. 129-141 [žiūrėta 2019-04-04]. Prieiga per: Taylor and Francis Online.
21. YOUSEF, K.M.A. ir kt. Analyzing Cyber-Physical Threats on Robotic Platforms. *Sensors* [interaktyvus]. 2018 [žiūrėta 2019-04-06]. Prieiga per: ResearchGate.
22. BOGUE, R. Robots for monitoring the environment. *Industrial Robot.* 2011, vol. 38, no. 6, pp. 560-566. Prieiga per: doi: https://doi.org/10.1108/01439911111179066
23. CAPRARI, G. ir kt. Highly compact robots for inspection of power plants. *2010 1st International Conference on Applied Robotics for the Power Industry* [interaktyvus]. Monrealis, 2010 [žiūrėta 2019-04-07]. Prieiga per: IEEE.
24. SONG, G. ir kt. A surveillance robot with hopping capabilities for home security. *IEEE Transactions on Consumer Electronics* [interaktyvus]. 2009, vol. 55, pp. 2034-2039 [žiūrėta 2019-04-07]. Prieiga per: IEEE.
25. ALUR, R. ir kt. Systems Computing Challenges in the Internet of Things. *Computing Community Consortium Catalyst* [interaktyvus]. 2015 [žiūrėta 2019-04-07]. Prieiga per: ResearchGate.
26. YONG, S. ir kt. Risk Mitigation Strategies for Mobile Wi-Fi Robot Toys from Online Pedophiles. *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing* [interaktyvus]. Bostonas, 2011 [žiūrėta 2019-04-15]. Prieiga per: IEEE.
27. SHIELDS, J. Smart Machines and Smarter Policy: Foreign Investment Regulation, National Security, and Technology Transfer in the Age of Artificial Intelligence. *SSRN Electronic Journal* [interaktyvus]. 2018 [žiūrėta 2019-04-15]. Prieiga per: ResearchGate.
28. LIN, P., BEKEY, G. ir ABNEY, K. Autonomous Military Robotics: Risk, Ethics, and Design. 2008, pp. 73-86. Prieiga per: doi: https://doi.org/10.21236/ada534697
29. JAVAID, A.Y. ir kt. Cyber security threat analysis and modeling of an unmanned aerial vehicle system. *2012 IEEE Conference on Technologies for Homeland Security (HST)* [interaktyvus]. Valthamas, 2013 [žiūrėta 2019-04-18]. Prieiga per: IEEE.
30. MORANTE, S., VICTORES, J.G. ir BALAGUER, C. Cryptobotics: Why Robots Need Cyber Safety. *Frontiers in Robotics and AI* [interaktyvus]. Madridas, 2015, vol. 2 [žiūrėta 2019-04-19]. Prieiga per: Semantic Scholar.
31. ABOMHARA, M., KOIEN, G.M. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. *Journal of Cyber Security* [interaktyvus]. 2015, vol. 4, pp. 65-88 [žiūrėta 2019-04-20]. Prieiga per: Semantic Scholar.

32. JENNEHAG, U., FORSSTROM, S. ir FIORDIGIGLI, F.V. Low Delay Video Streaming on the Internet of Things Using Raspberry Pi. *Electronics* [interaktyvus]. 2016 [žiūrėta 2019-04-20]. Prieiga per: Semantic Scholar.
33. PEREIRA, R. ir PEREIRA, E.G. Video Streaming Considerations for Internet of Things. *2014 International Conference on Future Internet of Things and Cloud* [interaktyvus]. Barselona, 2014 [žiūrėta 2019-04-25]. Prieiga per: IEEE.
34. SOMA, S. ir PATIL, A. Novel Architecture for IoT Based Video Streaming over Cloud. *International Journal of Emerging Technology in Computer Science and Electronics (IJETCSE)* [interaktyvus]. 2016, vol. 23, pp. 41-48 [žiūrėta 2019-04-28]. ISSN: 0976-1353. Prieiga per: IJETCSE.
35. WU, D. ir kt. Streaming Video over the Internet: Approaches and Directions. *IEEE Transactions on Circuits and Systems for Video Technology* [interaktyvus]. 2001, vol. 11, pp. 282-300 [žiūrėta 2019-04-29]. Prieiga per: IEEE.
36. LIU, P.X. ir kt. An UDP-based protocol for Internet robots. *Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No. 02EX527)* [interaktyvus]. Šanchajus, 2002 [žiūrėta 2019-05-01]. Prieiga per: IEEE.
37. LIU, P.X., MENG, M.Q.H. ir YANG, S.X. Data Communications for Internet Robots. *Autonomous Robots* [interaktyvus]. 2003, pp. 213-223 [žiūrėta 2019-05-04]. Prieiga per: ResearchGate.
38. SCHULZRINNE, H. ir kt. RTP: A Transport Protocol for Real-Time Applications. *RFC1889.* 1996. Prieiga per: doi: https://doi.org/10.17487/RFC1889
39. LIU, C. Multimedia Over IP: RSVP, RTP, RTCP,RTSP [interaktyvus]. 1999 [žiūrėta 2019-05-04]. Prieiga per: Semantic Scholar.
40. DURRESI, A. ir JAIN. R. RTP, RTCP, and RTSP - Internet protocols for Real-Time Multimedia Communication. *The Industrial Information Technology Handbook* [interaktyvus]. 2005 [žiūrėta 2019-05-20]. Prieiga per: https://www.cse.wustl.edu/~jain/books/ftp/rtp.pdf
41. BAUGHER, M. ir kt. The Secure Real-time Transport Protocol (SRTP). *RFC3711* [interaktyvus]. 2004 [žiūrėta 2019-10-02]. Prieiga per: https://tools.ietf.org/html/rfc3711
42. PERKINS, C. ir WESTERLUND, M. Securing the RTP framework: why RTP does not mandate a single media security solution. *RFC7202* [interaktyvus]. 2014 [žiūrėta 2019-10-02]. ISSN: 2070-1721. Prieiga per: Semantic Scholar.
43. SCHULZRINNE, H. ir kt. Real Time Streaming Protocol (RTSP). *RFC2326* [interaktyvus]. 1998 [žiūrėta 2019-10-02]. Prieiga per: https://tools.ietf.org/html/rfc2326
44. ALOMAN, A. ir kt. Performance evaluation of video streaming using MPEG DASH, RTSP, and RTMP in mobile networks. *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)* [interaktyvus]. Miunchenas, 2015 [žiūrėta 2019-10-02]. Prieiga per: IEEE.
45. MARTINEZ-JULIA, P. ir kt. Evaluating Video Streaming in Network Architectures for the Internet of Things. *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* [interaktyvus]. Taičungas, 2013 [žiūrėta 2019-10-10]. Prieiga per: IEEE.
46. YANG, G.J., CHOI, B.W. ir KIM, J.H. Implementation of HTTP Live Streaming for an IP Camera using an Open Source Multimedia Converter. *International Journal of Software Engineering and Its Applications* [interaktyvus]. 2014, vol. 8, no. 6, pp. 39-50 [žiūrėta 2019-10-10]. Prieiga per: ResearchGate.

47. THANG, T.C. ir kt. Adaptive Streaming of Audiovisual Content using MPEG DASH. *IEEE Transactions on Consumer Electronics* [interaktyvus]. 2012, vol. 58, pp. 78-85 [žiūrėta 2019-10-12]. Prieiga per: IEEE.
48. PEREIRA, R. ir PEREIRA, E. Video Streaming: H.264 and the Internet of Things. *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops* [interaktyvus]. 2015, pp. 711-714 [žiūrėta 2019-10-13]. Prieiga per: IEE.
49. LI, B. ir kt. Two Decades of Internet Video Streaming: A Retrospective View. *ACM Transactions on Multimedia Computing, Communications and Applications* [interaktyvus]. 2013, vol. 9 [žiūrėta 2019-10-15]. Prieiga per: ResearchGate.
50. THANG, T.C. ir kt. An Evaluation of Bitrate Adaptation Methods for HTTP Live Streaming. *IEEE Journal on Selected Areas in Communications* [interaktyvus]. 2014, vol. 32, pp. 693-705 [žiūrėta 2019-10-15]. Prieiga per: IEEE.
51. OZFATURA, E., ERCETIN, O. ir INALTEKIN, H. Optimal Network-Assisted Multiuser DASH Video Streaming. *IEEE Transactions on Broadcasting.* 2018, vol. 64, pp. 247-265. Prieiga per: doi: https://doi.org/10.1109/TBC.2018.2823644
52. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC 23009-4:2018. *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 4: Segment encryption and authentication.* Tarptautinis standartizacijos departamentas, 2018.
53. SUN, H., VETRO, A. ir XIN, J. An overview of scalable video streaming. *Wireless Communications and Mobile Computing*. 2007, vol. 7, pp. 159-172. Prieiga per: doi: https://doi.org/10.1002/wcm.v7:2
54. GONZALEZ, I.S. ir kt. Implementation and Analysis of Real-Time Streaming Protocols. *Sensors* [interaktyvus]. 2017, vol. 17 [žiūrėta 2019-10-25]. Prieiga per: ResearchGate.
55. SODAGAR, I. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia* [interaktyvus]. 2011, vol. 18, pp. 62-67 [žiūrėta 2019-10-25]. Prieiga per: IEEE.
56. BEGEN, A., AKGUL, T. ir BAUGHER, M. Watching Video over the Web. Part I: Streaming Protocols. *IEEE Internet Computing* [interaktyvus]. 2010, vol. 15, pp. 54-63 [žiūrėta 2019-10-26]. Prieiga per: IEEE.
57. ALCHARCHAFCHI, M., QUTQUT, M.H. ir ALMASALHA, F. Video Security in Internet of Things: An Overview. *International Journal of Computer Network and Information Security* [interaktyvus]. 2017, vol. 17 [žiūrėta 2019-10-25]. Priega per: ResearchGate.
58. KOTHMAYR, T. ir kt. A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication. *37th Annual IEEE Conference on Local Computer Networks – Workshops* [interaktyvus]. 2012 [žiūrėta 2019-10-28]. Priega per: IEEE.
59. NGUYEN, T.K., LAURENT, M. ir OUALHA, N. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks* [interaktyvus]. 2015, vol. 32, pp. 17-31 [žiūrėta 2019-11-03]. Prieiga per: ScienceDirect.
60. HUMMEN, R. ir kt. Towards viable certificate-based authentication for the internet of things. *HotWiSec '13: Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy* [interaktyvus]. 2013, pp. 37-42 [žiūrėta 2019-11-03]. Prieiga per: ACM DL.
61. RAY, S. ir BISWAS, P. Establishment of ECC-based Initial Secrecy Usable for IKE Implementation. *Proceedings of the World Congress on Engineering* [interaktyvus]. Londonas, 2012, vol. 1 [žiūrėta 2019-11-05]. Prieiga per: Semantic Scholar.

62. SONG, D. ir WEN, F. A Secure Authentication and Key Agreement Scheme in Smart Home. *Journal of Physics Conference Series* [interaktyvus]. 2019 [žiūrėta 2019-11-10]. Prieiga per: ResearchGate.
63. GALLENMULLER, S. ir kt. DTLS Performance - How Expensive is Security? *ArXiv* [interaktyvus]. 2019 [žiūrėta 2019-11-15]. Prieiga per: Semantic Scholar.
64. KAUP, F., GOTTSCHLING, P. ir HAUSHEER, D. PowerPi: Measuring and Modeling the Power Consumption of the Raspberry Pi. *39th Annual IEEE Conference on Local Computer Networks* [interaktyvus]. Edmontonas, 2014 [žiūrėta 2019-11-15]. Prieiga per: IEEE.
65. ELKILANI, W.S. ir ABDUL-KADER, H.M. Performance of Encryption Techniques for Real Time Video Streaming. *2009 International Conference on Networking and Media Convergence* [interaktyvus]. Kairas, 2009 [žiūrėta 2019-11-16]. Prieiga per: IEEE.
66. MATTSSON, U.T. Key Management for Enterprise Data Encryption. *SSRN Electronic Journal* [interaktyvus]. 2007 [žiūrėta 2019-11-16]. Prieiga per: ResearchGate.
67. KRAWCZYK, H., BELLARE, M. ir CANETTI, R. HMAC: Keyed-Hashing for Message Authentication. *RFC2104* [interaktyvus]. 1997 [žiūrėta 2019-11-16]. Prieiga per: https://tools.ietf.org/html/rfc2104
68. *BLUEKRYPT: Cryptographic Key Length Recommendation* [interaktyvus]. 2020 [žiūrėta 2019-11-18]. Prieiga per: https://www.keylength.com/en/4/
69. KIRSTEIN, P.T., BROWN, I. ir WHELAN, E. Secure multicast conferencing. *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00* [interaktyvus]. 2000 [žiūrėta 2019-11-18]. Prieiga per: IEEE.
70. LINLIN, Z. ir kt. The Implementation of An Secure RTP Transmission Method Based on DTLS. *2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control* [interaktyvus]. 2013 [žiūrėta 2019-11-20]. Prieiga per: IEEE.
71. *DISTRELEC: RASPBERRY PI 3B+ - Raspberry Pi 3 - Model B+ 1GB RAM* [interaktyvus]. 2020 [žiūrėta 2020-02-10]. Prieiga per: https://www.distrelec.lt/lt/raspberry-pi-model-1gb-ram-raspberry-pi-raspberry-pi-3b/p/30109158
72. *ALIBABA: New Green Stable HC-SR501 human body sensor pir motion sensor module Infrared Sensor* [interaktyvus]. 2020 [žiūrėta 2020-02-10]. Prieiga per: https://www.alibaba.com/product-detail/New-Green-Stable-HC-SR501-human_60730533276.html
73. *ALIEXPRESS: Raspberry Pi Zero Camera 5MP Webcam RPI Zero Camera Module for Raspberry Pi Zero* [interaktyvus]. 2020 [žiūrėta 2020-02-10]. Prieiga per: https://www.aliexpress.com/item/32788881215.html?spm=a2b0s.9042311.0.0.43fa4c4dGxmJnW
74. *AMAZON: KCX KCX-017 Power Bank Capacity Tester* [interaktyvus]. 2020 [žiūrėta 2020-02-10]. Prieiga per: https://www.amazon.com/KCX-017-Power-Capacity-Tester-White/dp/B00WBHQJ8W
75. WEI, C. ir kt. Capture-to-display delay measurement system for visual communication applications. *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* [interaktyvus]. 2013 [žiūrėta 2020-03-09]. Prieiga per: IEEE.
76. QUANG, N., GUNTUR, R. ir OOI, W.T. Towards Understanding User Tolerance to Network Latency in Zoomable Video Streaming. *Conference: Proceedings of the 19th International Conference on Multimedea* [interaktyvus]. Skotsdeilis, 2011 [žiūrėta 2020-03-09]. Prieiga per: ResearchGate.

77. *GSTREAMER: Open Source Multimedia Framework* [interaktyvus]. 2020 [žiūrėta 2020-03-10]. Prieiga per: https://gstreamer.freedesktop.org/
78. *UBUNTU: Install Ubuntu Server on a Raspberry Pi 2, 3 or 4* [interaktyvus]. 2020 [žiūrėta 2020-03-10]. Prieiga per: https://ubuntu.com/download/raspberry-pi
79. KAMACI, N. ir ALTUNBASAK, Y. Performance comparison of the emerging H.264 video coding standard with the existing standards. *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)* [interaktyvus]. Baltimorė, 2003 [žiūrėta 2020-03-29]. Prieiga per: IEEE.
80. CHEN, L. ir LEE, C. Multi-level Secure Video Streaming Over SRTP. *ACM-SE 43* [interaktyvus]. 2005 [žiūrėta 2020-03-29]. Prieiga per: Semantic Scholar.
81. HELLWAGNER, H. ir kt. Efficient in-network adaptation of encrypted H.264/SVC content. *Image Communication.* 2009. Prieiga per: doi: https://doi.org/10.1016/j.image.2009.07.002
82. DERMANILIAN, H.M. ir ELHAJJ, I.H. Evaluating Secure Real-Time Transport Protocol Performance on Power Constrained Handheld Devices. *2010 International Conference on Energy Aware Computing* [interaktyvus]. Kairas, 2010 [žiūrėta 2020-03-29]. Prieiga per: IEEE.
83. ALEXANDER, A.L., WIJESINHA, A.L. ir KARNE, R.K. An Evaluation of Secure Real-time Transport Protocol (SRTP) Performance for VOIP. *Conference: Third International Conference on Network and System Security* [interaktyvus]. 2009 [žiūrėta 2020-03-29]. Prieiga per: ResearchGate.
84. SUBRAMANIAN, S.V. ir DUTTA, R. Comparative Study of Secure vs Non-Secure Transport Protocols on the SIP Proxy Server Performance: An Experimental Approach. *2010 International Conference on Advances in Recent Technologies in Communication and Computing* [interaktyvus]. Kotajamas, 2010 [žiūrėta 2020-03-29]. Prieiga per: IEEE.

Priedai

1 priedas. Energijos suvartojimo matavimas

P1 lentelė. DTLS-SRTP energijos suvartojimo bandymo rezultatai

Pradžia, mm:ss,000	**Pabaiga, mm:ss,000**	**Laiko intervalas, ss,000**	**Srovė, A**	**Itampa, V**	**Galia per laiką, W·s**
00:00,000	00:00,653	0,653	0,37	5,26	1,271
00:00,653	00:01,184	0,531	0,58	5,26	1,620
00:01,184	00:02,954	1,770	0,82	5,26	7,634
00:02,954	00:03,620	0,666	0,62	5,26	2,172
00:03,620	00:05,763	2,143	0,87	5,26	9,807
00:05,763	00:06,783	1,020	0,83	5,26	4,453
00:06,783	00:08,365	1,582	0,84	5,26	6,990
00:08,365	00:09,032	0,667	0,63	5,26	2,210
00:09,032	00:09,874	0,842	0,62	5,26	2,746
00:09,874	00:10,756	0,882	0,64	5,26	2,969
00:10,756	00:11,815	1,059	0,81	5,26	4,512
00:11,815	00:13,114	1,299	0,62	5,26	4,236
00:13,114	00:13,900	0,786	0,60	5,26	2,481
00:13,900	00:15,015	1,115	0,62	5,26	3,636
00:15,015	00:16,202	1,187	0,63	5,26	3,933
00:16,202	00:17,554	1,352	0,62	5,26	4,409
00:17,554	00:19,258	1,704	0,86	5,26	7,708
				VISO:	**72,787**