KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Deividas Pranevičius

DAIKTŲ INTERNETO ĮRENGINIŲ AUTENTIFIKACIJA PANAUDOJANT BLOKŲ GRANDINĖS TECHNOLOGIJĄ

Baigiamasis magistro darbas

Vadovas
prof. dr. Algimantas Venčkauskas

KAUNAS, 2019

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

DAIKTŲ INTERNETO ĮRENGINIŲ AUTENTIFIKACIJA PANAUDOJANT BLOKŲ GRANDINĖS TECHNOLOGIJĄ

Baigiamasis magistro darbas
Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
(parašas) prof. dr. Algimantas Venčkauskas
(data)
Recenzentas
(parašas) doc. dr. Rasa Brūzgienė
(data)

Projektą atliko
(parašas) Deividas Pranevičius
(data)

KAUNAS, 2019

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos

(Fakultetas)

Deividas Pranevičius

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga, 621E10003

(Studijų programos pavadinimas, kodas)

„Daiktų interneto įrenginių autentifikacija panaudojant blokų grandinės technologiją“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 ___ m. _____ ____ d.

Kaunas

Patvirtinu, kad mano, **Deivido Pranevičiaus,** baigiamasis projektas tema „Daiktų interneto įrenginių autentifikacija panaudojant blokų grandinės technologiją“ yra parašytas visiškai savarankiškai, o 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)*

Pranevičius Deividas. „Daiktų interneto įrenginių autentifikacija panaudojant blokų grandinės technologiją“. Magistro baigiamasis projektas / vadovas prof. dr. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatikos fakultetas, Kompiuterių katedra.

Reikšminiai žodžiai: daiktų internetas, blokų grandinė, autentifikacija, ūko kompiuterija

Kaunas, 2019. 69 p.

SANTRAUKA

Įsibėgėjant ketvirtai pramonės revoliucijai ir technologijoms tampant vis labiau prieinamoms, prie interneto besijungiančių įrenginių kiekis eksponentiškai didėja. Tai kelia ekspertų susirūpinimą, kadangi tradicinės daiktų interneto architektūros nėra pritaikytos tokiam didžiuliam kiekiui dalyvių ir duomenų. Vienas iš būdų spręsti šią problemą yra daiktų interneto decentralizavimas. Šiame tiriamajame darbe analizuojama kaip blokų grandinės technologija gali padėti sprendžiant minėtas problemas, sukuriant decentralizuotą ir saugų sprendimą. Darbe suprojektuota architektūra remiasi ūko kompiuterija, kur dalis debesijos paslaugų krūvio perkeliama į tarpinį (ūko) sluoksnį, arčiau galinių įrenginių. Tiriamajame darbe suprojektuotoje architektūroje ūko sluoksnis naudojamas daiktų interneto įrenginių autentifikacijai blokų grandinėje atlikti. Remiantis suprojektuota architektūra sukurtas prototipas, atlikta jo kiekybinė ir kokybinė analizė.

Pranevičius Deividas. Blockchain-Based Device Authentication for the Internet of Things. Master‘s thesis in Information and Information Technology Security / supervisor prof. dr. Algimantas Venčkauskas. The Faculty of Informatics, Kaunas University of Technology.

Key words: internet of things, blockchain, authentication, fog computing

Kaunas, 2019. 69 p.

SUMMARY

As the fourth industrial revolution and technology become more and more accessible, the number of devices connected to the Internet is exponentially increasing. This is a concern for the experts, as the traditional Internet of Things architectures are not adapted to such many participants and data. One way to address this problem is to decentralize the Internet of Things. This research analyzes how blockchain technology can help solve these problems by creating a decentralized and secure solution. The architecture designed in the work is based on fog computing, where a part of the cloud service load is transferred to an intermediate (fog) layer close to the end devices. In the architecture designed in the research work, the ow layer is used to authenticate objects in the blockchain. Based on the designed architecture, a prototype was created, and its quantitative and qualitative analysis was performed.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Mazgas – blokų grandinių kontekste apibūdinamas blokų grandinės dalyvis.

Lygiarangis mazgas – blokų grandinių kontekste apibūdinamas kaip specializuotas blokų grandinės dalyvis, paprastai turintis blokų grandinės kopiją.

Butelio kaklelio efektas – programų inžinerijoje butelio kaklelio efektas reiškia situaciją, kai programos ar kompiuterinės sistemos sparta yra smarkiai ribojama vienintelio komponento.

HTTP – programinis protokolas skirtas informacinėms sistemoms.

M2M – tiesioginė komunikacija tarp įrenginių naudojant tam tikrą komunikacijos kanalą.

TCP – vienas pagrindinių interneto protokolų, suteikiantis patikimą, eiliškumą ir klaidų tikrinimą užtikrinantį susisiekimą tarp programų.

UDP – TCP analogas, pritaikytas programos, kurioms nereikia patikimo duomenų perdavimo.

Šliuzas – tai tinklo dalis, leidžianti duomenims keliauti iš vieno tinklo į kitą.

SSL ir TLS – tai kriptografiniai protokolai, suteikiantys komunikacijos saugą kompiuterių tinkle. SSL yra senesnė TLS versija, kurios šiais laikais stengiamasi nebenaudoti.

DTLS – tai komunikacijos protokolas, suteikiantis komunikacijos saugą duomenų paketais paremtoms programoms.

URI – serija simbolių, kurie vienareikšmiškai nusako tam tikrą resursą.

3GPP – standartų organizacija, plėtojanti mobiliosios telefonijos protokolus.

DoS ataka – kibernetinė ataka, kurios metu siekiama sutrikdyti kompiuterinio įrenginio ar tinklo resursų darbą taip, kad naudotojai negalėtų jų pasiekti.

QoS – apibūdinimas arba matavimas, skirtas el. paslaugos kokybei nustatyti, kaip pvz. telefonija, kompiuterių tinklas arba debesijos kompiuterijos paslauga.

WiMAX – bevielio plačiajuosčio ryšio standartų šeima.

LTE – bevielio plačiajuosčio ryšio standartas mobiliesiems įrenginiams ir duomenų terminalams.

EUTRAN – yra 3GPP LTE judriojo ryšio tinklų atnaujinimo kelio oro sąsaja.

4G – ketvirtoji mobiliojo ryšio karta.

EPC - 4G LTE tinkle teikiama konvergencinio balso ir duomenų sistema.

Bitkoinas – kriptovaliuta, elektroninių pinigų forma.

LoRaWAN – bevielio ryšio standartas, leidžiantis daiktų interneto įrenginiams bendrauti dideliu atstumu, naudojant minimalų baterijos naudojimą.

Negalėjimas išsižadėti – situacija, kai pareiškimo autorius negali sėkmingai ginčyti jo autorystės ar susijusios sutarties galiojimo.

RAM – operatyvioji kompiuterinio įrenginio atmintis.

VCPU – fizinis centrinis procesorius (CPU), priskirtas virtualiai mašinai (VM).

SQL – konkrečiai sričiai skirta kalba, naudojama programuojant ir skirta valdyti duomenis, saugomus reliacinės duomenų bazės valdymo sistemoje (RDBMS), arba srauto apdorojimui reliacinio duomenų srauto valdymo sistemoje (RDSMS).

NoSQL – duomenų saugojimo ir paieškos mechanizmas, kuris yra modeliuojamas kitomis priemonėmis nei reliacinėse duomenų bazėse naudojami ryšiai.

ĮVADAS

Daiktų internetas – tai koncepcija, kurioje visi įrenginiai, turintys tam tikrą prieigą prie interneto (arba prieigą vienas prie kito), gali susijungti į bendrą visumą [1]. Remiantis analitikos firmos „Gartnet“ duomenimis, 2020-aisiais metais turėtų būti daugiau nei 26 milijardai prisijungusių įrenginių. Kartu su gausybe galimybių, kurias daiktų internetas žada, jis taip pat pristato ir gausybę keblumų. Esant tokiam kiekiui įrenginių, kurie gali bet kuriuo metu susidurti su klaidomis, kritiniu veiksniu tampa įrenginių autentifikacija ir autorizacija [2]. Įrenginiai privalo identifikuotis prieš suteikiant jiems prieigą prie paslaugų ir programėlių. Tačiau didelė dalis daiktų interneto įrenginių autentifikacijos metu pasirodo labai prastai, naudodami silpnus slaptažodžius arba naudodami nepakeistus pradinius slaptažodžius.

Darbo problematika ir aktualumas

Šiame darbe tiriama problematika yra standartizuoto ir saugaus autentifikacijos mechanizmo tarp daiktų interneto įrenginių trūkumas. Vienas iš galimų sprendimų yra blokų grandinių (angl. blockchain) technologijos panaudojimas.

Darbo tikslas ir uždaviniai

Baigiamojo darbo tikslas yra ištirti blokų grandinės technologijos taikymo galimybes daiktų interneto įrenginių autentifikavimui ir pasiūlyti sprendimą. Tikslui įgyvendinti sudaryti tokie uždaviniai:

1) susipažinti su daiktų interneto architektūra;
2) atlikti daiktų internete naudojamų įrenginių autentifikacijos metodų analizę;
3) atlikti kibernetinių atakų prieš daiktų interneto įrenginius analizę;
4) susipažinti su blokų grandinės technologija ir jos taikymu daiktų interneto komunikacijoje;
5) išanalizuoti ir palyginti blokų grandinių kūrimo programinę įrangą;
6) pasiūlyti daiktų interneto įrenginių autentifikavimo metodą, panaudojantį blokų grandinės technologiją;
7) realizuoti pasiūlytą metodą ir palyginti jo rezultatus.

Darbo rezultatai ir jų svarba

Kaip darbo rezultatas numatomas praktiškai realizuotas autentifikavimo metodas bei atliekami testai pažeidžiamumams aptikti. Gauti rezultatai palyginami su kitais daiktų interneto įrenginių autentifikavimo metodais, atliekamos kokybinės ir kiekybinės charakteristikos. Rezultatai turi didžiulę svarbą tolimesnei blokų grandinių technologijos plėtrai daiktų interneto įrenginių saugumui užtikrinti.

1. DAIKTŲ INTERNETO ĮRENGINIŲ AUTENTIFIKACIJOS PROBLEMINĖS SRITIES ANALIZĖ

Daiktų interneto amžius jau įsibėgėjo ir didžiosios kompanijos rungiasi tarpusavyje kas taps lydere [3]. Tačiau su kiekvienu nauju įrenginiu padidėja susirūpinimas dėl privatumo ir apsaugos. Šis susirūpinimas aprėpia viską – nuo įsilaužėlių, bandančių pavogti duomenis, iki pasikėsinimų į gyvybę.

Fundamentali daiktų interneto saugos silpnybė yra ta, kad jis padidina įrenginių, esančių už tinklo užkardos, kiekį. Prieš dešimtį metų dauguma žmonių turėjo saugoti savo kompiuterius. Prieš penkerius metus teko susirūpinti ir išmaniaisiais telefonais. Dabar tenka rūpintis sauga įvairiausiuose objektuose, kaip mašina, namų apyvokos priemonės, dėvimi įrenginiai ir daugybė kitų daiktų interneto įrenginių. Taigi, akivaizdu, kad reikalingi nauji metodai, užtikrinantys daiktų interneto saugumą ir stabilumą.

1.1. Analizės tikslas

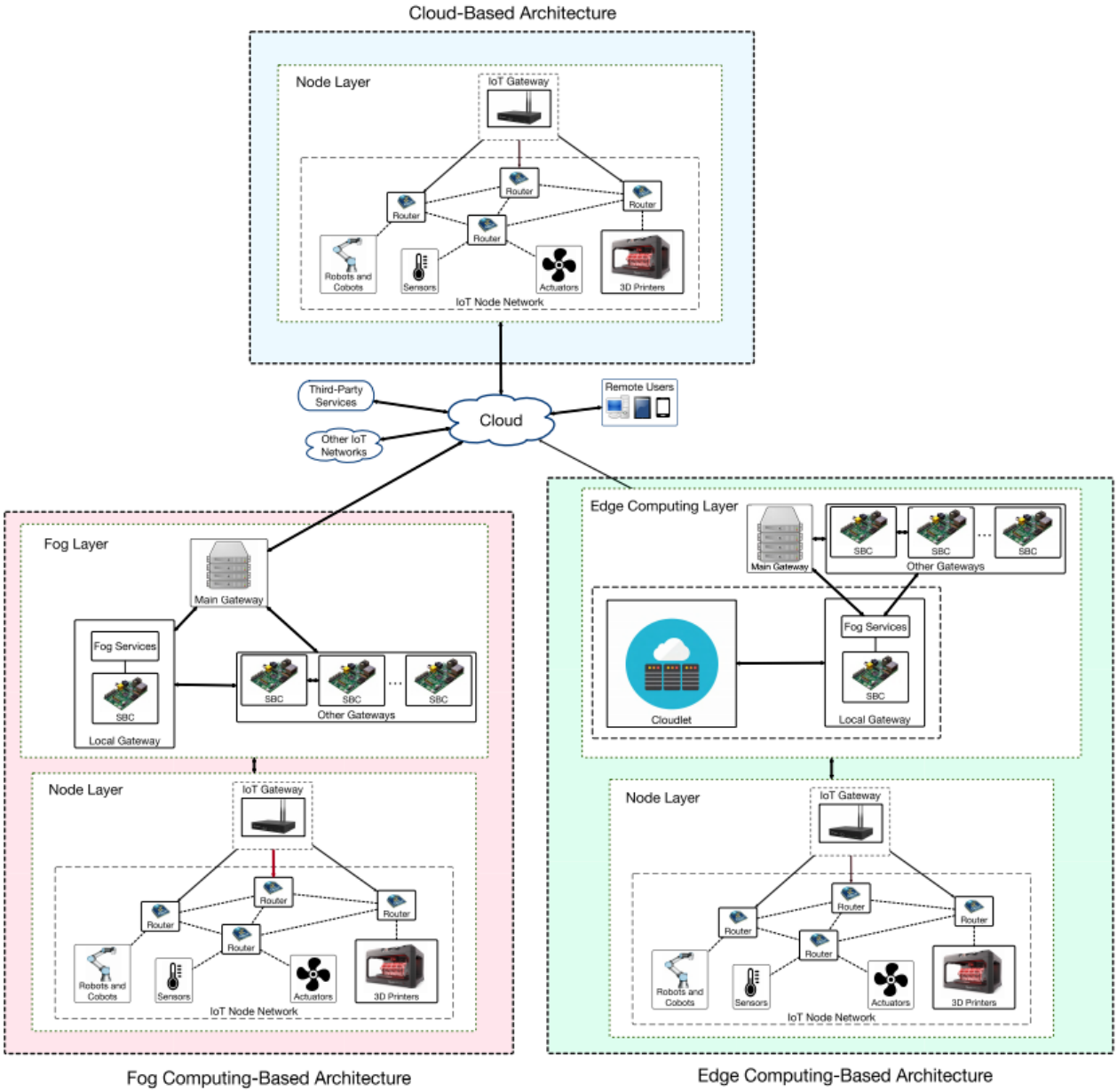
Šiame darbe dėmesys sutelkiamas į vieną iš pagrindinių ir diskutuotinai svarbiausią daiktų interneto saugumo dalį – autentifikaciją. Analizės metu išsamiai apžvelgiama daiktų interneto tikslinė auditorija, esami žinomi metodai bei išskiriami svarbiausi pažeidžiamumai, su kuriais reikia kovoti. Kitą analizės dalį sudaro blokų grandinės technologijos analizė bei jos panaudojimo atvejai daiktų interneto įrenginių autentifikacijai atlikti. Pagrindinis visos analizės tikslas yra ištirti, kaip blokų grandinės technologiją galima pritaikyti daiktų interneto įrenginių autentifikacijai realizuoti.

1.2. Daiktų interneto architektūra

Architektūra, kuri palaikytų blokų grandinės technologiją daiktų interneto taikymo atvejuose, turėtų būti pritaikoma pagal taikomo atvejo generuojamo srauto kiekį [4]. Toks aspektas kelia susirūpinimą tradicinėse debesijos technologija paremtose architektūrose, kaip pateikta 1.1 pav., kuriose išsivystė į sudėtingesnes:

1. **Ūko kompiuterijos (angl. Fog Computing) architektūra**. Gausiai virtualizuota platforma, kuri teikia skaičiavimo, talpinimo ir tinklo paslaugas tarp galinių įrenginių ir tradicinių debesijos skaičiavimo duomenų centrų [5]. Sluoksnio vieta tipiškai (tačiau ne išskirtinai) orientuota ties tinklo pakraščiu.
2. **Kraštinės kompiuterijos (angl. Edge Computing) architektūra.** Architektūros ideologija panaši į ūko kompiuterija, tačiau stengiamasi dar daugiau debesijos krūvio perkelti arčiau naudotojo [6]**.**

Paveiksle matoma, kad trys architektūros priklauso nuo debesijos. Debesijos technologija paremtoje architektūroje, mazgo sluoksnyje (angl. Node Layer) surinkti duomenys yra nukreipiami tiesiai į debesį per daiktų interneto šliuzus.



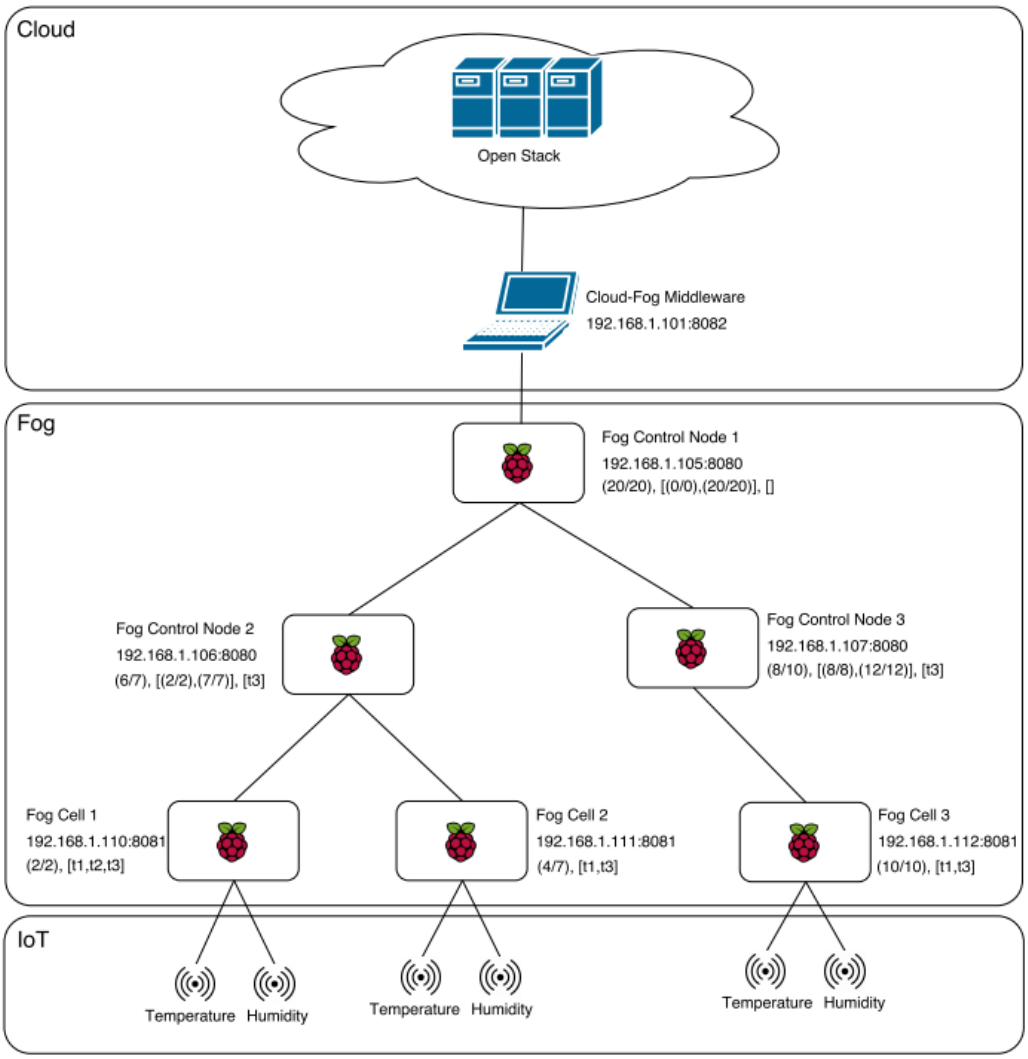
1.1 pav. Daiktų interneto architektūros [4]

Svarbu atkreipti dėmesį, kad tradicinės debesijos technologija paremtos daiktų interneto architektūros turi tam tikrų paveldimų pažeidžiamumų [7], kurie parodo, kad debesija yra klaidos taškas. Jeigu debesijos sluoksnis yra neprieinamas dėl kibernetinės atakos, tvarkymo darbų ar programinės įrangos problemų – visa sistema nustoja veikti.

Kitos dvi architektūros, ūko ir kraštinės, yra naujesnės už tradicinę debesijos architektūra. Šios architektūros perkelia dalį debesijos apdorojimo krūvio arčiau galinių įrenginių. Tai yra esminis

aspektas daiktų interneto taikymo atvejams, nes atsižvelgiant į apskaičiuotą sujungtų įrenginių augimo spartą [8], debesijos tinklo pajėgumas privalės augti. Atsižvelgiant į šią problemą, kraštinė ir ūko kompiuterijos gali būti naudojamos palaikant fiziškai paskirstytas, žemo vėlinimo ir paslaugų kokybę užtikrinančias programas, taip sumažinant tinklo srautą ir skaičiavimų apkrovą tradicinėse debesijos kompiuterijos sistemose.

Ūko kompiuteriją sudaro rinkinys lokalių šliuzų, kurie geba greitai reaguoti į daiktų interneto mazgų užklausas per konkrečias paslaugas. Toks mazgas gali sąveikauti vienas su debesijos sluoksniu, jeigu to reikia. Paveiksle 1.1 pav. pavaizduoti vienos plokštės kompiuteriai (angl. SingleBoard Computers arba SBC) yra mažos kainos ir elektros energijos sąnaudų reikalaujantis kompiuteris, kurį galima lengvai įdiegti daug vietos neturinčioje aplinkoje. Kaip pavyzdį (1.2 pav.) galima pateikti „Raspbbery Pi“ [9] kompiuterio panaudojimą tiriamajame darbe [10].



1.2 pav. Ūko kompiuterijos architektūros pavyzdys panaudojant „Rasberry Pi“ kompiuterį [10]

Pateiktame pavyzdyje realizuotas karkasas įrodo ūko kompiuterijos naudą, sumažindamas skaičiavimų krūvį debesijos sluoksnyje bei geba reaguoti į įvairius sistemos įvykius.

1.3. Daiktų interneto įrenginių autentifikacijos metodų analizė

Šiame skyrelyje apžvelgiami plačiai naudojami daiktų interneto komunikacijos protokolai bei konkretūs tyrėjų sukurti metodai komunikacijai realizuoti.

1.3.1. Daiktų interneto įrenginių komunikacijos protokolai

Daiktų interneto aplinkai būdingi faktoriai yra mažas perduodamų duomenų kiekis bei jų nepatikimumas [11]. Tokiomis sąlygomis efektyviausia naudoti MQTT arba CoAP protokolus duomenims perduoti. Taip pat svarbios priežastys MQTT ir CoAP protokolų naudojimui:

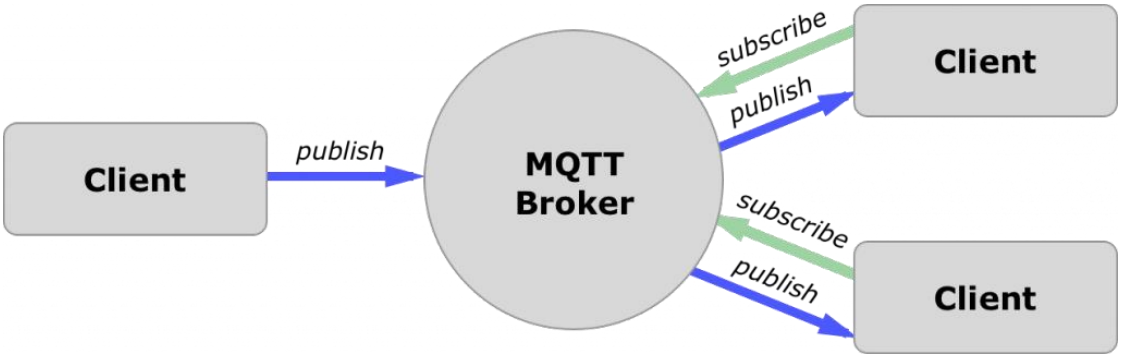
1. abu yra atviri standartai;
2. abu labiau nei HTTP tinkami suvaržytoms aplinkoms;
3. suteikia mechanizmą asinchroniškai komunikacijai;
4. turi platų realizacijų spektrą.

MQTT protokolas

MQTT yra „mašina su mašina“ (angl. machine-to-machine arba M2M) daiktų interneto susijungimo protokolas. Jis buvo sukurtas kaip ypač mažos apimties žinučių paskelbimo ir gavimo transportavimo priemonė. Jis naudingas prisijungimams prie nuotolinių vietų, kuriose reikalingas mažas kodo pėdsakas ir jeigu duomenys yra apmokestinti.

MQTT protokolo architektūra

MQTT protokolą sudaro kliento ir serverio modelis (1.3 pav.), kuriame kiekvienas jutiklis yra klientas, besijungiantis į serverį, žinomą kaip tarpininką (angl. broker), naudojantis TCP ryšiu.



1.3 pav. MQTT protokolo architektūra [12]

MQTT yra orientuotas į žinučių sistemą. Kiekviena žinutė yra diskreti duomenų dalis, neaiški tarpininkui. Kiekviena žinutė yra publikuojama adresu, žinomu kaip tema (angl. topic). Klientas taip

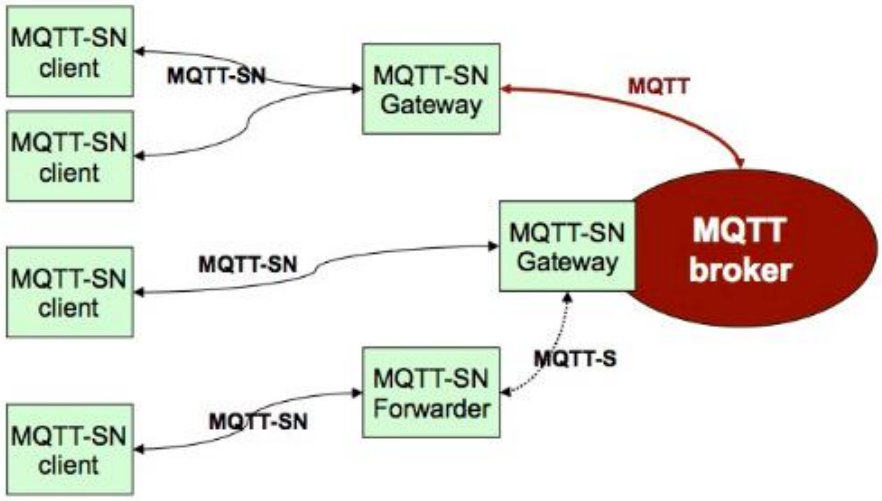
pat gali prisijungti prie daugybės temų. Kiekvienas temą užsiprenumeravęs klientas gauna kiekvieną jos publikuojamą žinutę pagal temą.

MQTT protokolo saugumas

MQTT tarpininkas (angl. broker) gali reikalauti vartotojo vardo ir slaptažodžio autentifikuojant klientų prisijungimą. Užtikrinant privatumą, TCP sujungimas gali būti užšifruojamas naudojant SSL arba TLS.

MQTT-SN

MQTT-SN galima traktuoti kaip vieną iš MQTT versijų, kuri yra pritaikyta belaidžio tinklo aplinkos ypatumams, kaip žemas pralaidumas, dažnos ryšio klaidos, trumpas pranešimo ilgis ir t.t. [13]. MQTT-SN taip pat optimizuotas mažos kainos, baterija valdomiems įrenginiams su ribotais skaičiavimo ir talpos resursais. Protokolo architektūra nuo standartinio MQTT protokolo skiriasi tarp MQTT-SN kliento ir tarpininko esančiais papildomais komponentais (1.4 pav.).



1.4 pav. MQTT-SN architektūra [13]

Vienas iš galimų komponentų yra MQTT-SN šliuzas (angl. MQTT-SN Gateway), kurio funkcija yra vertimas tarp MQTT ir MQTT-SN protokolų. Šliuzas gali būti tiesiog integruotas į MQTT tarpininką ir arba naudojamas kaip atskiras tinklo komponentas. MQTT-SN klientai taip pat gali pasiekti šliuzą per MQTT-SN persiuntėją (angl. MQTT-SN forwarder).

CoAP protokolas

CoAP yra suvaržytos programos protokolas (angl. Constrained Application Protocol), sukurtas CoRE (Constrained Resource Environments) IETF grupės.

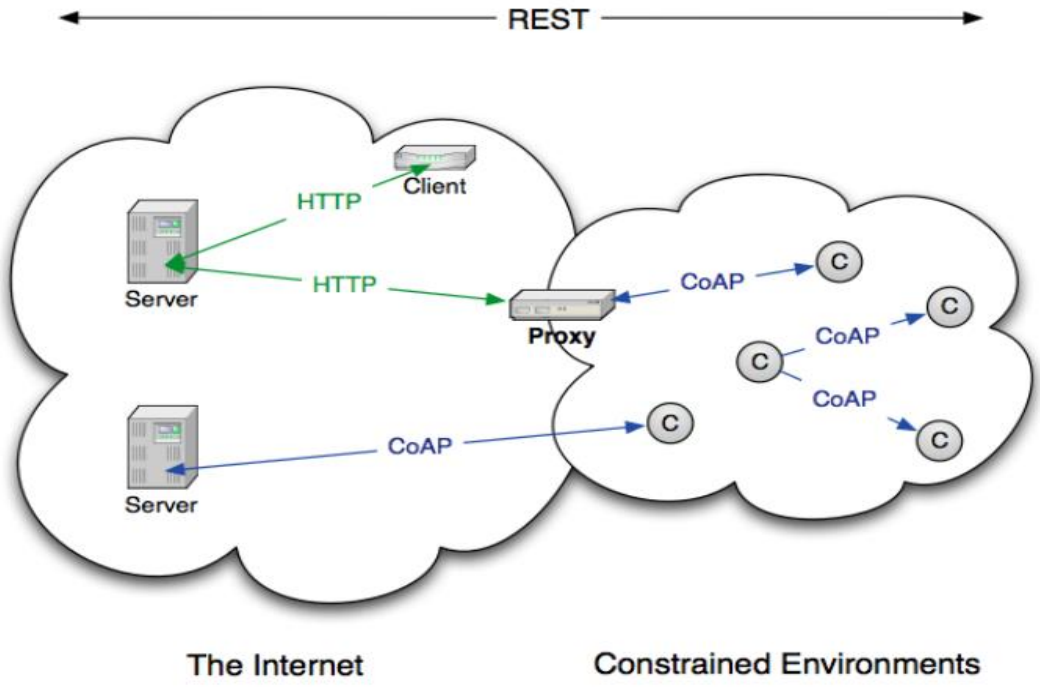
CoAP protokolo architektūra

Kaip ir HTTP, CoAP yra dokumentų perdavimo protokolas. Tačiau priešingai nei HTTP, CoAP yra sukurtas suvaržytoms aplinkoms.

CoAP paketas yra gerokai mažesnis už HTTP TCP srautus. Bitų laukai ir formavimas iš „string“ tipo į sveikus skaičius naudojami nuolatos vietai sutaupyti. Paketai yra paprastai sugeneruoti ir gali būti išnagrinėjami vietoje nenaudojant papildomos RAM atminties suvaržytuose prietaisuose.

CoAP veikia per UDP, o ne TCP. Klientai ir serveriai komunikuoja per sujungimo neturinčias datagramas. Pakartotiniai bandymai ir pertvarkymai yra diegiami programos dėkle. Panaikinant poreikį naudoti TCP, suteikiama pilna IP tinklaveika. CoAP leidžia adresavimui naudoti UDP įprastą ir daugiaabonentį transliavimą.

CoAP seka kliento ir serverio modelį. Klientai atlieka užklausą į serverius, o serveriai atsiunčia atsakymą (1.5 pav.). Klientai gali gauti GET, PUT, POST ir DELETE metodų resursus.



1.5 pav. CoAP protokolo architektūra [14]

CoAP yra suprojektuotas suderinamumui su HTTP ir RESTful didelio masto tinklu per paprastus tarpininkus (angl. proxies). Kadangi CoAP yra paremtas datagrama, protokolas gali būti naudojamas viršuje SMS ir kitų paketais paremtų komunikacijos protokolų.

CoAP protokolo saugumas

Kadangi CoAP yra sukurtas UDP pagrindu, o ne TCP, tai reiškia, kad SSL arba TLS nėra galimi apsaugai užtikrinti. DTLS (angl. Datagram Transport Layer Security) suteikia panašų užtikrinimą kaip ir TLS, tačiau perduodamiems duomenims per UDP.

MQTT ir CoAP palyginimas

MQTT ir CoAP yra naudingi protokolai daiktų interneto technologijų srityje, tačiau turi svarbius skirtumus (1.1 lentelė) [15].

1.1 lentelė. MQTT ir CoAP palyginimas

Charakteristika	MQTT	CoAP
Žinučių formavimas	Daugelis su daugeliu komunikacija	Komunikacija nuo taško iki taško
	Temos	URI
Transportavimas	TCP	UDP
Architektūra	Paskelbimo ir prenumeravimo modelis	Užklausos ir atsako modelis
Sluoksniai	Vienas sluoksnis	Du sluoksniai: 1. Užklausos ir atsako 2. Transakcijos
Sauga	SSL ir TLS	DTLS
Patikimumas	3 paslaugos kokybės lygiai	4 paslaugos kokybės lygiai
Sparta	Žemesnis vėlavimas	Mažesnis paketų praradimas

MQTT yra „many-to-many“ (liet. daugelis su daugeliu) komunikacijos protokolas, perduodantis žinutes tarp klientų per centrinį tarpininką. Jis atskiria gamintoją ir naudotoją, leisdamas klientams publikuoti ir tarpininkui nuspręsti, kur nukreipti ir kopijuoti žinutes. Nors MQTT geba išlaikyti žinutes, jis geriausiai veikia kaip komunikacijos magistralė tiesioginiams duomenims. Tuo tarpu CoAP yra labiau skirtas būsenos perdavimo modeliui – didžiulis trūkumas, siekiant perduoti konkrečias reikšmes. CoAP taip pat stokoja saugumo ir jo lankstumo. MQTT šią problemą išsprendžią, suteikdamas saugią ir autentifikuotą kliento (prietaiso) prieigą prie serverio.

1.3.2. Tyrėjų suprojektuoti daiktų interneto komunikacijos metodai

Daiktų interneto komunikacijos metodai yra labai įvairūs ir skirtingi. Straipsnyje „Daiktų interneto autentifikacijos protokolai: išsamus tyrimas“ daiktų interneto autentifikacijos metodai suskirstyti į keturias aplinkas:

1) aparato ir aparato komunikacija (angl. machine to machine communication arba M2M);
2) automobilių internetas (angl. internet of vehicles arba IoV);
3) energijos internetas (angl. internet of energy arba IoE);
4) jutiklių internetas (angl. internet of sensors arba IoS) [16].

Autoriai analizuoja ir apibendrina daugiau nei 40 daiktų interneto autentifikacijos protokolų. Dėl baigiamojo darbo apimties susitelkiama tik į vieną iš keturių aplinkų – M2M komunikaciją. Šios komunikacijos protokolai pateikiami lentelėje.

1.2 lentelė. M2M komunikacijos protokolų palyginimo lentelė

Protokolas	Tinklo modelis	Tikslas	Pagrindiniai procesai	Gebėjimai (+) ir trūkumai (-)
GLARM [17]	Paremtas 3GPP standartu su trejais domenais, kurių tarpe prieigos tinklai, išsivystę paketų branduoliai bei ne 3GPP domenas (pvz. internetas).	Užtikrinama subjekto bendra autentifikacija ir saugus pagrindinis susitarimas.	1. Inicijavimo etapas. 2. Grupinė autentifikacija ir pagrindinio susitarimo etapas.	+DoS, nukreipimo ir žmogaus viduryje (angl. man-in-the-middle) atakų atsparumas. +Pakankamai mažai perteklinių skaičiavimų. +Skaičiavimas sudėtingumas gerokai mažesnis nei schemose [18] [19] ir [20]. +Gali užtikrinti QoS mašinos tipo komunikacijos įrenginiams. -Tam tikri privatumo modeliai nėra analizuojami, pvz.

				vietos arba tapatybės privatumas. -Neatsižvelgiama į talpos sąnaudas.
S2M [21]	Du belaidžiai įrenginiai.	Pasiekiama kintančio nuotolio autentifikacija ir aktyvus atakų aptikimas.	1. Audio rankos paspaudimo (angl. audio-handshake) etapas. 2. Mišraus signalo generavimo etapas. 3. Parametrų išgavimo ir talpinimo etapas.	+Efektyvus žemo klaidų rodiklio atžvilgiu lyginant su DISWN [22], LDTLS [23], PLTEA [24] ir SeArray [25]. +Aktyvus atakų aptikimas (pvz. audio kartojimo ataka). -Privatumo išlaikymas nėra analizuojamas lyginant su GLARM [26]. -Neatsižvelgiama į talpos sąnaudas.
SEGR [27]	3GPP WiMAX mašinos tipo komunikacija.	Užtikrinama abipusė autentifikacija ir saugus pagrindinis susitarimas tarp mašinos tipo įrenginių.	1. Inicijavimo etapas. 2. Tarptinklinio ryšio etapas.	+Efektyvus komunikacijos pertekliaus prasme, lyginant su tradicine tarptinklinio ryšio autentifikacijos schema ir optimizuota tarptinklinio ryšio schema, nurodyta [28]. +Efektyvus skaičiavimo sudėtingumą atžvilgiu lyginant su schema neturinčia agregavimo -Atsparumas atakoms nėra studijuojamas. -Privatumo išlaikymas nėra analizuojamas

				lyginant su GLARM [26]. -Neatsižvelgiama į talpos sąnaudas.
SE-AKA [29]	3GPP standartas, turintis tris domenus, įvardintus kaip prieigos tinklo domenas, prižiūrimo tinklo domenas bei namų tinklo domenas.	Garantuojamas privatumo išsaugojimas ir dvikryptis raktų slaptumas.	1. Pasiruošimas ir inicializavimas. 2. Protokolo įvykdymas pirmajai įrangai. 3. Protokolo įvykdymas likusiai grupės įrangai. 4. Grupės nario prisijungimas prie grupės ir jos palikimas.	+Atsižvelgiama į duomenų integralumą ir užtikrinamas naudotojo privatumas. +Atsparumas atakoms (DoS atakos, nukreipimo atakos, žmogaus viduryje ataka ir atkartojimo ataka) +SE-AKA autentifikavimo žinučių perteklius yra žemesnis nei kituose egzistuojančiuose AKA protokoluose. +Skaičiavimų perteklius didesnis nei kitų tradicinių protokolų, tokių kaip [30] šaltinyje. +Mažesnės talpos sąnaudos lyginant su kitais protokolais. -Tam tikri privatumo modeliai nėra analizuojami, pvz. vietos arba tapatybės privatumas.
[31]	Mobilus WiMAX tinklas, turintis prieigos paslaugos tinklą	Užtikrinama abipusė autentifikacija, privatumo išsaugojimas bei	1. Išankstinio diegimo etapas. 2. Pradinio autentifikavimo etapas.	+Efektyvus skaičiavimų ir komunikacijos pertekliaus prasme

		atsparumas domino efektui.	3. Perdavimo autentiškumo užtikrinimas.	lyginant su trimis schemomis. +Atsižvelgiama į privatumo išsaugojimą. -Neatsižvelgiama į talpos sąnaudas. -Atakų atsparumas nėra studijuojamas. -Nėra pateiktas joks grėsmių modelis. -Klaidų aptikimas ir defektų tolerancija nėra svarstoma.
[32]	Mobilieji naudotojai, namų tinklo sąsajos ir M2M serveris.	Užtikrinamas abipusė autentifikacijos procesas M2M namų tinklo paslaugoje.	1. Parengimas. 2. Registracijos etapas. 3. Prisijungimo ir autentifikacijos etapas. 4. Slaptažodžio atnaujinimo etapas. 5. Namų tinklo sąsaja prisijungia prie TD-SCDMA (Time Division Synchronous Code Division Multiple Access network).	+Efektyvus skaičiavimų kiekio ir komunikacijos apimties atžvilgiu, lyginant su [33] protokolu. +Atsparumas spėliojimo atakai, pavogto tikrintojo atakai, apsimetinėjimo atakai ir atkartojimo atakai. -Privatumo išlaikymas nėra analizuojamas lyginant su GLARM [26]. -Neatsižvelgiama į talpos sąnaudas. -Neatsisakymo trūkumas lyginant su PBA schema [34].
CPAL [35]	Tarptinklinio ryšio tinklo architektūra, naudojanti namų	-Suteikiamas stipri anonimiška autentifikacija.	1. Sistemos inicijavimas.	+Efektyvus komunikacijos pertekliaus ir

	autentifikacijos centrą (HAC), pasitikėjimo susiejimo serverį (TLS) ir apsilankymo autentifikacijos serverį (VAS).	-Garantija naudotojui stebėti ginčijamą prieigos užklausą -Pasiekiamas anoniminio naudotojo susiejimas ir veiksmingas naudotojo dinaminės narystės panaikinimas.	2. Tarptinklinis ryšys. 3. Naudotojo sekimo algoritmas. 4. Anoniminio naudotojo susiejimas. 5. Naudotojo narystės panaikinimas.	skaičiavimų kainos atžvilgiu, lyginant su dviem stipriomis anonimiškomis schemomis [36] ir [37]. +Atsižvelgiama į duomenų integralumą ir užtikrinamas naudotojo privatumas. +Atsparumas atakoms, tokios kaip DoS ir apsimetinėjimo ataka. -Tam tikri privatumo modeliai nėra analizuojami, tokie kaip vietos privatumas. -Neatsisakymo trūkumas lyginant su PBA schema [34].
Duth [38]	-Android išmanieji įrenginiai.	-Patenkinamas naudotojo patogumas esant mažam klaidų atmetimo lygiui. -Pasiekiamas autentifikacijos procesas Android išmaniesiems įrenginiams.	1. Savybių rinkinio išgavimas ir talpinimas registracijai. 2. Dvigubo faktoriaus autentifikacija.	+Gali pagerinti patogumą naudotojui. +Pagerinamas saugumas nepridedant papildomos aparatinės įrangos. -Nėra jokio grėsmių modelio.
LGTH [28]	Paremtas 3GPP standartu su trejais domenais: 1. EUTRAN 2. EPC 3. Ne 3GPP domenas, pvz. internetas	Užtikrinama subjekto abipusė autentifikacija ir saugus pagrindinis susitarimas.	1. Inicijavimo etapas. 2. Grupinė autentifikacija ir pagrindinio susitarimo etapas.	+Efektyvus signalizavimo ir skaičiavimų pertekliaus atžvilgiu, lyginant su schemomis [18] ir [39]. +Atsparumas atakoms, kaip pavyzdžiui atkartojimo ataka,

				nukreipimo ataka ir žmogaus viduryje ataka. +Atsižvelgiama į duomenų integralumą. -Privatumo išlaikymas nėra analizuojamas lyginant su GLARM [26]. -Neatsižvelgiama į talpos sąnaudas.-Neatsisakymo trūkumas lyginant su PBA schema [34].

Atsižvelgiant į lentelės duomenis, galima išskirti tokius dažniausiai pasikartojančius autentifikacijos protokolų trūkumus:

1) privatumo išlaikymas nėra analizuojamas;
2) neatsižvelgiama į talpos sąnaudas;
3) negalėjimo išsižadėti (angl. non-repudation) trūkumas.

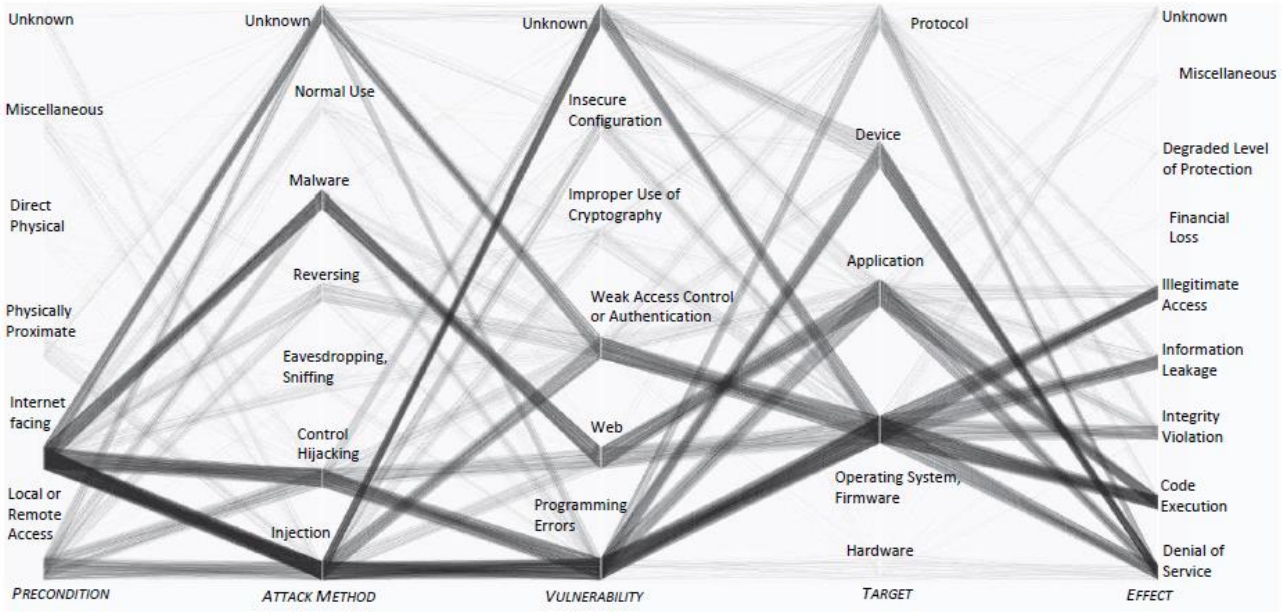
1.4. Atakų prieš daiktų interneto įrenginius analizė

B. Šneieris įvardina dabartinę įterptinių sistemų situaciją kaip krizę, kai apskaičiavimai įterpiami tiesiai į aparatinę įrangą, kaip yra daroma su daiktų internetu [40]. Šie įterptiniai kompiuteriai yra pilni pažeidžiamumų ir nėra jokio tinkamo būdo juos ištaisyti. Šiuos įrenginius kurianti industrija yra mažiau pajėgi taisyti problemas, lyginant su kompiuterių ir programinės įrangos industrijomis.

Sistemų kūrėjų problema yra ta, kad atnaujinimams tiesiog nėra laiko. Baigus gaminti vieną produktą, iškart pradedamas kurti kitas. Senesnių produktų palaikymas tiesiog nėra prioritetas. Tai ilgainiui sukelia saugumo spragų, kadangi įsilaužėliai atranda vis naujų kūrėjų paliktų klaidų.

Straipsnyje „Įterptinių sistemų saugumo rizikos“ sudarytas paveikslas (1.6 pav.) grafiškai parodo atakų prieš įterptines sistemas scenarijus ir jų tendencingumą [41].

Panaudota heuristika identifikavo ir išskyrė artimiausius duomenis iš viešos CVE (angl. common vulnerabilities and exposures, liet. dažni pažeidžiamumai ir išryškėjimai) duomenų bazės su daugiau nei 60000 įrašų.



1.6 pav. Dažniausių atakų prieš įterptines sistemas scenarijai [41]

Iš atliktos taksonomijos sprendimo matoma, kad išryškėja autentifikacijai jautrūs aspektai. Kaip konkretų pavyzdį galima pateikti sąryšį tarp šių punktų:

1) sąsaja su internetu (angl. internet facing);
2) injekcija;
3) silpna prieigos kontrolė ar autentifikacija;
4) operacinė sistema, programinė aparatinė įranga (angl. firmware);
5) neteisėte prieiga.

1.5. Blokų grandinės technologija

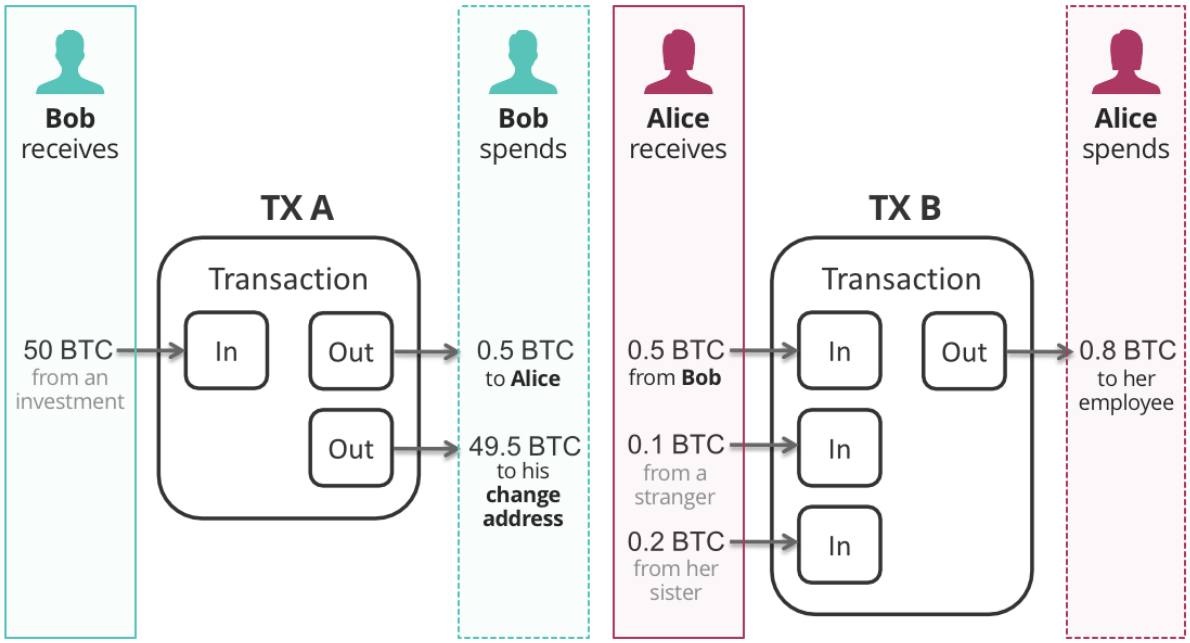
Blokų grandinės technologija yra nepaprastai skaidrus ir decentralizuotas būdas suvedinėti transakcijų sąrašą [42]. Žinomiausias technologijos panaudojimas yra orientuotas į skaitmenines valiutas, tokias kaip bitkoinas. Blokų grandinės panaudojimų naujoms valiutoms kurti yra įvairių. Šis būdas, leidžiantis sukurti greitus, pigius ir saugius viešus įrašus, reiškia, kad jis gali būti naudojamas daugeliui su finansinėmis užduotimis nesusijusioms užduotims atlikti. Blokų grandinė konkrečiai tinka situacijoms, kuriose reikalinga išmanyti nuosavybės teisės istoriją. Technologijos vystymosi proveržis

yra savarankiškai įvykstantys kontraktai, grindžiantys kelią kompanijoms, kurios veikia be žmogaus įsikišimo.

1.5.1. Blokų grandinės veikimas

Blokų grandinės technologija suteikia panašų įrašų saugojimo funkcionalumą kaip tradicinės suvestinės, tačiau tai atliekama be centralizuotos architektūros. Blokų grandinė decentralizuoja suvestinę taip, kad kiekvienas asmuo turėtų po kopiją. Visi gali prašyti, kad jų transakcijos būtų įrašytos į suvestinę, tačiau įrašai priimami tik jeigu visi naudotojai sutinka, kad jis yra teisėtas. Šis patikrinimas yra atliekamas patikimai ir automatiškai vietoje kiekvieno naudotojo, sukuriant greitą ir saugią suvestinės sistemą, kuri yra nepaprastai saugi nuo įsilaužimų.

Su kiekviena ketinama įrašyti transakcija yra sukomponuojama kita nauja transakcija, sudarant vadinamą „bloką“, kuris pridedamas kaip naujausia nuoroda ilgoje grandinėje istorinių transakcijų (1.7 pav.) [43].



1.7 pav. Bitkoino transakcijos vykdymo pavyzdys [43]

Ši grandinė suformuoja blokų grandinės suvestinę, kurią turi visi naudotojai. Visas procesas įvardinamas kaip „kasinėjimas“. Visi gali tapti „kasinėtojais“ ir varžytis dėl sudėtingo matematinio uždavinio, reikalingo pridėti galiojantį užšifruotą transakcijos bloką į blokų grandinę. Atsiradus neatitikimui, blokas yra atmetamas. Priešingu atveju jis pridedamas ir lieka amžinas viešas įrašas, kurio negali ištrinti joks naudotojas. Tokio tipo blokų grandinės yra įvardinamos kaip nereikalaujančios

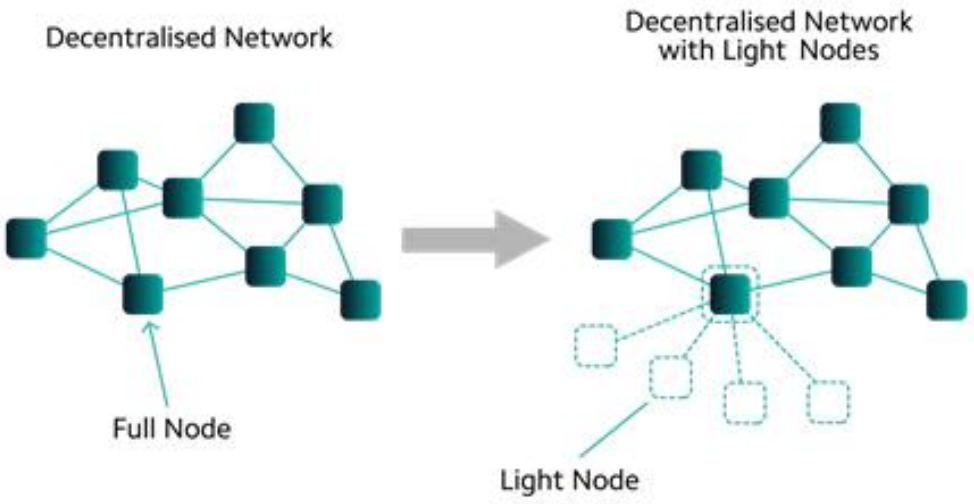
leidimų, kadangi yra skaidrios, decentralizuotos ir prie jų prisijungti gali visi. Taip pat yra galimybė sukurti leidimų reikalaujančią blokų grandinę, kurioje veiksmus gali atlikti tik ribota grupė naudotojų.

1.5.2. Pilnieji ir lengvieji mazgai

Mazgas yra bet kuris įrenginys, prijungtas prie blokų grandinės tinklo [44]. Pavyzdžiui, bet kuris kompiuteris, prisijungęs prie bitkoino tinklo, gali būti prilyginamas bitkoino mazgui. Dalis mazgų blokų grandinėse yra žinomi kaip pilni mazgai (angl. full nodes). Jie žinomi kaip pilni mazgai, kadangi šie mazgai tikrina kiekvieną pristatytą bloką ir transakciją.

Esminis faktorius yra tai, kad pilnas mazgas turi turėti visos blokų grandinės kopiją. Tad kiekviena bet kada sukurta transakcija ir blokas turi būti parsiųsti. Tai užtikrina, kad blokų grandinės negalėtų valdyti vienas konkretus subjektas. Taigi, kuo daugiau blokų grandinėje yra pilnų mazgų, tuo labiau sistema tampa decentralizuota ir reikalauja mažiau pasitikėjimo.

Tam tikri pilni mazgai taip pat gali prižiūrėti ir lengvuosius mazgus (angl. lightweight nodes). Lengvieji mazgai patikrina transakcijas naudodami metodą – supaprastintą pervedimo verifikaciją (angl. simplified payment verification arba SPV). SPV leidžia mazgui patikrinti, ar transakcija buvo įtraukta į bloką išvengiant reikmės parsisiųsti visą blokų grandinę. Naudojant SPV, pilni mazgai prižiūri lengvuosius mazgus, suteikdami jiems prieigą prisijungti ir transliuoti savo transakcijas į tinklą, taip pat pranešant jeigu transakcija juos paveikė.



1.8 pav. Lengvųjų mazgų panaudojimas blokų grandinės tinkle [45]

Lengviesiems mazgams reikia parsisiųsti tik visų blokų, esančių blokų grandinėje, antraštes, o tai reiškia, kad parsisiuntimo ir talpos reikalavimai tampa gerokai mažiau intensyvūs, lyginant su pilnu mazgu.

1.6. Blokų grandinės technologija paremtas daiktų internetas

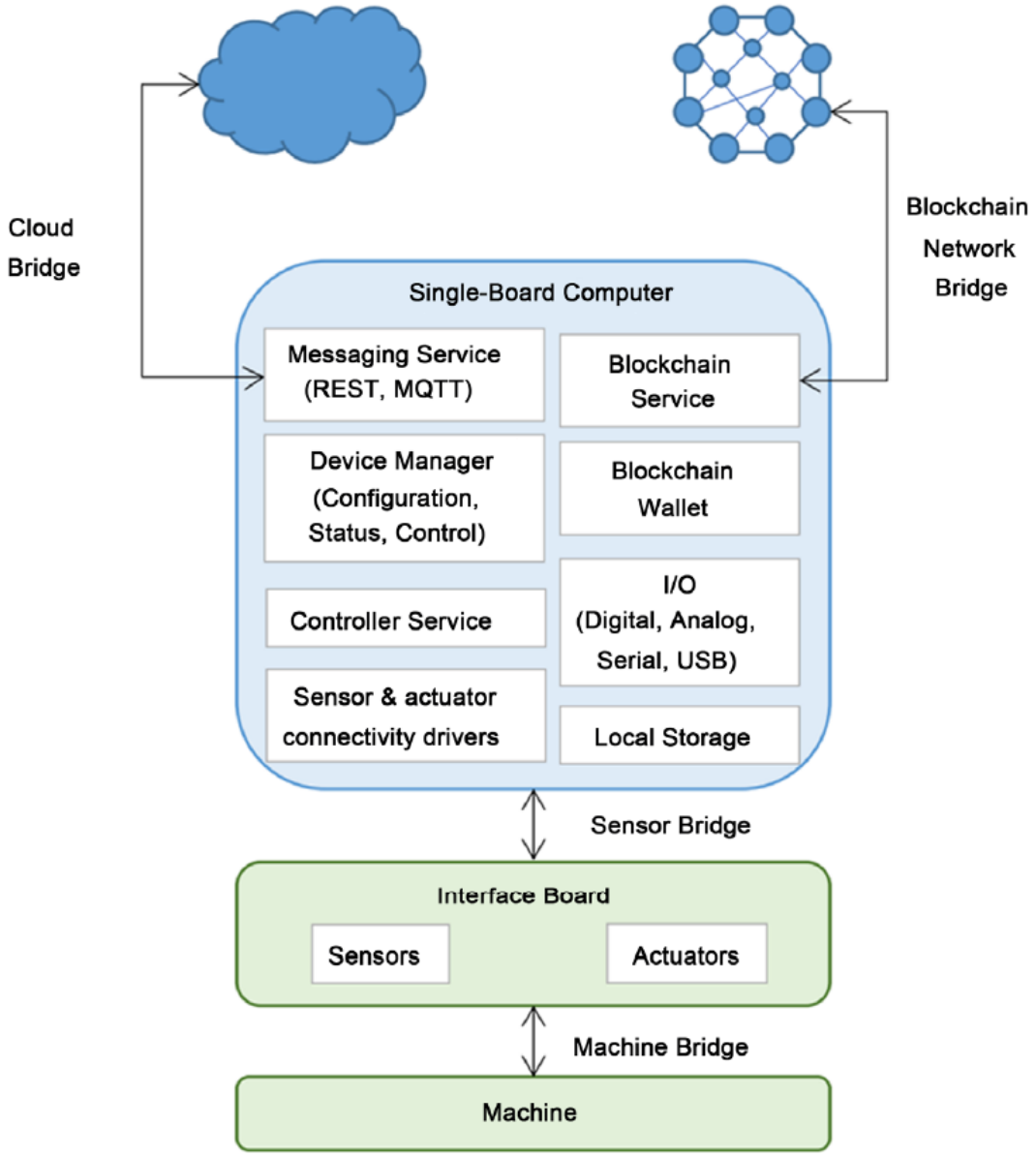
Blokų grandinės technologijos privalumai, kaip nekintamumas, skaidrumas, patikimumas, duomenų šifravimas ir atsparumas, gali padėti išspręsti daugumą daiktų interneto architektūros trūkumų [46]. Esminiai blokų grandinės skirtumai lyginant su tradicine daiktų interneto architektūra pateikti 1.3 lentelė.

1.3 lentelė. Tradicinio ir blokų grandinės technologija paremto daiktų interneto palyginimas

Charakteristika	Tradicinis daiktų internetas	Blokų grandinėmis paremtas daiktų internetas
Plečiamumas	Centralizuotos sistemos valdomų įrenginių kiekis yra ribotas.	Lengva pridėti naujus mazgus dėl sistemos pasiskirstymo.
Efektyvumas	Duomenų apdorojimas ir saugojimas centralizuotoje sistemoje yra brangus.	Sistemos pasiskirstymas sumažina duomenų apdorojimo ir talpinimo kainą lyginant su centralizuota sistema.
Stabilumas	Jeigu centrinis serveris ar tinklas nustoja veikti, prijungti įrenginiai negali būti naudojami.	Atsiradus problemoms su serveriu ar daliai tinklo problemų, visa sistema nėra paveikiama.
Sauga	Sunku patvirtinti duomenų klastojimą ir atstatymą.	Duomenų klastojimas yra sudėtingas, kadangi jie yra paskirstyti per sistemą.

Daiktų internetas yra ypač naudingas industriniam ir gamybiniam taikymui, pavyzdžiui, automatizacija, nuotolinė mašinų diagnostika, prognozuojamas sveikatos valdymas industrinėse mašinose bei tiekimo grandinės valdymas. Debesija paremta gamyba (angl. Cloud-Based Manufacturing) pastaruoju metu yra didelį poreikį turintis gamybos modelis, laviruojantis daiktų interneto technologijomis. Nors debesija paremta gamyba pagal poreikį suteikia prieigą prie gamybos resursų, tačiau transakcijoms tarp naudotojų, kurie nori pasinaudoti gamybos paslaugomis, reikalingas patikimas tarpininkas.

Straipsnio [47] autoriai pateikia decentralizuotą, pasitikėjimo nereikalaujančią (angl. trustless) ir lygiarangią platformą (angl. peer-to-peer platform) pavadinimu BPIIoT.



1.9 pav. Blokų grandinės platforma skirta industriniam daiktų internetui (BPIIoT) [47]

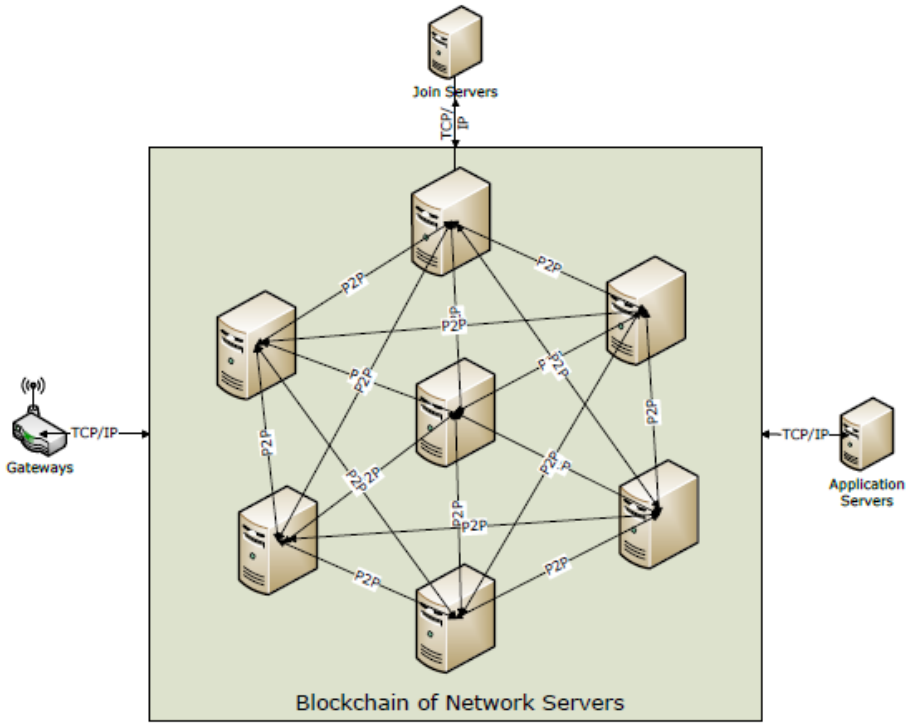
Tai blokų grandinės technologija paremtas industrinis daiktų internetas.

Išmaniosios gamybos branduolyje kibernetinės fizinės sistemos (angl. Cyber-Physical Systems) turi rimtų saugumo problemų, ypač terminalų M2M komunikacijos saugumo srityje. Straipsnyje [48] blokų grandinės technologija pateikiama kaip galimas sprendimas komunikacijos saugos problemoms spręsti tarp skirtingo tipo įrenginių kibernetinėse fizinėse. Atsižvelgiant į blokų grandinės technologijos principus, suprojektuota blokų grandinė saugiai M2M komunikacijai. Kaip ir įprastą komunikacijos sistemą, M2M sudaro viešojo tinklo vietos, įrenginių vietos bei privačios vietos, kur blokų grandinės struktūra sukurta tarp viešosios ir privačiosios dalies. Autoriai pademonstravo,

kad blokų grandinės technologija gali efektyviai spręsti mašinų plėtimosi saugumą produkcijos proceso metu, o komunikacijos duomenys tarp mašinų negali būti manipuliuojami.

Su daiktų interneto nuolatiniu augimu mažos energijos plataus ploto (angl. low power wide area arba LPWA) technologijos tapo populiaresnės. Viena iš technologijų – sistema tolimo nuotolio plataus ploto tinklui (angl. long range wide-area network arba LoRaWAN), dažniausiai naudojama privačių kompanijų ir organizacijų, o tai sukelia pasitikėjimo problemų tarp programos naudotojo ir tinklo operacijų.

Straipsnyje [49] pateiktas blokų grandinės technologija paremtas sprendimas, kuris sukuria atvirą, patikimą, decentralizuotą ir įsilaužimams atsparią sistemą tolimo nuotolio plataus ploto tinklui. 1.10 pav. pavaizduota blokų grandinės sistema, sukurta LoRaWAN tinklo serveriuose.



1.10 pav. Blokų grandinės architektūra skirta LoRaWAN serveriui [49]

Tokia blokų grandinės realizacijos vieta pasirinkta dėl šių priežasčių:

1. LoRaWAN šliuzai įprastai yra ribotų resursų ir patalpinti lauko sąlygomis, dėl to nėra tinkami kandidatai blokų grandinės realizacijai.
2. LoRaWAN prijungimo serveriai (angl. join servers) įprastai skirti generuoti sesijos raktus ir nėra tinkami blokų grandinės funkcijoms atlikti.
3. LoRaWAN taikomuosius serverius paprastai pateikia klientai, siekdami apdoroti pagrindinius verslo duomenis, todėl jie nėra tinkami blokų grandinės funkcijoms atlikti.

1.7. Blokų grandinių valdymo programinė įranga

Blokų grandinių valdymo programinė įranga, naudojama kriptovaliutose, atlieka veiksmus su skaičiais, kurie nurodo tam tikrą turimą elektroninės valiutos kiekį. Norint blokų grandinėse talpinti duomenis, reikalingus autentifikacijai, toks principas nėra tinkamas. Autentifikacijos atveju tikslingiau naudoti išmaniųjų kontraktų (angl. Smart contracts) principą, kai vietoje skaičių naudojamas autentifikacijai reikalingų duomenų rinkinys.

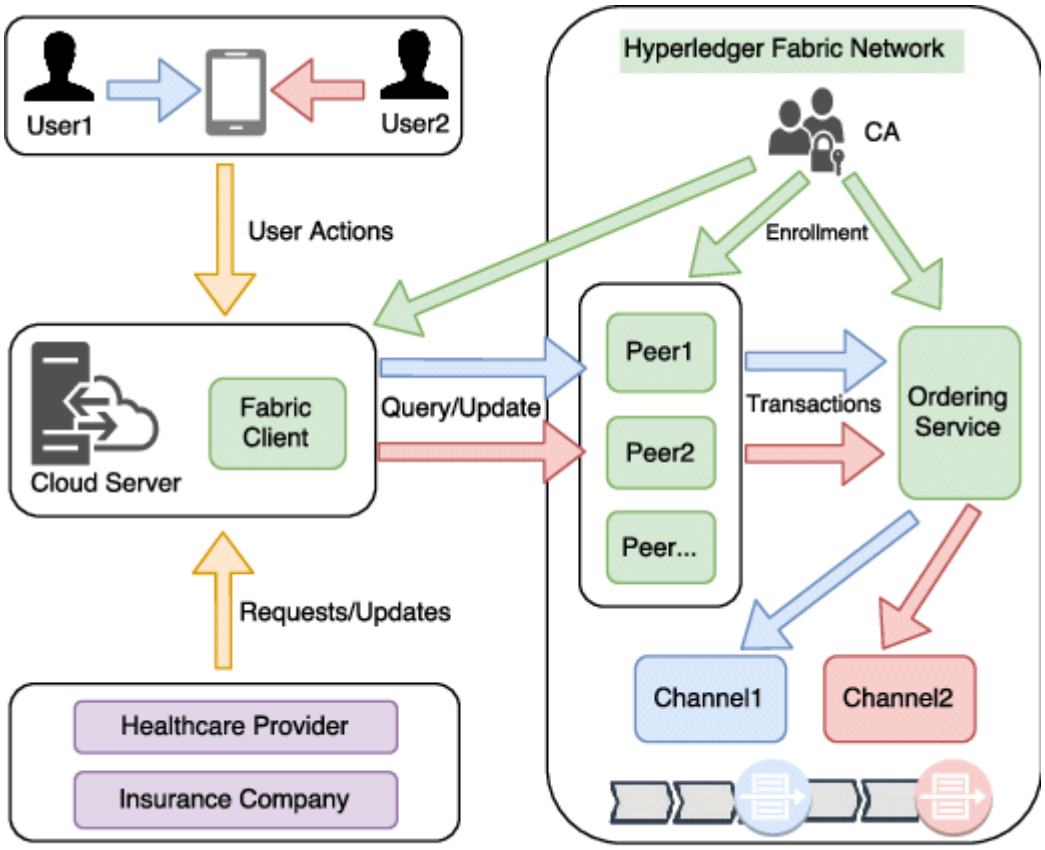
Šiame tyrime lyginami trys skirtingos išmaniuosius kontraktus realizuojančios programinės įrangos, kurių charakteristikų palyginimas pateiktas 1.4 lentelėje [50].

1.4 lentelė. Blokų grandinių valdymo programinė įranga

Charakteristika	Ethereum	Hyperledger Fabric	R3 Corda
Platformos apibrėžimas	▪ Bendrinė blokų grandinės platforma	▪ Modulinė blokų grandinės platforma	▪ Specializuota paskirstyta suvestinės platforma skirta finansinėms institucijoms
Valdomas	▪ Ethereum kūrėjai	▪ Linux Foundation	▪ R3
Veikimo rėžimas	▪ Be leidimų ▪ Viešas arba privatus	▪ Su leidimais ▪ Privatus	▪ Su leidimais ▪ Privatus
Sutarimas	▪ „Kasinėjimas“ paremtas darbo įrodymu (angl. proof-of-work) ▪ Suvestinės lygis	▪ Platus sutarimo supratimas, leidžiantis daugybę būdų ▪ Transakcijos lygis	▪ Specifinis sutarimo supratimas ▪ Transakcijos lygis
Išmanieji kontraktai	▪ Išmaniųjų kontraktų kodas Solidity kalba	▪ Išmaniųjų kontraktų kodas Go arba Java kalba	▪ Išmaniųjų kontraktų kodas Kotlin arba Java kalba ▪ Išmanioji teisinė sutartis
Valiuta	▪ Ether ▪ Prieigos raktai naudojami išmaniuosiuose kontraktuose	▪ Nėra ▪ Valiuta ir prieigos raktai naudojami grandinės kode	▪ Nėra

Visi trys blokų grandinių valdymo karkasai turi labai skirtingas vizijas, orientuotas į galimas taikymo sritis.

Tiek „Fabric“, tiek „Corda“ plėtra turi konkrečius panaudos atvejus. „Corda“ pagrindinė sritis yra finansinių paslaugų industrija. „Fabric“ stengiasi suteikti modulinę ir plečiamą architektūrą, kurią būtų galima naudoti įvairiose industrijose, pavyzdžiui, bankai, sveikatos apsauga, tiekimo grandinės ir t.t. Paveiksle 1.11 pav. pateiktas vienas iš šių taikymo pavyzdys, kai „Hyperledger Fabric“ naudojamas suteikti saugų duomenų dalinimosi tarp asmenų, sveikatos apsaugos įstaigų ir draudimo kompanijų.



1.11 pav. „Hyperledger Fabric“ taikymo pavyzdys [51]

„Ethereum“ taip pat nėra išskirtinai pritaikytas tam tikrai industrijai. Tačiau lyginant su „Fabric“, „Ethereum“ nėra orientuotas į modulinį sprendimą. „Ethereum“ orientuotas į bendrinės platformos tiekimą, kuri skirta įvairių tipų transakcijoms ir programoms.

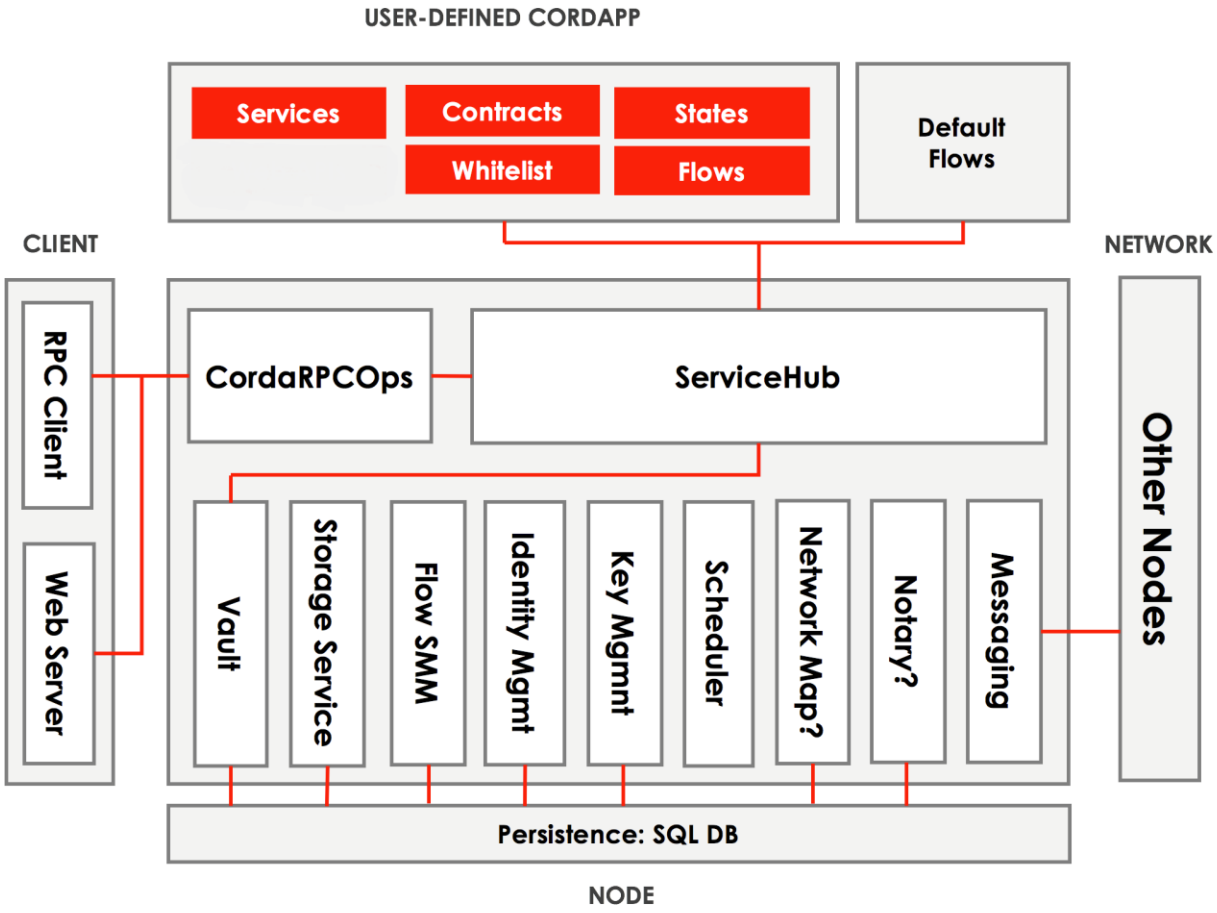
Tyrime išskiriamas taikymo atvejis, kai blokų grandines norima naudoti daiktų interneto įrenginių autentifikacijai. Autentifikacijos metodas turi lengvai integruotis su kitomis sistemomis.

„Fabric“ atveju, daiktų interneto autentifikaciją galima realizuoti kaip vieną iš modulių, kurį galima integruoti tiek su kitomis sistemomis, tiek su esamais „Fabric“ diegimo variantais. Leidimais paremtas valdymas suteikia galimybę autorizuotai administruoti daiktų interneto įrenginius, valdyti

prieigą prie jų. Blokų grandinės privatumas yra būtinas daiktų interneto įrenginių prisijungimo duomenims apsaugoti. Daiktų interneto įrenginių autentifikacijos metodo plėtrą taip pat palengvina didesnis „Fabric“ išmaniųjų kontraktų programavimo kalbų pasirinkimas, kurių tarpe yra plačiai naudojamos programavimo kalbos kaip „Go“, „Java“ ir kita.

„Ethereum“ taip pat yra lankstus ir įvarius taikymo atvejus turintis sprendimas. Tačiau „Ethereum“ siekia visiško skaidrumo ir nenaudoja leidimais paremto valdymo. Toks funkcionalumas neigiamai veikia spartą, plečiamumą ir privatumą. Išmaniųjų kontraktų realizavimui naudojama specialiai „Ethereum“ sukurta objektinio programavimo kalba „Solidity“. Tad diegiant daiktų interneto įrenginių autentifikacijos metodą naudojant „Ethereum“, papildomo sudėtingumo galimai suteikia papildomos programavimo kalbos mokymasis.

„Corda“ kaip ir „Fabric“ užtikrina privatų ir leidimais paremtą blokų grandinių valdymą (1.12 pav.).



1.12 pav. „R3 Corda“ architektūra [52]

„Corda“ išmaniųjų kontraktų kūrimas taip pat galimas plačiai naudojamomis programavimo kalbomis. Tačiau lyginant su kitais karkasais, „Corda“ yra specializuotas finansinėms operacijoms vykdyti.

1.8. Siekiamo sprendimo apibrėžimas

Tyrimo metu siekiama suformuoti sprendimą, kuris panaudotų blokų grandinės mechanizmą savarankiškai autentifikacijai tarp daiktų interneto įrenginių. Autentifikacijos metodo realizacijai naudojama MQTT protokolą realizuojanti programinė įranga, o blokų grandinių realizacijai naudojamas „Hyperledger Fabric“ karkasas.

Daiktų tarpusavio komunikacijai realizuojamas MQTT protokolo tinkamumą apibrėžia tokie faktoriai:

1) galimybė autentifikuoti ir autorizuoti įrenginius;
2) protokolas pritaikytas ribotus išteklius turintiems įrenginiams;
3) suteikiamas mechanizmas asinchroniškai komunikacijai.

Daiktų interneto įrenginių autentifikacijos užklausose nurodyti prisijungimo duomenys bus tikrinami blokų grandinėje patalpintais duomenimis. Tokį sprendimą efektyviausia realizuoti su „Hyperledger Fabric“, kadangi šis karkasas sukuria privatų, leidimais paremtą ir modulinį blokų grandinių tinklą.

Lyginant su kitais daiktų interneto autentifikacijos metodais, sprendimas turėtų išsiskirti šiais aspektais:

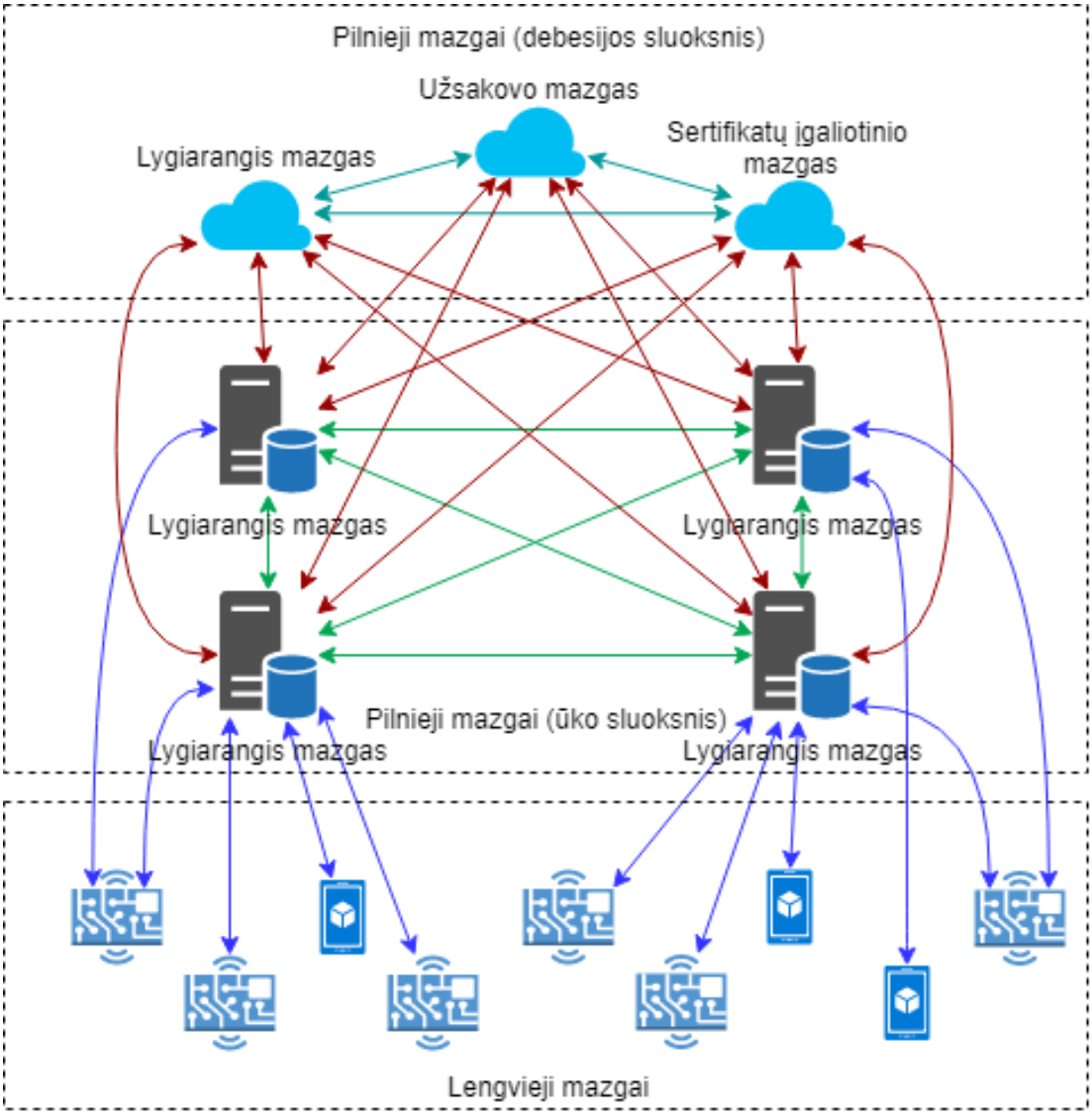
1) privatumo išsaugojimas;
2) veikimas suvaržytoje aplinkoje;
3) negalėjimo išsižadėti (angl. non-repudation) užtikrinimas.

1.9. Analizės išvados

Analizės metu apžvelgta daiktų interneto naudotojai, plačiai naudojami protokolai bei metodai. Atsižvelgiant į daiktų interneto pažeidžiamumų taksonomiją, išskirti svarbiausi pažeidžiamumai parodė kritinę autentifikavimo situaciją. Tolimesnė analizės dalis pateikia blokų grandinės technologijos analizę bei jos panaudojimo atvejus daiktų interneto įrenginių autentifikavimui atlikti. Analizės metu suformuluotas galimas blokų grandinės technologijos pritaikymas daiktų interneto įrenginių autentifikavimui realizuoti. Siekiamas sprendimas paremtas gerąja kitų tyrėjų praktika realizuojant panašius modelius, šiuo atveju suderinant blokų grandinių karkasą „Hyperledger Fabric“ su plačiai naudojamu daiktų interneto protokolu – MQTT.

2. DAIKTŲ INTERNETO ĮRENGINIŲ AUTENTIFIKACIJOS REALIZAVIMAS PANADOJANT BLOKŲ GRANDINES

Šiame skyriuje pateikiami modeliai ir metodika siekiamo sprendimo realizavimui. Realizacija atliekama trijose ūko kompiuterijos architektūros sluoksniuose, kurių išpildymas pateiktas 2.1 pav.

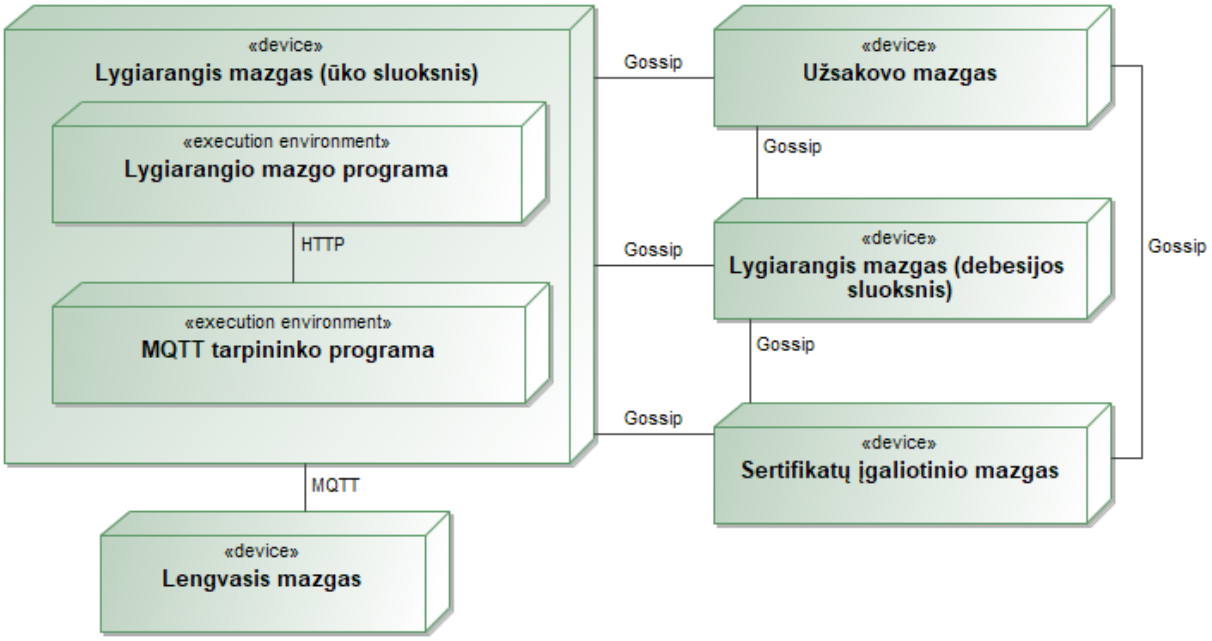


2.1 pav. Blokų grandinės panaudojimas ūko kompiuterijos architektūroje

Apatiniame sluoksnyje pateikti suvaržytas aplinkas turintys lengvieji mazgai. Lengvieji mazgai geba išsiųsti ar gauti duomenis ūko sluoksnyje esančių lygiarangių mazgų dėka, kurie atlieka sudėtingas blokų grandinių operacijas. Lygiarangiai mazgai turi tą pačia blokų grandinės kopiją, kuri gali būti papildoma tik esant ryšiui su debesijos sluoksnio mazgais. Tačiau net ir dingus ryšiui su debesijos sluoksniu, skaitymas iš blokų grandinės išlieka įmanomas ir lengvųjų mazgų veikla nenutraukiama.

2.1. Daiktų interneto įrenginių autentifikacijos blokų grandinėje diegimo modelis

Daiktų interneto įrenginių autentifikacijos blokų grandinėje diegimo modelis parodo kiekvieno sluoksnio sandarą ir ryšius (2.2 pav.).



2.2 pav. Daiktų interneto įrenginių autentifikacijos blokų grandinėje diegimo diagrama

Lengvasis mazgas. Įrenginys komunikuoja su MQTT tarpininko programa tradiciniais MQTT protokolo mechanizmais be jokių papildomų modifikacijų.

Lygiarangis mazgas (ūko sluoksnis). Šiame mazge diegiamos dvi programos: MQTT tarpininkas ir lygiarango mazgo programa. MQTT tarpininko programa naudojasi lygiarangio mazgo programa duomenų nuskaitymui iš blokų grandinės. Duomenys nuskaitomi atliekant HTTP užklausas. Lygiarangiai mazgai yra esminiai blokų grandinės elementai, kadangi jie talpina suvestines ir išmaniuosius kontraktus [53].

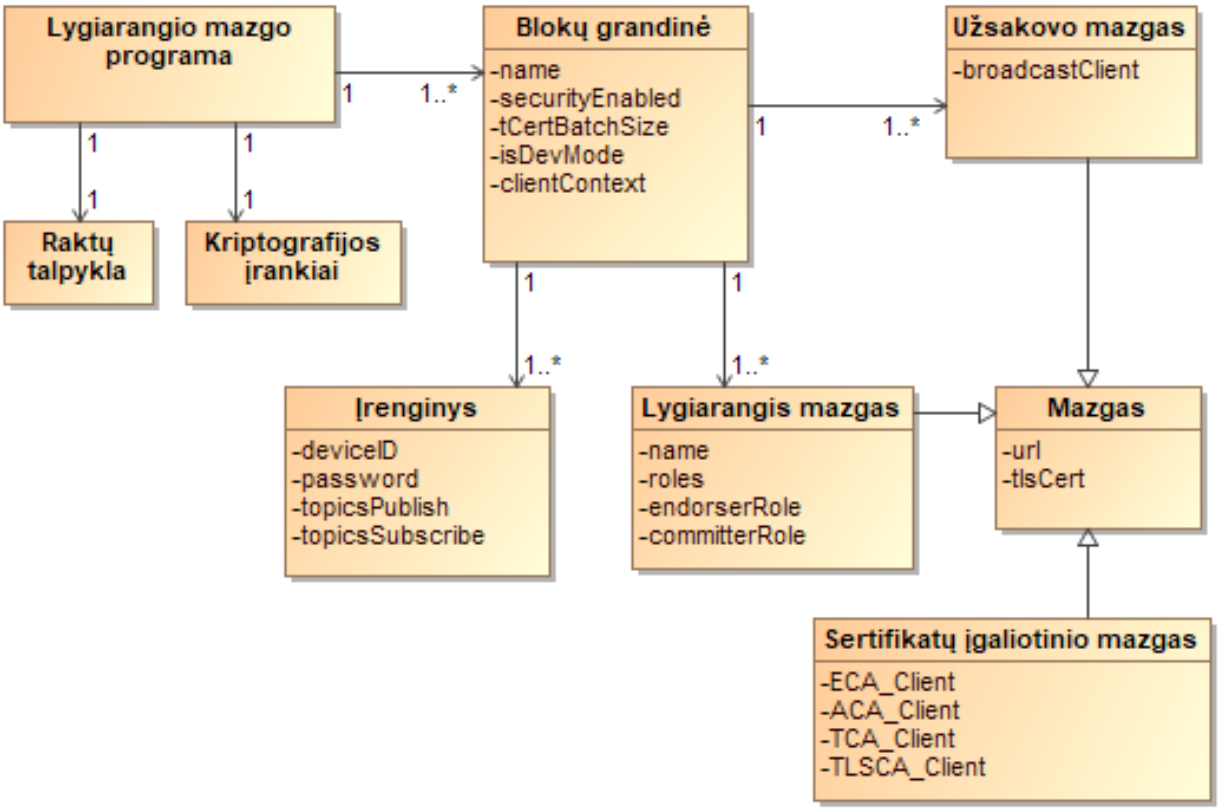
Lygiarangis mazgas (debesijos sluoksnis). Šis lygiarangis mazgas skiriasi nuo ūko sluoksnio varianto tuo, kad jis neturi MQTT tarpininko programos. Debesijos sluoksnio variante naudojama tik lygiarangio mazgo programa. Lygiarangiai mazgai debesijos sluoksnyje naudojami kaip pagrindiniai. Šie mazgai taip pat užtikrina, kad ūko sluoksnis visada turėtų atsarginę blokų grandinės kopiją. Komunikacijai su kitais mazgais naudojamas Gossip protokolas [54].

Užsakovo mazgas. Debesijos sluoksnyje esantis mazgas, atsakingas už transakcijų patvirtinimą [55]. Blokų grandinė lygiarangiuose mazguose negali būti papildoma be šio mazgo patvirtinimo.

Sertifikatų įgaliotinio mazgas. Sertifikatų įgaliotinis yra debesijos sluoksnio elementas, atsakingas už mazgų tapatybės išdavimą ir patvirtinimą [56].

2.2. Koncepcinis duomenų modelis

Koncepcinį duomenų modelį (2.3 pav.) sudaro įrenginių autentifikacijai pritaikyti blokų grandinių kūrimo ir valdymo elementai, naudojami programinėje įrangoje [57].



2.3 pav. Koncepcinio duomenų modelio diagrama

Raktų talpykla ir **kriptografijos įrankiai**. Šios duomenų esybės reikalingos kriptografinių procesų realizacijai.

Mazgas. Blokų grandinės mazgo apibendrinimas, turinti tris specializacijas: lygiarangis mazgas, užsakovo mazgas, sertifikatų įgaliotinio mazgas. Mazgo bazinį duomenų rinkinį sudaro mazgo adresas „url“ ir TLS sertifikatas „tlsCert“.

Užsakovas. Duomenų esybė nusako blokų grandinės elementą, atsakingą už transakcijų galutinį patvirtinimą (pasirašymą ir įtraukimą į blokų grandinę). Užsakovo pagrindinis komponentas „broadcastClient“ yra atsakingas už blokų grandinės atnaujinimo ištransliavimą visiems blokų grandinės dalyviams.

Sertifikatų įgaliotinis. Duomenų esybė atsakinga už mazgų tapatybių valdymą. Ją sudaro keturi esminiai komponentai, pateikti 2.1 lentelė.

2.1 lentelė. Sertifikatų įgaliotinio atributai

Atributas	Aprašas
ECA_Client	Registracijos sertifikatų įgaliotinis, kuris išskiria registracijos sertifikatus (ECert). ECerts ilgalaikiai sertifikatai naudojami identifikuoti blokų grandinės tinkle dalyvaujantiems mazgams.
ACA_Client	Atributų sertifikatų įgaliotinis, skirstantis sertifikatus atributų šifravimui.
TCA_Client	Transakcijų sertifikatų įgaliotinis skirsto transakcijų sertifikatus ECert sertifikatų savininkams.
TLSCA_Client	TLS sertifikatų įgaliotinis skirsto TLS sertifikatus sistemoms, kurios transliuoja žinutes grandinės tinkle.

Lygiarangis mazgas. Specializuotas mazgas, kuris talpina blokų grandinės kopiją. Lygiarangiai mazgai gali papildomai turėti specialias roles. Lygiarangį mazgą aprašantys atributai pateikti 2.2 lentelė.

2.2 lentelė. Lygiarangio mazgo atributai

Atributas	Aprašas
name	Mazgo pavadinimas.
roles	Mazgo rolės.
endorserRole	Būsena nurodanti, ar mazgas atlieka patvirtintojo rolę.
comitterRole	Būsena nurodanti, ar mazgas atlieka fiksuotojo rolę.

Kiekviename blokų grandinės kode galima nurodyti patvirtinimo politiką, nuo kurios priklauso patvirtintojo rolę turintys lygiarangiai mazgai. Tik patvirtintos transakcijos yra fiksuojamos už tai atsakingų lygiarangių mazgų.

Blokų grandinė. Ši duomenų esybė yra viso blokų grandinės veikimo branduolys. Čia atliekamos užklausos ir transakcijos, pridedami ir ištrinami blokų grandinėje dalyvaujantys mazgai bei atliekami kiti procesai. Blokų grandinėje šiame projekte talpinami registruotų lengvųjų mazgų duomenys (esybė **Įrenginys**). Blokų grandinės atributai aprašyti 2.3 lentelė

2.3 lentelė. Blokų grandinės atributai

Atributas	Aprašas
name	Blokų grandinės pavadinimas.
securityEnabled	Būsena, nusakanti ar naudojami saugos mechanizmai.
tCertBatchSize	Kiek „tCert“ sertifikatų nuskaityti per vieną paketą.
isDevMode	Būsena, nusakanti ar reikia naudoti kūrimo rėžimą.
clientContext	Kliento klasės atmaina, kuri jau buvo inicializuota.

Įrenginys. Blokų grandinėje talpinami duomenys, aprašantys įrenginio autentifikacijai ir autorizacijai reikalingus duomenis (2.4 lentelė).

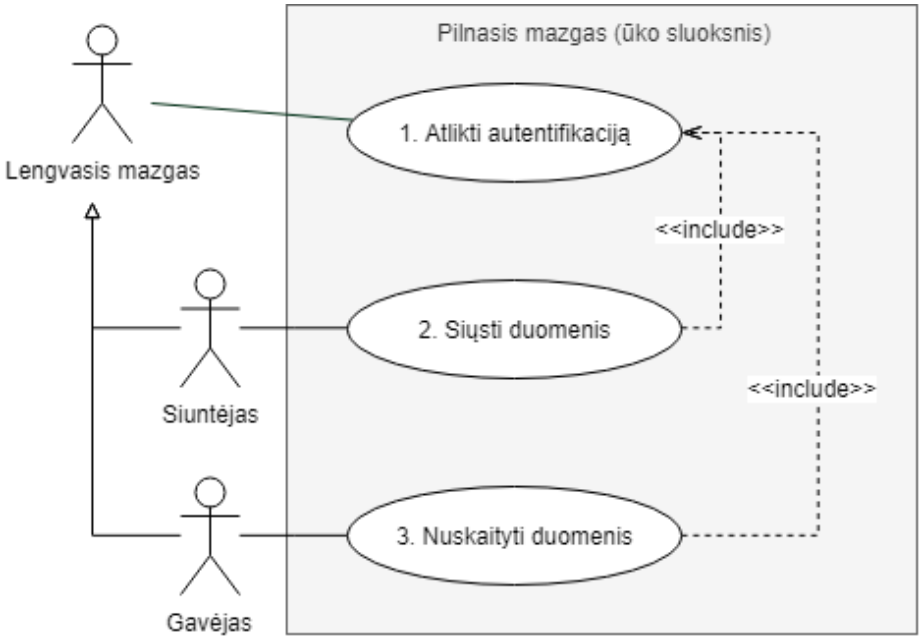
2.4 lentelė. Įrenginio duomenys blokų grandinėje

Atributas	Tipas	Pavyzdys	Aprašas
deviceID	String	„myDevice1“	Įrenginio identifikatorius.
password	String	„IGZhY2ExYTc5YThkZWNiYjgzZTRjNDE1ZGE1YWMwM2FlNzkzNWIwZWQ=„	Įrenginio slaptažodis.
topicsPub	String[]	[„/building1/lights/corridor/“, „/building1/lights/entrance/front“]	Temos, kuriomis įrenginys gali pranešti duomenis.
topicsSub	String[]	[„/building1/sensors/movement“]	Temos, kuriomis įrenginys gali prenumeruotis.

2.3. Daiktų interneto įrenginių reikalavimų modelis

Daiktų interneto įrenginių reikalavimų modelyje išskiriami du pagrindiniai aktoriai: lengvasis mazgas ir ūko sluoksnio pilnasis mazgas. Lengvojo mazgo funkcionalumą paveldi du specializuoti

aktoriai turintys po savo funkciją: duomenų siuntimas ir duomenų gavimas (2.4 pav.). Abi funkcijos pirmiausia vykdo autentifikacijos funkciją, be kurios šių funkcijų veikimas nėra galimas.



2.4 pav. Lengvojo mazgo įrenginio panaudos atvejų diagrama

2.5 lentelė. Panaudos atvejo „atlikti autentifikaciją“ aprašymas

ID	1
Pavadinimas	Atlikti autentifikaciją
Aprašymas	Lengvasis mazgas bando autentifikuotis, išsiųsdamas užklausą į jam nurodytą pilnąjį mazgą ir sulaukia atsakymo.
Aktoriai	Lengvasis mazgas, siuntėjas, gavėjas
Pradinės sąlygos	Lengvasis mazgas bando atlikti veiksmą, kuris reikalauja autentifikacijos.
Pagrindiniai žingsniai	Siunčiama autentifikacijos užklausa. Prisijungimo duomenys tikrinami blokų grandinėje. Grąžinamas sėkmingos autentifikacijos rezultatas.
Išskirtinės situacijos	Antrame žingsnyje prisijungimo duomenys netinkami ir grąžinamas klaidos kodas.
Galutinės sąlygos	Sėkmingai atlikta autentifikacija.

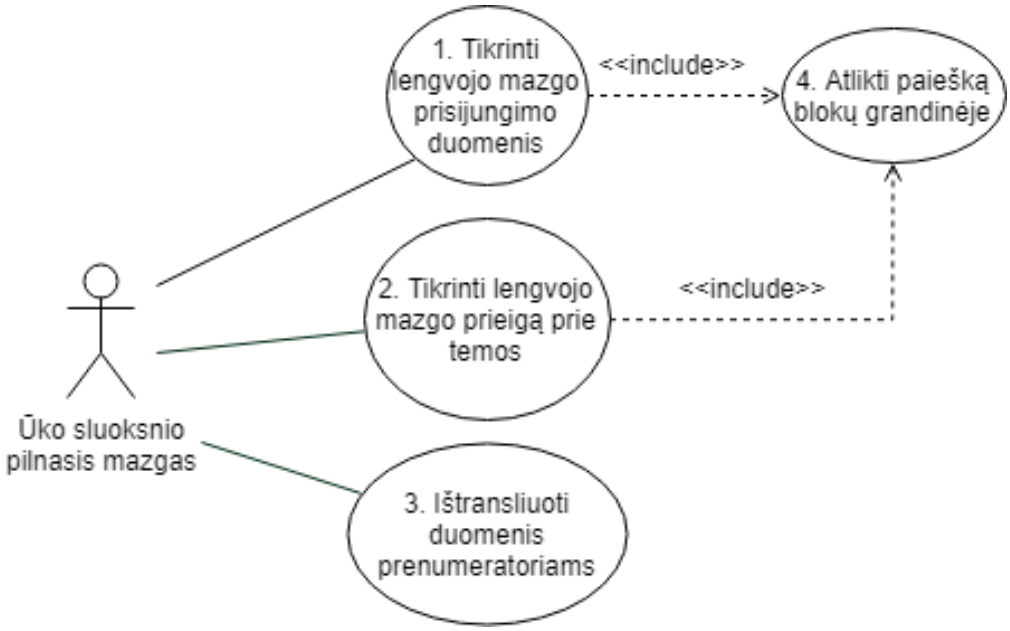
2.6 lentelė. Panaudos atvejo „siųsti duomenis“ aprašymas

ID	2
Pavadinimas	Siųsti duomenis
Aprašymas	Siuntėjas išsiunčia duomenis nurodyta tema per pilnąjį mazgą.
Aktoriai	Siuntėjas
Pradinės sąlygos	Siuntėjas paruošęs duomenis siuntimui į nurodytą temą.
Pagrindiniai žingsniai	Siuntėjas atlieka autentifikaciją. Siuntėjas siunčia duomenis į tam tikrą temą. Tikrinama siuntėjo prieiga prie nurodytos temos. Duomenys ištransliuojami temą prenumeruojantiems gavėjams.
Išskirtinės situacijos	Antrame arba trečiame žingsnyje nutrūkus autentifikuotai sesijai bandoma prisijungti iš naujo. Trečiame žingsnyje siuntėjui neturint prieigos prie nurodytos temos, grąžinamas klaidos kodas.
Galutinės sąlygos	Duomenys sėkmingai paskelbti.

2.7 lentelė. Panaudos atvejo „nuskaityti duomenis“ aprašymas

ID	3
Pavadinimas	Nuskaityti duomenis
Aprašymas	Gavėjas nuskaito siuntėjo paskelbtus duomenis.
Aktoriai	Gavėjas
Pradinės sąlygos	Gavėjas ruošiasi klausytis nurodyta tema skelbiamų duomenų.
Pagrindiniai žingsniai	Gavėjas atlieka autentifikaciją. Tikrinama gavėjo prieiga prie temos. Gavėjas laukia, kol nurodyta tema bus paskelbti duomenys. Gavėjas gauna duomenis, paskelbtus nurodyta tema.
Išskirtinės situacijos	Antrame arba trečiame žingsnyje nutrūkus autentifikuotai sesijai bandoma prisijungti iš naujo. Antrame žingsnyje lengvajam mazgui neturint prieigos prie nurodytos temos grąžinamas klaidos kodas.
Galutinės sąlygos	Gauti duomenys apdorojami.

Ūko sluoksnio pilnieji mazgai neturi specializacijų ir yra vienodi pagal funkcionalumą (2.5 pav.).



2.5 pav. Ūko sluoksnio pilnojo mazgo panaudos atveju diagrama

2.8 lentelė. Panaudos atvejo „tikrinti lengvojo mazgo prisijungimo duomenis“ aprašymas

ID	1
Pavadinimas	Tikrinti įrenginio prisijungimo duomenis
Aprašymas	Ūko sluoksnio pilnasis mazgas patikrina ar lengvojo mazgo prisijungimo duomenys egzistuoja blokų grandinėje.
Aktoriai	Ūko sluoksnio pilnasis mazgas
Pradinės sąlygos	Ūko sluoksnio pilnasis mazgas laukia autentifikacijos užklausų iš lengvųjų mazgų.
Pagrindiniai žingsniai	Gaunama lengvojo mazgo autentifikacijos užklausa. Ieškoma lengvojo mazgo duomenų blokų grandinėje. Lyginami autentifikacijos užklausoje ir blokų grandinėje rasti duomenys. Grąžinama sėkmingos autentifikacijos rezultatas.
Išskirtinės situacijos	Antrame žingsnyje blokų grandinėje neradus jokių įrašų apie lengvąjį mazgą grąžinamas klaidos kodas. Trečiame žingsnyje nesutapus duomenims grąžinamas klaidos kodas.
Galutinės sąlygos	Lengvojo mazgo įrenginys sėkmingai autentifikuotas.

2.9 lentelė. Panaudos atvejo „tikrinti lengvojo mazgo prieigą prie temos“ aprašymas

ID	2
Pavadinimas	Tikrinti lengvojo mazgo prieigą prie temos
Aprašymas	Ūko sluoksnio pilnasis mazgas patikrina ar lengvasis mazgas turi prieigą prie nurodytos temos.
Aktoriai	Ūko sluoksnio pilnasis mazgas
Pradinės sąlygos	Ūko sluoksnio pilnasis mazgas laukia siuntimo arba prenumeravimo užklausų iš lengvųjų mazgų.
Pagrindiniai žingsniai	Gaunama siuntimo arba prenumeravimo užklausa nurodyta tema. Ieškoma lengvojo mazgo duomenų blokų grandinėje. Tikrinama ar blokų grandinėje įrašyta prieiga prie nurodytos temos.
Išskirtinės situacijos	Antrame žingsnyje neradus lengvojo mazgo duomenų grąžinamas klaidos kodas. Trečiajame žingsnyje blokų grandinėje esančiuose lengvojo mazgo duomenyse neradus nurodytos temos grąžinamas klaidos kodas.
Galutinės sąlygos	Lengvajam mazgui leista atlikti norima veiksmą.

2.10 lentelė. Panaudos atvejo „ištransliuoti duomenis prenumeratoriams“ aprašymas

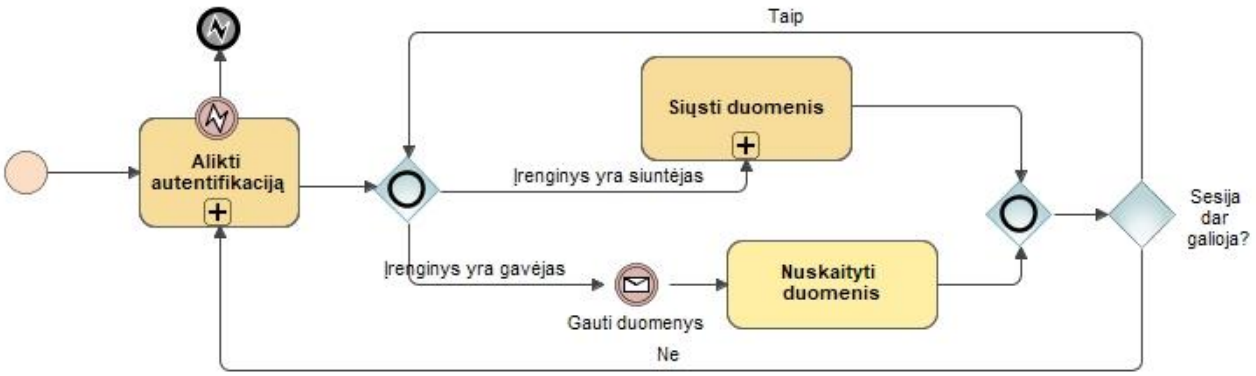
ID	3
Pavadinimas	Ištransliuoti duomenis prenumeratoriams
Aprašymas	Ūko sluoksnio pilnasis mazgas ištransliuoja gautus duomenis temą prenumeruojantiems gavėjams.
Aktoriai	Ūko sluoksnio pilnasis mazgas
Pradinės sąlygos	Ūko sluoksnio pilnasis mazgas laukia siuntimo užklausų iš lengvųjų mazgų.
Pagrindiniai žingsniai	Lengvasis mazgas paskelbia duomenis nurodyta tema. Tikrinama lengvojo mazgo prieiga prie temos. Duomenys ištransliuojami prieigą prie temos turintiems prenumeratoriams.
Išskirtinės situacijos	Antrajame žingsnyje siuntėjui neturint prieigos prie nurodytos temos jam grąžinamas klaidos kodas.
Galutinės sąlygos	Temą prenumeruojantys gavėjai gauna siuntėjo paskelbtus duomenis.

2.11 lentelė. Panaudos atvejo „atlikti paiešką blokų grandinėje“ aprašymas

ID	4
Pavadinimas	Atlikti paiešką blokų grandinėje
Aprašymas	Ūko sluoksnio pilnasis mazgas ištransliuoja gautus duomenis temą prenumeruojantiems gavėjams.
Aktoriai	Ūko sluoksnio pilnasis mazgas
Pradinės sąlygos	Ūko sluoksnio pilnasis mazgas laukia siuntimo užklausų iš lengvųjų mazgų.
Pagrindiniai žingsniai	Lengvasis mazgas paskelbia duomenis nurodyta tema. Tikrinama lengvojo mazgo prieiga prie temos. Duomenys ištransliuojami prieigą prie temos turintiems prenumeratoriams.
Išskirtinės situacijos	Antrajame žingsnyje siuntėjui neturint prieigos prie nurodytos temos jam grąžinamas klaidos kodas.
Galutinės sąlygos	Temą prenumeruojantys gavėjai gauna siuntėjo paskelbtus duomenis.

2.4. Daiktų interneto įrenginių autentifikacija panaudojant blokų grandines veiklos proceso modelis

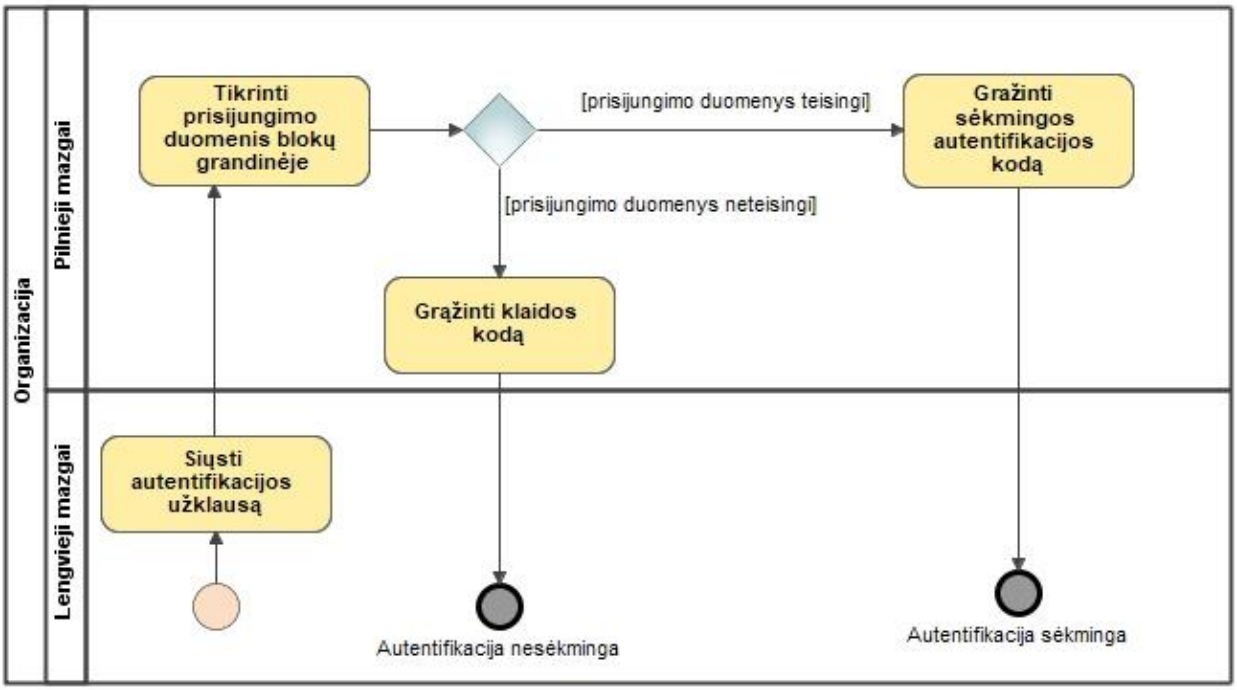
Lengvasis mazgas atlikdamas vieną iš dviejų veiksmų (siųsti arba nuskaityti duomenis) pirmiausia turi atlikti autentifikaciją, kadangi be autentifikuotos sesijos veiksmai negali būti vykdomi (2.6 pav.).



2.6 pav. Lengvojo mazgo proceso diagrama

Pasibaigus sesijos galiojimui lengvasis mazgas vėl privalo atlikti autentifikaciją prieš pradedant vykdyti tolimesnius veiksmus.

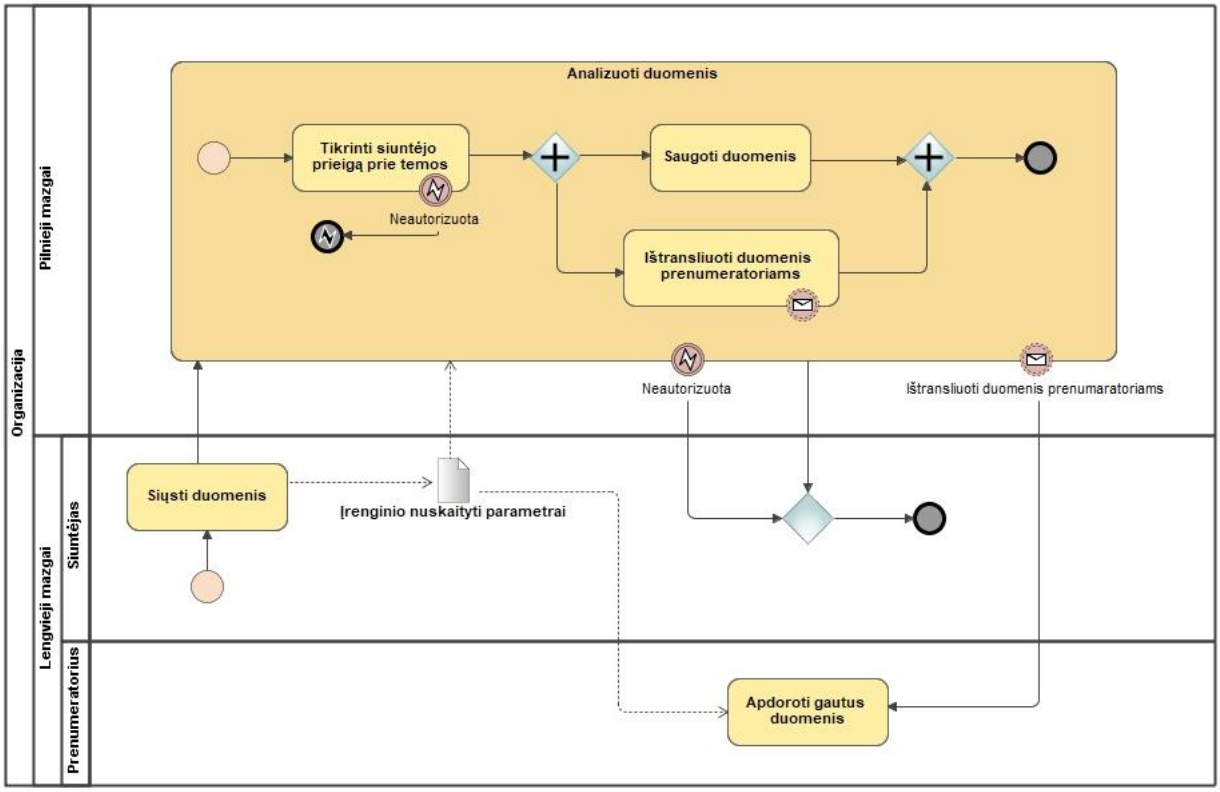
Autentifikacijos užklausa siunčiama į ūko sluoksnio pilnąjį mazgą ir prisijungimo duomenys tikrinami blokų grandinėje (2.7 pav.). Jeigu prisijungimo duomenys yra neteisingai arba įrenginys buvo nerastas blokų grandinėje, tokiu atveju grąžinamas klaidos kodas. Jeigu prisijungimo duomenys sutapo su duomenimis, patalpintais blokų grandinėje, tokiu atveju grąžinamas sėkmingos autentifikacijos kodas.



2.7 pav. Lengvojo mazgo antrinio proceso „atlikti autentifikaciją“ diagrama

Po sėkmingos autentifikacijos įrenginys gali vykdyti autentifikacijos reikalaujančius veiksmus. Autentifikuotos sesijos reikalauja du pagrindiniai lengvojo mazgo veiksmai: duomenų siuntimas ir duomenų nuskaitymas.

Žemiau pateiktas procesas (2.8 pav.) parodo veiksmų seką nuo duomenų išsiuntimo iki gavimo.

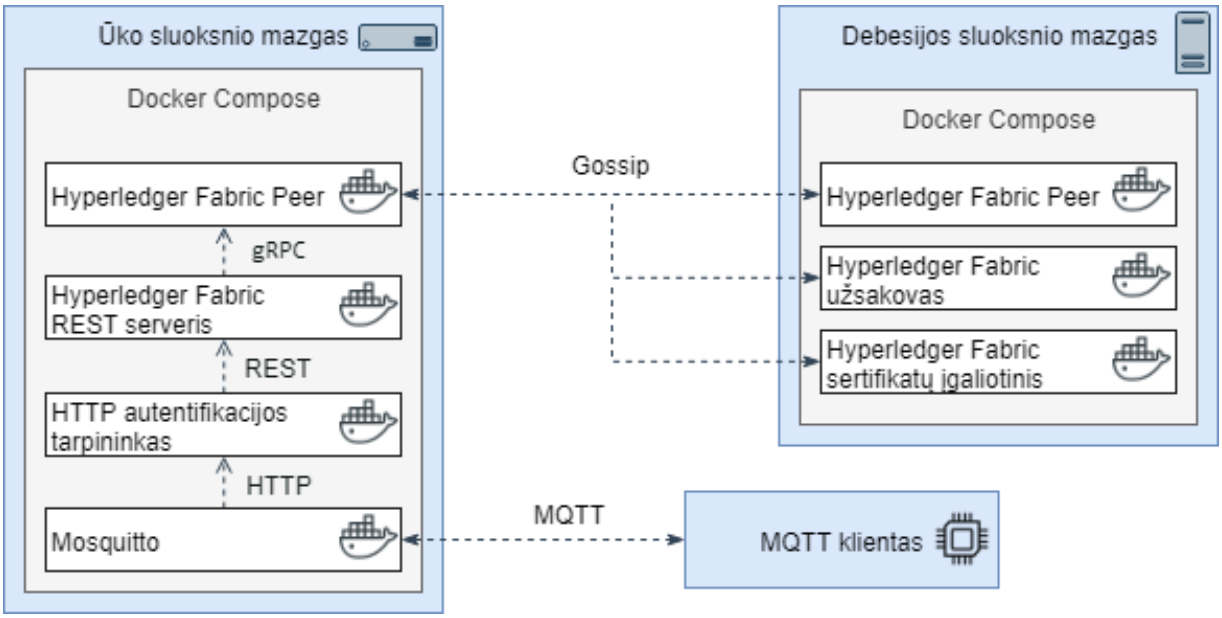


2.8 pav. Lengvojo mazgo antrinio proceso „siųsti duomenis“ diagrama

Siunčiant duomenis tikrinama ar lengvasis mazgas turi prieigą transliuoti nurodyta tema (pvz. „/tema/subtema1“). Sėkmingu atveju duomenys išsaugomi ir ištransliuojami temos prenumeratoriams.

3. BLOKŲ GRANDINĖS TECHNOLOGIJA PAREMTO DAIKTŲ INTERNETO ĮRENGINIŲ AUTENTIFIKACIJOS PROTOTIPO REALIZACIJA

MQTT susiejimas su „Hyperledger Fabric“ prototipo realizavime atliekamas panaudojant „Docker Compose“ konteinerių orkestravimo įrankį. Įrankyje aprašomi prototipo realizavimui reikalingi komponentai (įrankyje įvardinami kaip paslaugos), kaip pateikta 3.1 paveiksle.



3.1 Prototipo architektūra

Susiejimo prototipui naudojami tokie konteineriai:

1. **MQTT tarpininkas („Mosquitto“)**. Nustatytas naudoti HTTP autentifikacijos tarpininką kaip HTTP autentifikacijos šaltinį. MQTT tarpininko konteineris sukuriamas įrašant „Mosquitto“ programinę įrangą ir papildomą autentifikacijos modulį, kuris suteikia galimybę naudoti skirtingus MQTT įrenginių autentifikacijos metodus. Modulyje nustatomas autentifikacijos metodas HTTP, kuris autentifikuoja įrenginius į pasirinktą Web serverį. Web serveriu šiame prototipe yra pasirenkamas konteineris Proxy.
2. **HTTP autentifikacijos tarpininkas**. Tarpininkas tarp MQTT tarpininko ir „Hyperledger Fabric“ REST serverio. Šiame konteineryje talpinamas smulkus Web serveris, kurio vienintelė paskirtis yra perduoti iš „Mosquitto“ konteinerio ateinančias autentifikacijos užklausas į REST konteinerį reikiamu formatu.

3. **„Hyperledger Fabric“ REST serveris**. Blokų grandinės funkcionalumo pateikimas per REST sąsają. Konteineris REST priima iš HTTP autentifikacijos tarpininko konteinerio perduodamas autentifikacijos užklausas REST API formatu ir paverčia jas užklausomis į blokų grandinę.
4. **„MongoDB“**. Prisijungimo prie blokų grandinės ir kitų svarbių duomenų talpinimas. Komponentas reikalingas „Hyperledger Fabric“ REST sąsajai.

MQTT tarpininkas, HTTP autentifikacijos tarpininkas ir „Hyperledger Fabric“ REST konteineriai sujungiami į vieną lokalų tinklą. REST ir „MongoDB“ konteineriai taip pat prijungiami prie jau esamo blokų grandinės mazgo tinklo, per kurį galiausiai pasiekiama blokų grandinė.

3.1. Hyperledger Fabric diegimas

Blokų grandinės objektus aprašo žemiau pateiktas programinio kodo fragmentas (3.2 pav.), kuriame objektų reikšmės aprašytos komentaruose.

```
/**
 * Failas: org.iotfabric.network.cto
 *
 * Įrenginių autentifikacijos tinklas
 */
namespace org.iotfabric.network

// Dalyvio objektas, aprašantis įrenginio duomenis
participant Device identified by deviceID {
    o String deviceID // Įrenginio prisijungimo vardas
    o String password // Įrenginio prisijungimo slaptažodis
    o String[] topicsPub // Temos, kuriomis įrenginys gali pranešti
    o String[] topicsSub // Temos, kuriomis įrenginys gali prenumeruotis
}

// Įrenginio prieigos prie konkrečios temos keitimo transakcija
transaction TopicAccess {
    --> Device device // Įrenginys, kurio teisės keičiamos transakcijos metu
    o String topic // Tema, kurios prieiga bus keičiama
    o Boolean grant // true – suteikia prieigą, false – panaikina prieigą
    o Boolean type // true - pranešti, false - prenumeruoti
}
```

3.2 pav. Lengvojo mazgo duomenis aprašantis programinis kodas

Kadangi įrenginys yra aprašomas kaip dalyvis, jo kūrimas, atnaujinimas ir trynimas aprašomas automatiškai.

Todėl blokų grandinės logikoje papildomai aprašomas tik prieigos prie temų valdymas, vykdomas transakcijomis (3.3 pav.).

```
/**
 * Failas: Logic.js
 *
 * Ši transakcija atnaujina įrenginio teises į tam tikrą temą
 * @param {org.iotfabric.network.TopicAccess} topicAccess – tema kuri bus apdorojama
 * @transaction
 */
async function updateTopicAccess(topicAccess) {
    if (topicAccess.grant) {
        if (topicAccess.type){
            if (!topicExists(topicAccess.device.topicsPub, topicAccess.topic)) {
                topicAccess.device.topicsPub.push(topicAccess.topic);
            }
        } else {
            if (!topicExists(topicAccess.device.topicsSub, topicAccess.topic)) {
                topicAccess.device.topicsSub.push(topicAccess.topic);
            }
        }
    } else {
        if (topicAccess.type){
            var searchedTopic = topicExists(topicAccess.device.topicsPub,
topicAccess.topic);
            if (Number.isInteger(searchedTopic)) {
                topicAccess.device.topicsPub.splice(searchedTopic, 1);
            }
        } else {
            var searchedTopic = topicExists(topicAccess.device.topicsSub,
topicAccess.topic);
            if (Number.isInteger(searchedTopic)) {
                topicAccess.device.topicsSub.splice(searchedTopic, 1);
            }
        }
    }
    let participantRegistry = await
getParticipantRegistry('org.iotfabric.network.Device');
    await participantRegistry.update(topicAccess.device);
}
```

3.3 pav. Prieigos prie temos atnaujinimo logikos programinis kodas

Hyperledger Fabric paleidimas atliekamas panaudojant Hyperledger Composer įrankius. Įdiegus įrankius startuojama bandomoji aplinka.

Pirmiausia suarchyvuojamas aukščiau pavaizduotas programos kodas:

```
composer archive create -t dir -n .
```

Suarchyvuotas kodas įrašomas į Hyperledger Fabric aplinką kaip naujas tinklas, nurodant versijos numerį:

```
composer network install --card PeerAdmin@hlfv1 --archiveFile iot-fabric@[VERSIJOS_NUMERIS].bna
```

Startuojamas naujai sukurtas tinklas, kuriam nurodomi tinklo administratoriaus prisijungimo duomenys ir tinklo administratoriaus kortelės failas, kuris bus išsaugomas tolimesniam naudojimui:

```
composer network start --networkName iot-fabric --networkVersion [VERSIJOS_NUMERIS] --networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card
```

Sugeneruota tinklo administratoriaus kortelė importuojama į Hyperledger Fabric aplinką:

```
composer card import --file networkadmin.card
```

Nauja tinklo administratoriaus kortelė išbandoma:

```
composer network ping --card admin@iot-fabric
```

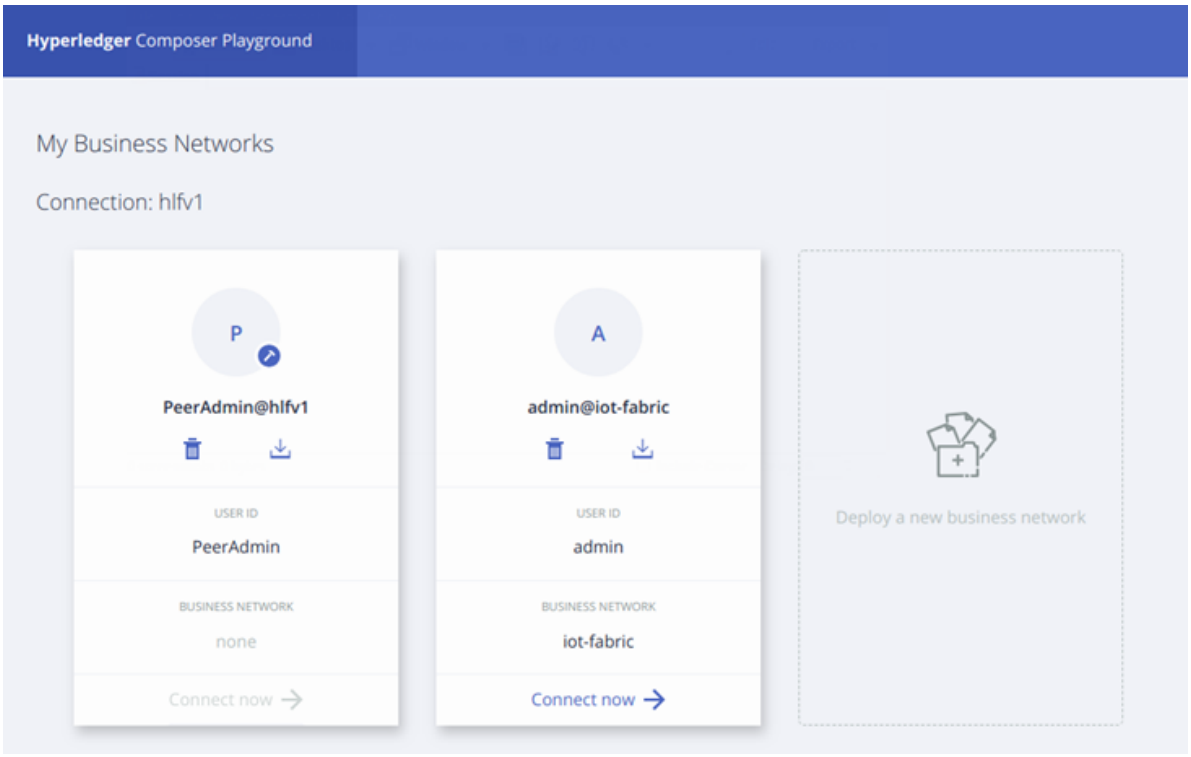
Blokų grandinės logika atnaujinama suarchyvavus naujesnę programos kodo versiją ir įdiegus ją, panaudojant tokias komandas:

```
composer archive create -t dir -n .
composer network install -a iot-fabric@[VERSIJOS_NUMERIS].bna -c PeerAdmin@hlfv1
composer network upgrade -c PeerAdmin@hlfv1 -n iot-fabric -V [VERSIJOS_NUMERIS]
```

Po Hyperledger Fabric bandomosios aplinkos išjungimo, pakartotiniam įjungimui reikalingos tokios komandos:

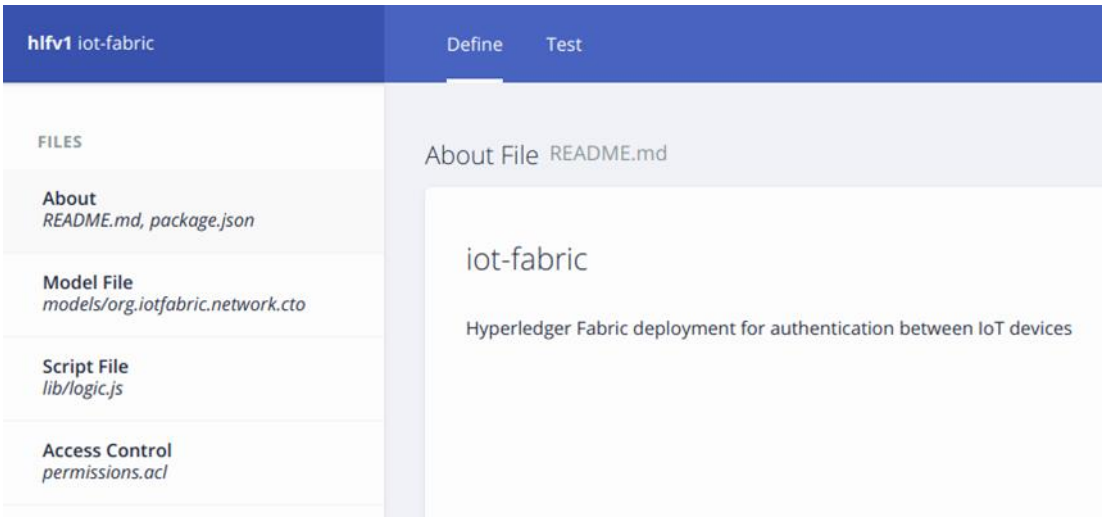
```
composer network install -a iot-fabric@[NAUJOS_VERSIJOS_NUMERIS].bna -c PeerAdmin@hlfv1
composer network start --networkName iot-fabric --networkVersion [NAUJOS_VERSIJOS_NUMERIS] --networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card
```

Sėkmingai atlikus tinklo diegimą, bandomosios aplinkos grafinėje sąsajoje matoma naujo tinklo administratoriaus kortelė (3.4 pav.).



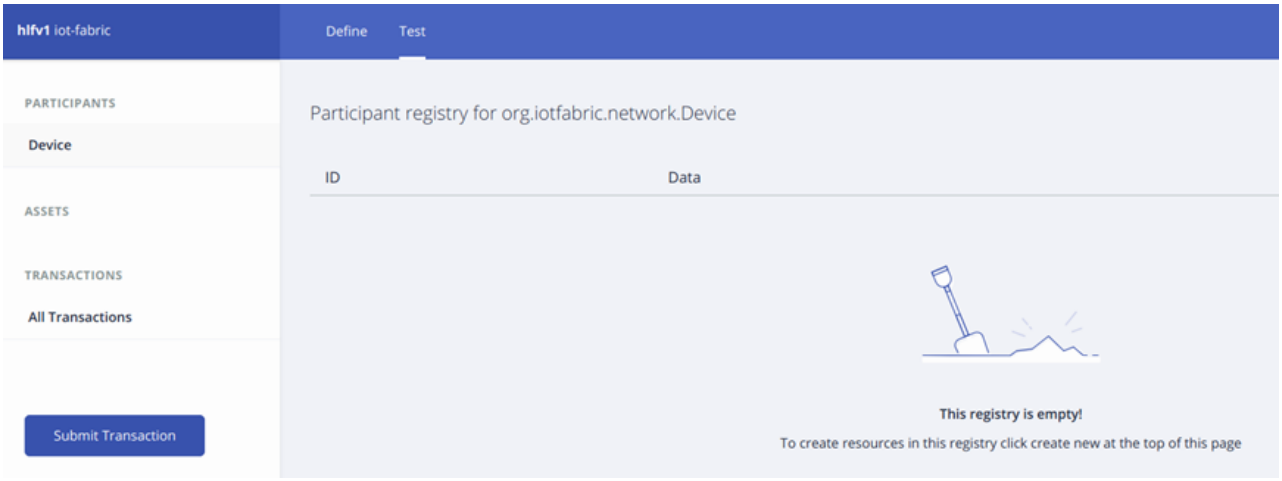
3.4 pav. Hyperledger Composer bandomosios aplinkos pagrindinis langas

Atidarius „admin@iot-fabric" tinklo administratoriaus kortelę matomi tinklo logiką nusakantys programinio kodo failai (3.5 pav.).



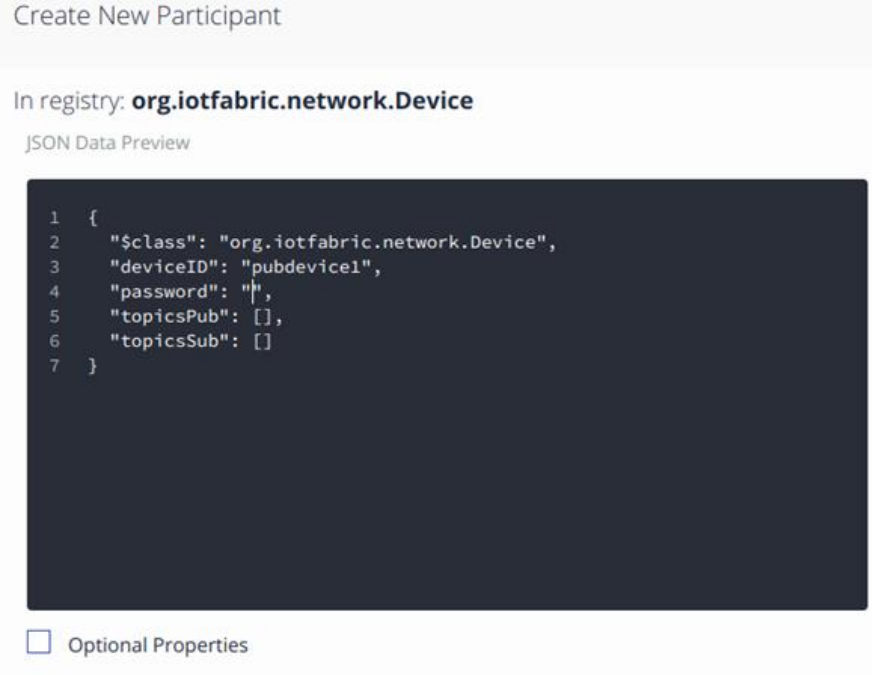
3.5 pav. „iot-fabric" tinklo informacijos langas

Lange paspaudus skiltį „Test" atidaromas aprašytos logikos testavimo puslapis (3.6 pav.).



3.6 pav. „iot-fabric“ tinklo testavimo langas

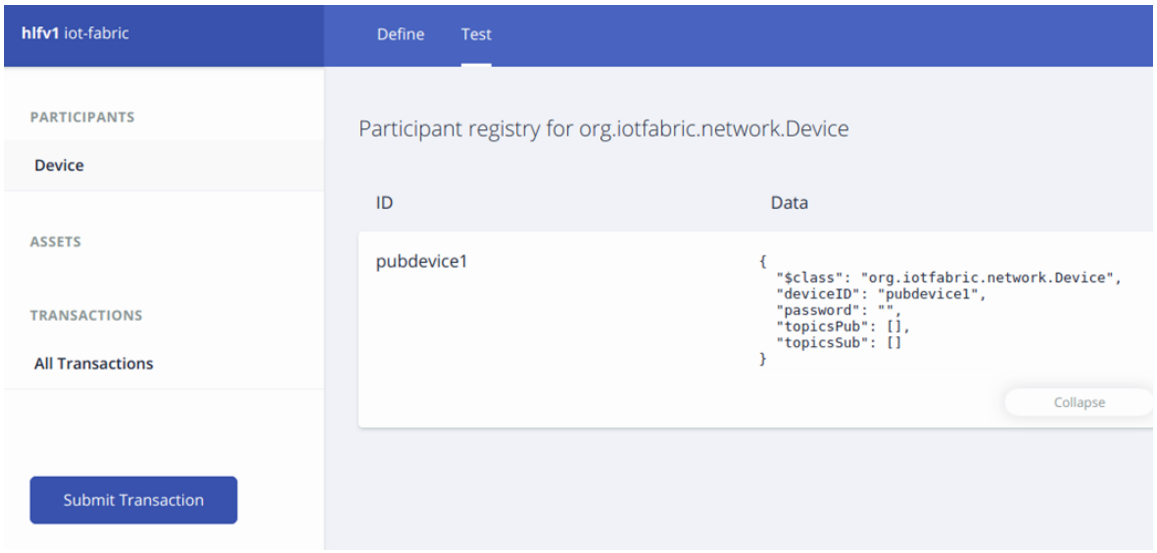
Testo pavyzdžiui realizuoti pirmiausia į blokų grandinę kaip tinklo dalyvis įtraukiamas naujas įrenginys. Tai atliekama bandomojoje aplinkoje paspaudus mygtuką „New participant“, kuris atidaro naujo dalyvio kūrimo langą (3.7 pav.).



3.7 pav. Naujo įrenginio įtraukimas į blokų grandinę

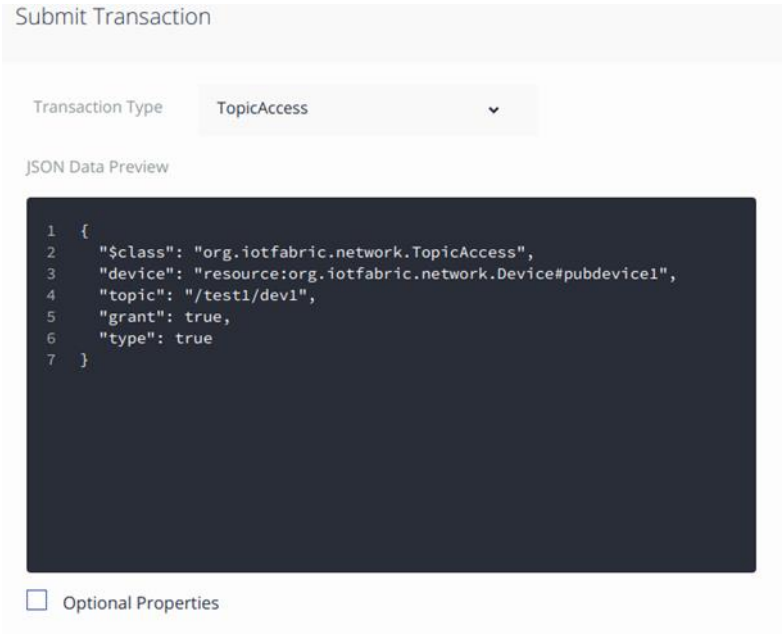
Naujo dalyvio kūrimo lange matomi ankščiau programiniame kode aprašyti atributai, nusakantys lengvojo mazgo autentifikacijai ir prieigos nustatymui reikalingus duomenis.

Užpildžius ir patvirtinus duomenis naujas dalyvis įtraukiamas į blokų grandinę ir jį galima pamatyti bendrame dalyvių sąraše (3.8 pav.).



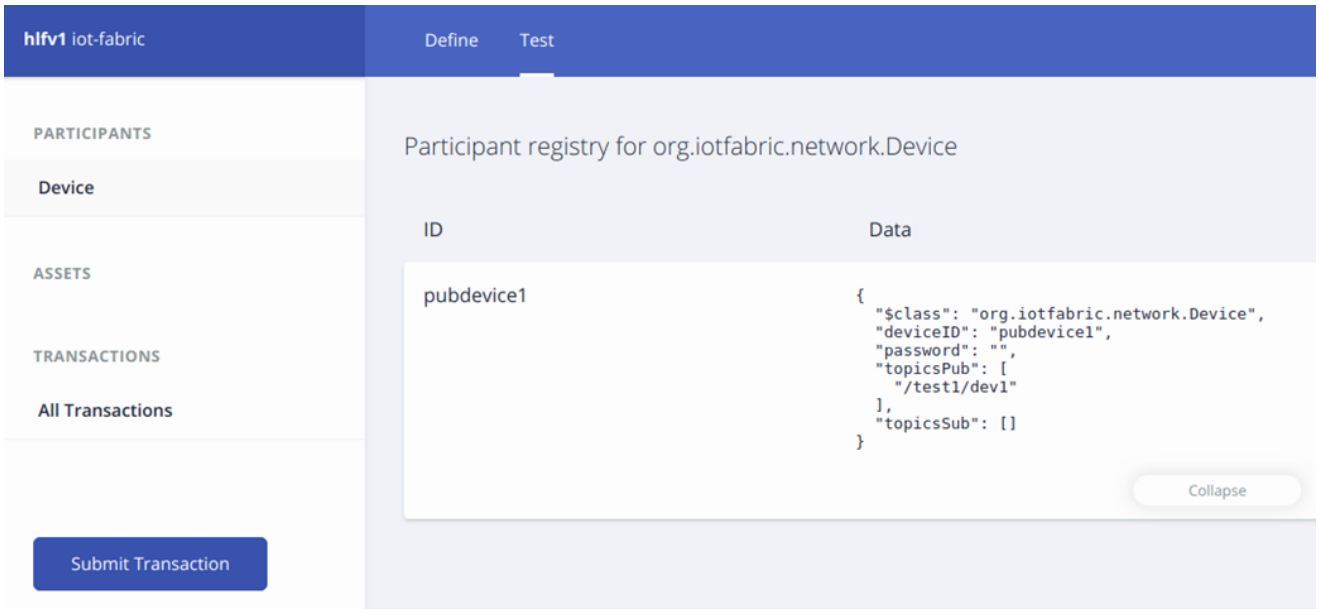
3.8 pav. „iot-fabric“ tinklo testavimo langas įtraukus naują įrenginį

Pavyzdyje pateiktas naujai sukurtas lengvasis mazgas neturi jokių prieigos teisių. Teisės prie tam tikros temos suteikiamos sukuriant blokų grandinės transakciją prieigai suteikti. Paspaudus mygtuką „Submit transaction“ atvaizduojama transakcijos pateikimo forma (3.9 pav.).



3.9 pav. Prieigos suteikimo transakcijos forma

Pateikus transakciją atliekama jos apdorojimo procesų seka ir galutinis pokytis matomas atsinaujinus dalyvių sąraše esančio įrenginio duomenims. Atnaujintame lange matoma įrenginio prieigos teises nusakanti informacija (3.10 pav.).



3.10 pav. Įrenginio duomenys po transakcijos

Įvykdžius transakciją, prieiga prie duomenų paskelbimo buvo papildyta transakcijos formoje nurodyta tema „/test1/dev1“.

4. BLOKŲ GRANDINĖS TECHNOLOGIJA PAREMTO DAIKTŲ INTERNETO ĮRENGINIŲ AUTENTIFIKACIJOS PROTOTIPO EKSPERIMENTINĖ ANALIZĖ

Šiame skyriuje pateikiama eksperimentais paremta projekte sukurto prototipo kiekybinė ir kokybinė analizė.

4.1. Eksperimente naudojama įranga

Eksperimentų aplinkai naudojamos septynios virtualios mašinos, kurių sistemos specifikacija ir paskirtis pateikta 4.1 lentelė.

4.1 lentelė. Eksperimente naudojamos virtualios mašinos

Pavadinimas	RAM (GB)	VCPUs (brand.)	Diskas (GB)	Paskirtis
PeerMain	8	4	10	Debesijos sluoksnio pilnasis mazgas. Jį sudaro blokų grandinės lygiarangio, užsakovo ir sertifikatų įgaliotinio mazgai.
Peer1	2	2	10	Ūko sluoksnio pilnasis mazgas.
Peer2	2	2	10	Ūko sluoksnio pilnasis mazgas.
Peer3	2	2	10	Ūko sluoksnio pilnasis mazgas.
Peer4	2	2	10	Ūko sluoksnio pilnasis mazgas.
Peer5	2	2	10	Ūko sluoksnio pilnasis mazgas.
ExternalDB	2	2	10	Išorinės duom. bazės „MySQL“ ir „MongoDB“

Lengvųjų mazgų simuliacija atlikta iš darbinio kompiuterio, kurio specifikacija yra:

1. 6 branduolių (12 gijų) 3.7 GHz procesorius;
2. 8 GB RAM;

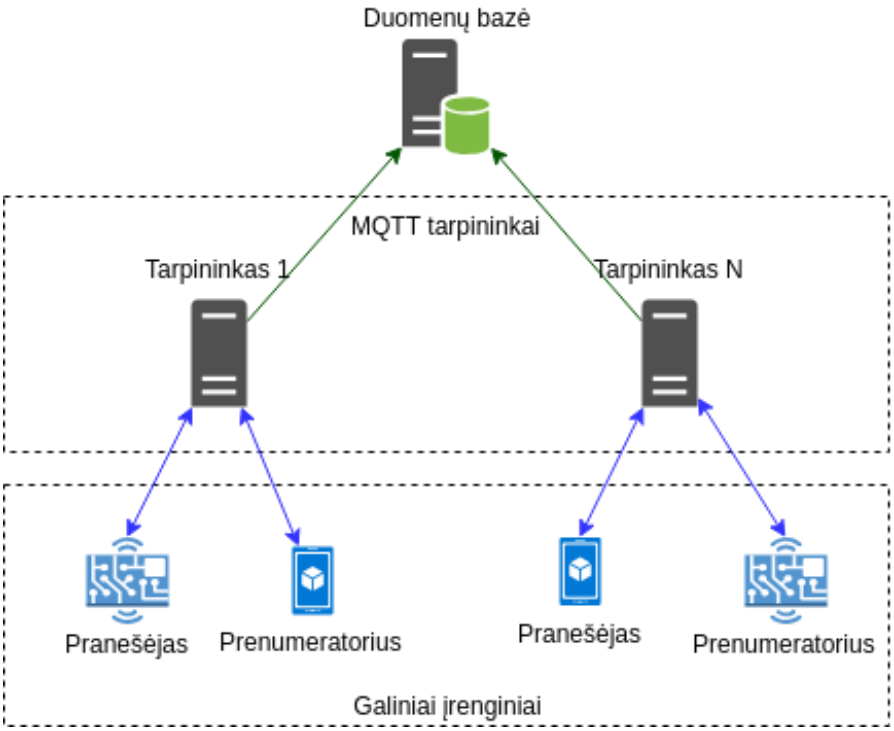
Lengvųjų mazgų simuliacijos metu buvo sekamas tinklo srautas, procesoriaus ir RAM apkrova. Minėti kompiuteriniai resursai nepasiekia savo limitų, todėl simuliacijos sparta nebuvo ribojama lengvųjų mazgų dalyje. Vienintelis ribojantis faktorius lengvųjų mazgų dalyje yra simuliuojamų lengvųjų mazgų skaičius, kuris realioje situacijoje atitinka branduolių skaičių. Tačiau simuliacijos ir realios situacijos skirtumas yra labai mažas, kadangi eksperimente kiekvienam simuliuojamam lengvajam mazgui išskiriama gija.

4.2. Blokų grandinės autentifikacijos prototipo kiekybinė analizė

Eksperimentinėje dalyje atliekami bandymai su blokų grandinės technologijos panaudojimu atliekant daiktų interneto įrenginių autentifikaciją. Bandymų metu modelis lyginamas su kitais MQTT autentifikacijos analogais:

1. failo autentifikacija;
2. išorinė SQL tipo duomenų bazė;
3. išorinė NoSQL tipo duomenų bazė;

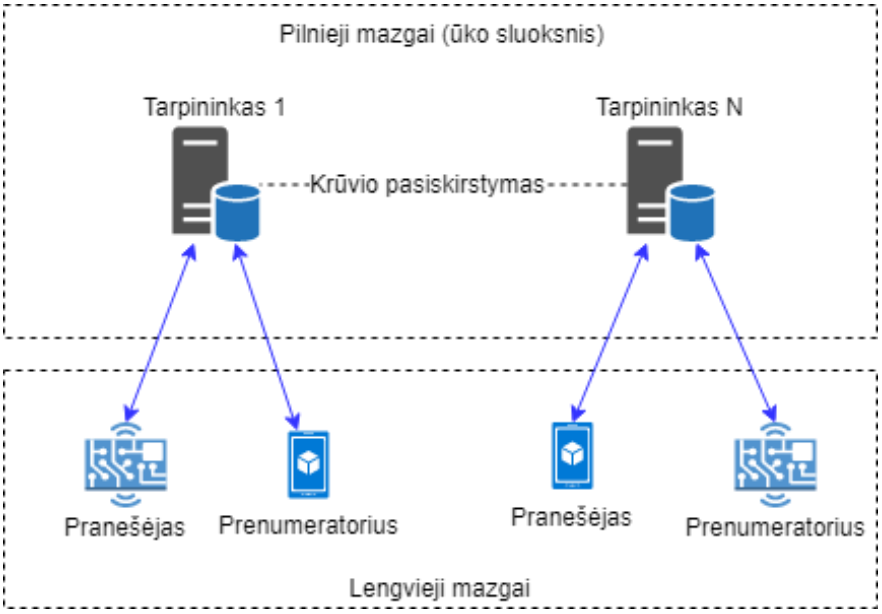
Išorinių duomenų bazių atveju visi tarpininkai kreipiasi į vieną tašką, tad jame sutelkiamas didžiausias srauto krūvis (4.1 pav.).



4.1 pav. MQTT veikimas autentifikacijai panaudojant išorinę duomenų bazę

Dėl šios priežasties numatoma, kad didėjant MQTT tarpininkų skaičiui ir prie jų besijungiančių galinių įrenginių skaičiui, lėtėja autentifikacijos užklausų vykdymo laikas. Lėtėjant autentifikacijos užklausų vykdymo laikui atitinkamai lėtėja duomenų paskelbimo laikas.

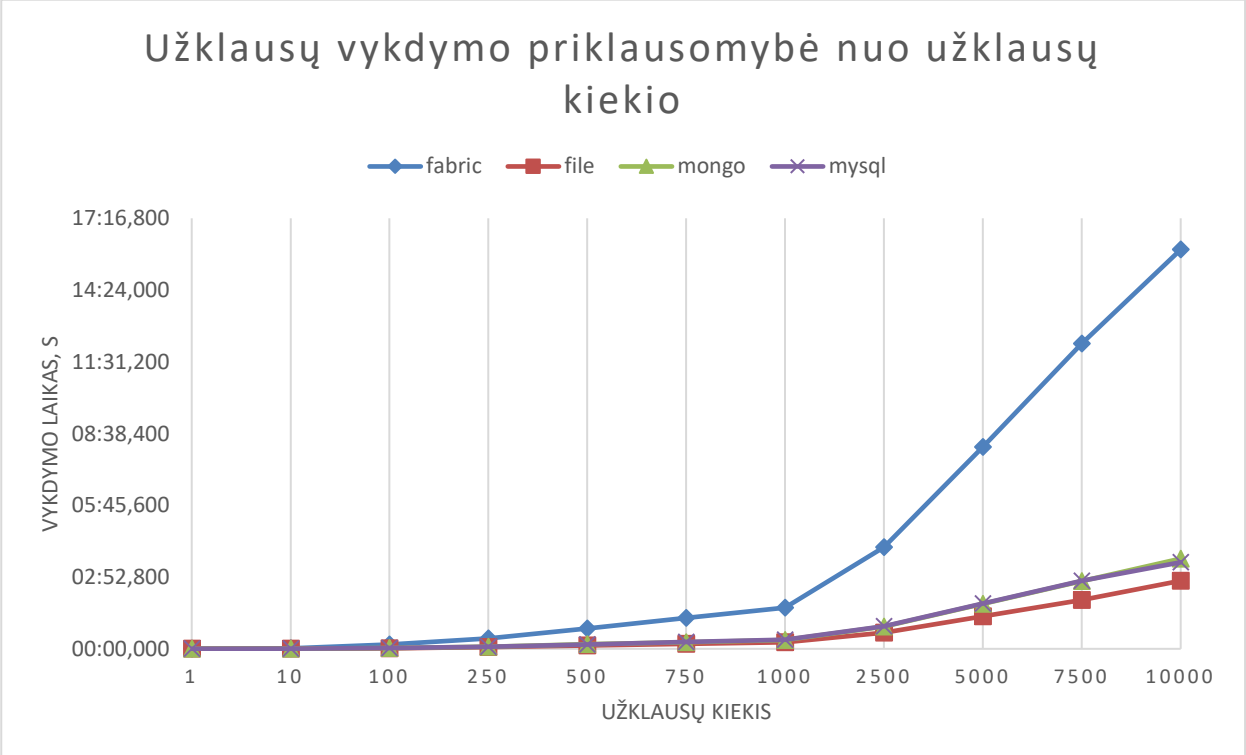
Blokų grandinės ir failo autentifikacijos atveju nėra kreipiamasi į išorinius duomenų šaltinius. Tokiu atveju krūvis yra paskirstomas atitinkamai per visus tarpininkus arba blokų grandinių atveju - pilnuosius ūko sluoksnio mazgus (4.2 pav.).



4.2 pav. MQTT veikimas autentifikacijai panaudojant blokų grandines

Kiekvienas ūko sluoksnio mazgas priima visą prie jo besijungiančių lengvųjų mazgų krūvį ir taip nesusidaro butelio kaklelio efektas. Tokia architektūra taip pat sudaro sąlygas lengvai plėsti infrastruktūrą. Jeigu viename ar keliuose ūko sluoksnio mazguose susidaro didelė apkrova, ūko sluoksnį galima išplėsti dar keliais pilnaisiais mazgais, kurie perimtų dalį krūvio.

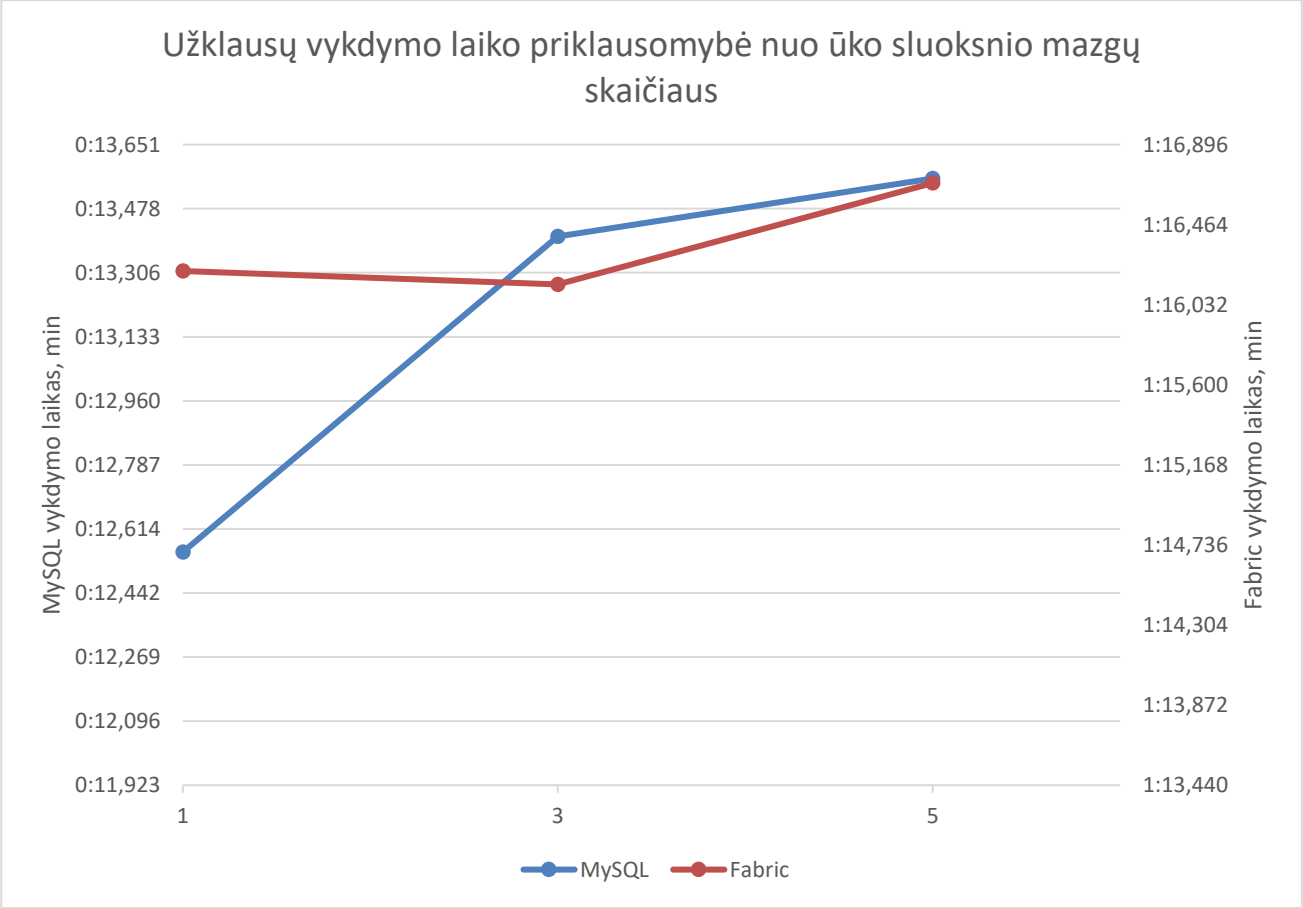
Žemiau pateiktame eksperimente (4.3 pav.) matuojama kaip pasikeičia pranešimų perdavimo laikas vietoje analogiškų autentifikacijos metodų panaudojant blokų grandinės technologiją.



4.3 pav. Užklausų įvykdymo priklausomybės nuo jų kiekio grafikas

Grafike matomas padidėjęs pranešimų perdavimo laikas panaudojant blokų grandinių autentifikaciją. Remiantis pateiktais matavimais galima nustatyti, kad prototipo sparta yra 4,5 karto lėtesnė, lyginant su išorinėmis duomenų bazėmis ir 5,5 karto lėtesnė lyginant su failo autentifikacija. Šį santykį galima sumažinti atliekant tolimesnę prototipo optimizaciją.

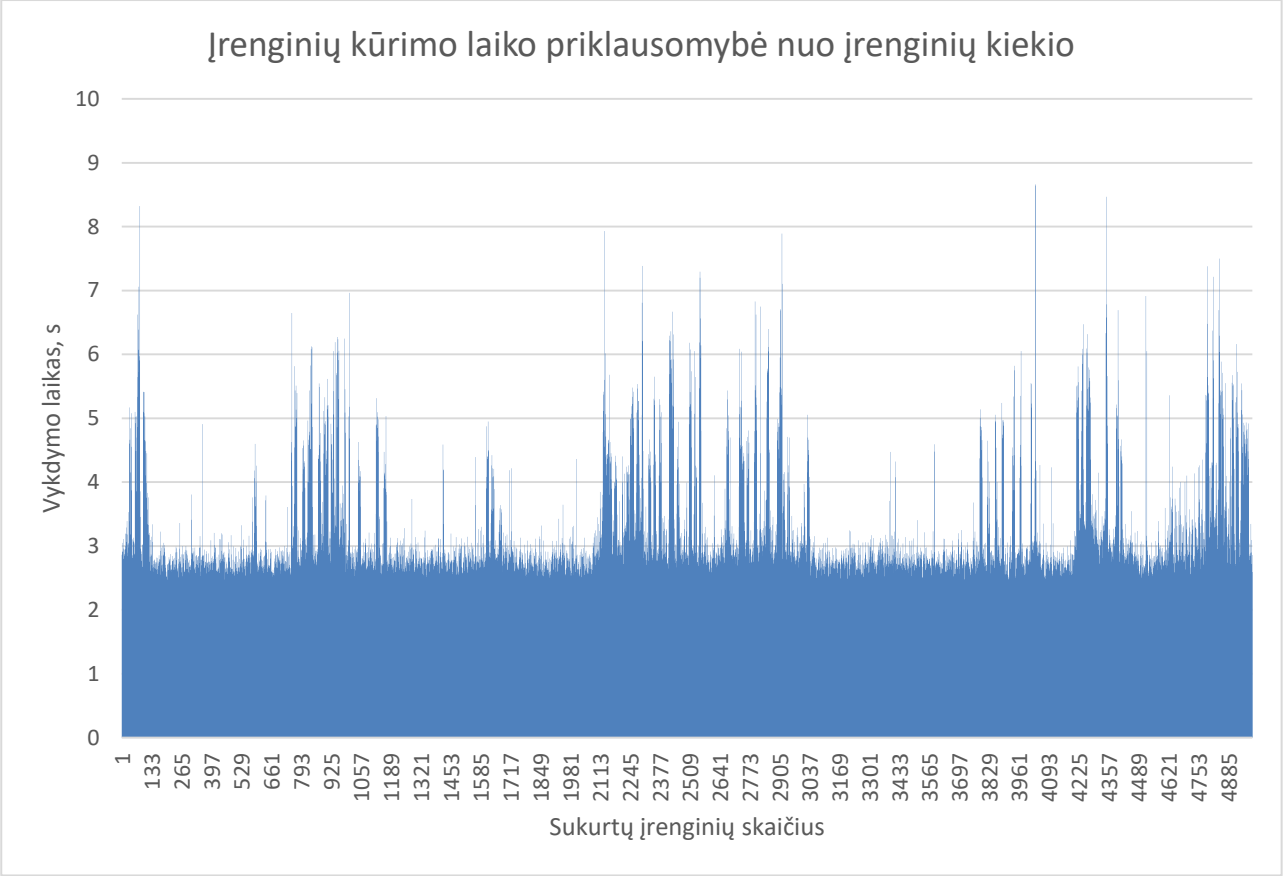
Kitame eksperimente atliekamas krūvio pasiskirstymo tyrimas. Atskaitos taškui naudojamas bandymas su vienu ūko mazgu, o toliau eksperimentas kartojamas su 3 ir 5 ūko mazgais. Tyrimo rezultatas pateiktas grafike (4.3 pav.).



4.4 pav. Užklausų vykdymo laiko priklausomybė nuo ūko sluoksnio mazgų skaičiaus

Kiekvienas ūko mazgas pasiekiamas tokio paties kiekio įrenginių ir užklausų. „MySQL“ atveju į kiekvieną ūko mazgą kreipiasi 50 lengvųjų mazgų, atlikdami 100 užklausų, o „Fabric“ atveju naudojamas toks pat kiekis lengvųjų mazgų, kur kiekvienas lengvasis mazgas vykdo 20 užklausų. Grafike matomas rezultatas parodo, kaip pradedant plėstis ūko sluoksniui pradeda ilgėti vykdymo laikas. „Fabric“ atveju ūko sluoksnio plėtimas žymių pokyčių nesukėlė, kadangi nėra priklausomybės nuo išorinio duomenų šaltinio ir kiekvienam ūko mazgui tenka vienodas krūvis.

Eksperimento metu taip pat atliktas masinis naujų įrenginių įtraukimas į blokų grandinę. Blokų grandinėje esant keliems tūkstančiams įrenginių, tolimesnis jų įtraukimas pastebimai nesulėtėjo. Žemiau pateiktame grafike (4.5 pav.) matomas įrenginių įtraukimo laikas.

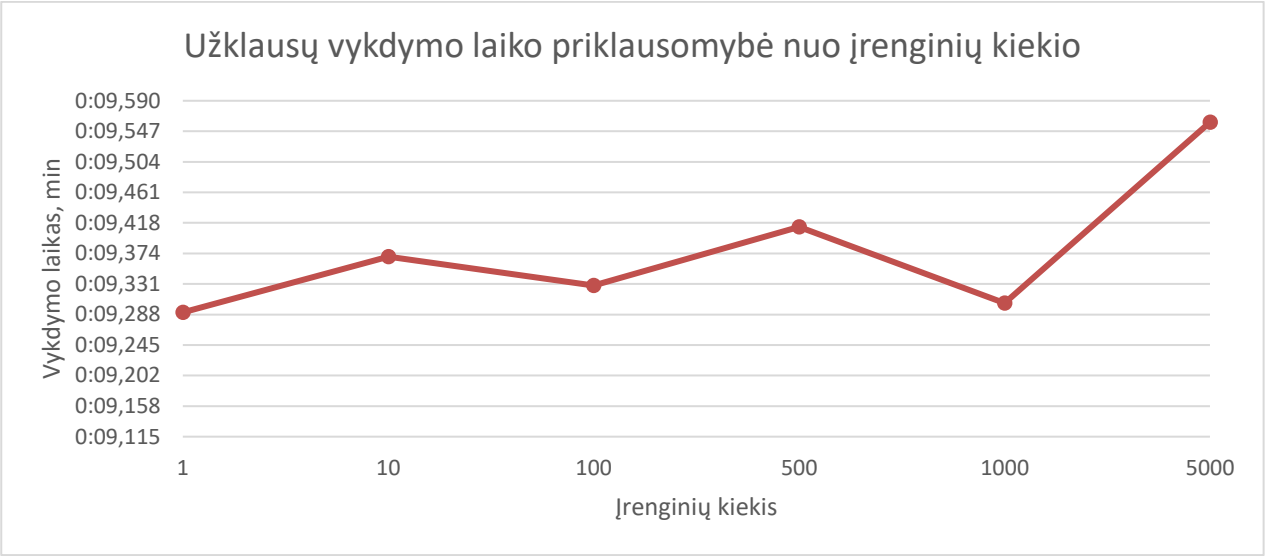


4.5 pav. Įrenginio įtraukimo į blokų grandinę laiko priklausomybė nuo blokų grandinėje esančių įrenginių kiekio

Grafike matomi reti atvejai, kai vykdymo laikas gerokai išauga, nepriklausomai nuo įtrauktų įrenginių skaičiaus. Nepaisant šio realiais atvejais pasitaikančio triukšmo, pildant blokų grandinę iki 5000 įrenginių vykdymo laikas iš esmės nepakito.

Didinat įrenginių kiekį blokų grandinėje taip pat buvo matuojama kaip tai paveikia užklausų įvykdymo laiką.

Žemiau pateikta statistika (4.6 pav.), kai 1 lengvasis mazgas siunčia 100 užklausų į vieną ūko sluoksnio mazgą.



4.6 pav. Užklausų vykdymo laiko priklausomybė nuo įrenginių kiekio blokų grandinėje

Statistika parodo, kad įrenginių skaičiaus padidėjimas neturėjo didelės įtakos užklausų vykdymo spartai.

4.3. Blokų grandinių autentifikacijos kokybinė analizė

Atsižvelgiant į eksperimento metu ištirtus kiekybinius parametrus ir diegimo metu įgyta praktika, galima išskirti tokius tiriamojo darbo prototipo kokybinius privalumus:

1. **Krūvio balansavimas.** Krūvį galima paskirstyti per pasirinkta kiekį ūko mazgų. Taip sumažinamas krūvis debesijos sluoksnyje ir išvengiama butelio kaklelio efekto.
2. **Matematiškai pagrįstas saugumas.** Visi įrenginio duomenys blokų grandinėje yra užšifruoti ir jų neteisėtas iššifravimas dabartinėmis technologijomis yra neįmanomas.
3. **Aukštas prieinamumas.** Kiekvienas ūko sluoksnio įrenginys turi identišką blokų grandinės kopiją, o tai sudaro idealias sąlygas diegiant aukšto prieinamumo ūko sluoksnį.
4. **Greitas ir efektyvus infrastruktūros plėtimas.** Decentralizavimas padeda ne tik diegiant aukštą prieinamumą, bet ir užtikrinant bendrą infrastruktūros plečiamumą.
5. **Veikimas praradus ryšį su kitais mazgais.** Net ir praradus ryšį su debesijos sluoksniu, ūko sluoksnis ir lengvieji mazgai geba ir toliau vykdyti savo veiklą.

5. TYRIMO IŠVADOS

1. Atliktoje daiktų interneto architektūrų analizėje pastebėta, kad tradicinės architektūros nebėra pajėgios susitvarkyti su dabartiniu kiekiu įrenginių ir duomenų. Debesijos kompiuterijos atsiradimas buvo didelis šuolis modernizuojant duomenų valdymą, tačiau centralizuota architektūra sudaro butelio kaklelio efektą debesijos sluoksnyje. Problemai spręsti reikalingi decentralizuoti sprendimai, gebantys dalį debesijos sluoksnio krūvio perkelti arčiau paslaugų naudotojų. Ūko kompiuterija ir jos analogai bando atlikti būtent tai. Tačiau decentralizacijos įgyvendinimas yra sudėtingas procesas, galintis sukelti infrastruktūros nestabilumų ir pažeidžiamumų. Net ir perkėlus tam tikrus procesus į tarpinį sluoksnį, jie dažnai lieka priklausomi nuo debesijos sluoksnio. Tokiu atveju praradus ryšį su debesijos sluoksniu nustoja veikti ir likusi infrastruktūra.
2. Palyginus tyrimais paremtus daiktų interneto įrenginių autentifikacijos metodus, išskirti pagrindiniai jų trūkumai. Mažas dėmesys yra skiriamas privatumo išlaikymui ir negalėjimui išsižadėti. Taip pat neatsižvelgiama į kompiuterinių resursų sąnaudas, kaip, pavyzdžiui, įrenginio talpa.
3. Tradicinių daiktų interneto architektūrų spragos taip pat atsispindi saugumo kontekste. Didelę įrenginių gausą sunku prižiūrėti ar atnaujinti. Ilgainiui daugybė pasenusių įrenginių tampa pažeidžiami. Didelė dalis kibernetinių atakų prieš įrenginius įvyksta dėl silpnos prieigos kontrolės ar autentifikacijos.
4. Blokų grandinės technologijos taikymas pradedamas įsisavinti ne tik kriptovaliutų rinkoje, bet ir kitose srityse. Blokų grandinės technologijos panaudojimas daiktų internete sprendžia esmines tradicinės architektūros spragas.
5. Blokų grandinės technologijoje naudojamas išmaniųjų kontraktų principas yra tikslingas būdas atlikti veiksmus su daiktų interneto įrenginių autentifikacijos duomenimis. Realizuojant išmaniuosius kontraktus autentifikacijai valdyti, svarbu, kad juos realizuojanti programinė įranga gebėtų lengvai integruotis su likusia infrastruktūros aplinka. „Hyperledger Fabric“ yra tikslingas to pavyzdys, kadangi ši programinė įranga integruojasi su kitais sisteminiais komponentais.
6. Sprendimo projektavimui pasirinktas vienas populiariausių daiktų interneto protokolų - MQTT. Protokolas užtikrina greitą ir autentifikuotą duomenų perdavimą tarp įrenginių per MQTT tarpininką. Tiriamajame darbe suprojektuotas specialus autentifikacijos metodas, kuris prisijungimo duomenis tikrina „Hyperledger Fabric“ blokų grandinėje. Tokiu būdu, tik blokų grandinėje registruotas įrenginys galės atlikti duomenų perdavimą ar nuskaitymą, priešingu atveju MQTT tarpininkas to padaryti neleis. Kiekvienas tarpininkas turi identišką blokų

grandinės kopiją, todėl architektūra yra decentralizuota ir įrenginiai gali naudotis bet kuriuo tarpininku.

7. Decentralizavus įrenginių komunikacijos tašką užtikrinamas krūvio pasiskirstymas, aukštas prieinamumas ir efektyvus infrastruktūros plėtimas. Sprendimas taip pat sumažina priklausomybę nuo debesijos sluoksnio. Nutrūkus ryšiui su juo, MQTT tarpininkas ir per jį komunikuojantys įrenginiai nenustoja veikti. Tiriamajame darbe sukurtas autentifikacijos mechanizmas yra lėtesnis už analogiškus sprendimus, tačiau jis turi savo unikalių privalumų, kurie sprendžia sparčiai besivystančias daiktų interneto problemas.

5.1. Galimi tolimesni tyrimo etapai

Tiriamajame darbe pateiktas prototipas sukurtas ištirti blokų grandinės panaudojimo galimybes daiktų interneto autentifikacijai atlikti. Sekantys tyrimo etapai gali būti susiję su prototipo spartos didinimu ir geresniu diegimo ir plėtimo automatizavimu (pvz. automatinis sertifikatų valdymas). Taip pat svarbūs tyrimo etapai galėtų būti integracija su kitomis sistemomis ar blokų grandinių logika.

6. LITERATŪRA

[1] J. Morgan, „A Simple Explanation Of 'The Internet Of Things,“ 13 Gegužės 2014. [Tinkle]. Available: https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#160c477e1d09. [Kreiptasi 7 Sausio 2018].

[2] A. Gerber, „Top 10 IoT security challenges,“ 17 Lapkričio 2017. [Tinkle]. Available: https://developer.ibm.com/dwblog/2017/iot-security-challenges/. [Kreiptasi 7 Sausio 2018].

[3] G. Eastwood, „4 critical security challenges facing IoT,“ 7 Vasario 2017. [Tinkle]. Available: https://www.networkworld.com/article/3166106/internet-of-things/4-critical-security-challenges-facing-iot.html. [Kreiptasi 7 Sausio 2018].

[4] T. M. Fernández-Caramés ir P. Fraga-Lamas, „A Review on the Use of Blockchain for the Internet of Things,“ *IEEE Access*, t. 6, nr. 2018, p. 32979–33001, 6 Liepos 2018.

[5] F. Bonomi, R. Milito, J. Zhu ir S. Addepalli, „Fog Computing and Its Role in the Internet of Things,“ įtraukta *MCC Workshop Mobile Cloud Comput.*, Helsinkis, Suomija, 2012.

[6] Y. Pan, P. Thulasiraman ir Y. Wang, „Overview of Cloudlet, Fog Computing, Edge Computing, and Dew Computing,“ [Tinkle]. Available: http://www.dewcomputing.org/publications/Overview.pdf. [Kreiptasi 23 Balandžio 2019].

[7] N. Kshetri, „Can blockchain strengthen the Internet of Things?,“ *IT Professional*, t. 19, nr. 4, pp. 68-72, 2017.

[8] S. Yerpude ir T. K. Singhal, „Impact of Internet of Things (IoT) Data on Demand Forecasting,“ *Indian Journal of Science and Technology*, t. 10(15), pp. 1-5, 2017.

[9] „Raspberry Pi Products,“ The Raspberry Pi Foundation, [Tinkle]. Available: https://www.raspberrypi.org/products/. [Kreiptasi 23 Balandžio 2019].

[10] K. Bachmann, „Design and Implementation of a Fog Computing Framework,“ 10 Vasario 2017. [Tinkle]. Available: http://www.infosys.tuwien.ac.at/staff/sschulte/paper/Bachmann_Master.pdf. [Kreiptasi 23 Balandžio 2019].

[11] T. Jaffrey, „MQTT and CoAP, IoT Protocols,“ Vasaris 2014. [Tinkle]. Available: https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php. [Kreiptasi 10 Balandžio 2018].

[12] „MQTT Protocol – How it Works,“ 1Sheeld, 4 Liepos 2018. [Tinkle]. Available: https://1sheeld.com/mqtt-protocol/. [Kreiptasi 2 Gegužės 2019].

[13] A. Stanford-Clark ir H. L. Truong, „MQTT For Sensor Networks (MQTT-SN) Protocol Specification," 14 Lapkričio 2013. [Tinkle]. Available: http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf. [Kreiptasi 22 Birželio 2018].

[14] Z. Shelby, „Introduction to Resource-Oriented Applications in Constrained Networks," [Tinkle]. Available: https://www.iab.org/wp-content/IAB-uploads/2011/04/Shelby.pdf. [Kreiptasi 2 Gegužės 2019].

[15] E. H. Al-Hemiary, „Internet of Things," įtraukta *The 1stInternational Conference on Information Technology*, Erbil, Kurdistan Region, 2017.

[16] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang ir L. Shu, „Authentication Protocols for Internet of Things: A Comprehensive Survey," 21 Gruodžio 2016. [Tinkle]. Available: https://arxiv.org/abs/1612.07206. [Kreiptasi 7 Sausio 2018].

[17] C. Lai, R. Lu, D. Zheng, H. Li ir X. Shen, „GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications," *Computer Networks,* t. 99, pp. 66-81, 22 Balandžio 2016.

[18] R. Sule, R. S. Kati ir R. G. Kavasseri, „A variable length fast Message Authentication Code for secure communication in smart grids," 2012. [Tinkle]. Available: https://ieeexplore.ieee.org/document/6345622. [Kreiptasi 7 Sausio 2018].

[19] H. Li, R. Lu, L. Zhou, B. Yang ir X. Shen, „An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid," *IEEE Systems Journal,* t. vol 8, nr. no. 2, pp. 665-663, 2014.

[20] K. Xue, C. Ma, P. Hong ir R. Ding, „A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications,* t. vol. 36, nr. no. 1, pp. 316-323, 2013.

[21] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Quin, X. Shen ir X.-y. Li, „S2M: A Lightweight Acoustic Fingerprints-Based Wireless Device Authentication Protocol," *IEEE INTERNET OF THINGS JOURNAL,* t. 4, nr. 1, pp. 88-100, 2017.

[22] C. Zhao, L. Huang ir Y. Zhao, „Secure Machine-Type Communications toward LTE Heterogeneous Networks," 2017. [Tinkle]. Available: https://ieeexplore.ieee.org/abstract/document/7864794. [Kreiptasi 7 Sausio 2018].

[23] „A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN Networks," 2016. [Tinkle]. Available: https://ieeexplore.ieee.org/abstract/document/7557067. [Kreiptasi 7 Sausio 2018].

[24] R. Amin, N. Kumar, G. Biswas, R. Iqbal ir V. Chang, „A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment," *Future Generation Computer Systems,* 2016.

[25] S. H. Islam, P. Vijayakumar ir M. Z. A. Bhuiyan, „A Provably Secure Three-factor Session Initiation Protocol for Multimedia Big Data Communications," 2018. [Tinkle]. Available: A Provably Secure Three-factor Session Initiation Protocol for Multimedia Big Data Communications. [Kreiptasi 7 Sausio 2018].

[26] C. Lyu, D. Gu, Y. Zeng ir P. Mohapatra, „PBA: Prediction-based Authentication for Vehicle-to-Vehicle Communications," *IEEE Transactions on Dependable and Secure Computing,* t. vol. 13, nr. no. 1, pp. 71-83, 2016.

[27] C. Lai, H. Li, R. Jiang ir X. Shen, „SEGR: A secure and efficient group roaming scheme for machine to machine communications between 3GPP and WiMAX networks," 2014. [Tinkle]. Available: https://ieeexplore.ieee.org/document/6883452. [Kreiptasi 1 Gegužės 2019].

[28] C. Lai, R. Lu, R. Jiang ir X. Shen, „LGTH: A lightweight group authentication protocol for machine-type communication in LTE networks," 2013. [Tinkle]. Available: https://ieeexplore.ieee.org/document/6831176. [Kreiptasi 7 Sausio 2018].

[29] C. Lai, H. Li, R. Lu ir X. Shen, „SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks," *Computer Networks,* t. 57, nr. 17, pp. 3492-3510, 2013.

[30] R. Amin, R. S. Sherrat, D. Giri, S. H. Islam ir M. K. Khan, „A software agent enabled biometric security algorithm for secure file access in consumer storage devices," *IEEE Transactions on Consumer Electronics,* t. vol. 63, nr. no. 1, pp. 53-61, 2017.

[31] A. Fu, S. Lan, B. Huang, Z. Zhu ir Y. Zhang, „A Novel Group-Based Handover Authentication Scheme with Privacy Preservation for Mobile WiMAX Networks," 19 Rugsėjo 2012. [Tinkle]. Available: https://ieeexplore.ieee.org/document/6307796. [Kreiptasi 1 Gegužės 2019].

[32] X. Sun, S. Men, C. Zhao ir Z. Zhou, „A security authentication scheme in machine-to-machine home network service," *Security and Communication Networks,* t. 8, nr. 16, pp. 2678-2686, 10 Gegužės 2012.

[33] „Introduction to NISTIR 7628 Guidelines for Smart Grid Cyber Security," Rugsėjis 2010. [Tinkle]. Available: https://www.nist.gov/sites/default/files/documents/smartgrid/nistir-7628_total.pdf. [Kreiptasi 7 Birželio 2018].

[34] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon ir H. F. Ahmad, „A lightweight message authentication scheme for Smart Grid communications in power sector," *Computers & Eletrical Engineering,* t. vol. 52, pp. 114-124, 2016.

[35] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang ir X. Shen, „CPAL: A Conditional Privacy-Preserving Authentication With Access Linkability for Roaming Service," 17 Vasario 2014. [Tinkle]. Available: https://ieeexplore.ieee.org/document/6742585. [Kreiptasi 1 Gegužės 2019].

[36] „A secure and efficient user authentication protocol for two-tiered wireless sensor networks," *Proceedings of the 2010 Second Pacific-Asia Conference on Circuits, Communications and System, PACCS 2010,* pp. 425-428, 2010.

[37] „IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 3: Advanced Air Interface,“ *IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009),* 2011.

[38] H. Zhu, Y. Zhang ir R. Lu, „Duth: a user-friendly dual-factor authentication for Android smartphone devices,“ *Security and Communication Networks,* t. 8, nr. 7, pp. 1213-1222, 2015.

[39] I. Rivin, „Symmetrized Chebyshev Polynomials,“ *Proceedings of the American Mathematical Society,* t. vol. 133, nr. no. 5, pp. 1299-1305, 2005.

[40] B. Schneier, „Security risks of embedded systems,“ 9 Sausio 2014. [Tinkle]. Available: https://www.schneier.com/blog/archives/2014/01/security_risks_9.html. [Kreiptasi 7 Sausio 2018].

[41] D. Papp, Z. Ma ir L. Buttyan, „Embedded systems security: Threats, vulnerabilities, and attack taxonomy,“ *13th Annual Conference on Privacy, Security and Trust (PST),* pp. 145-152, 2015.

[42] P. Boucher, S. Nascimento ir M. Kritikos, „How blockchain technology could change our lives,“ Scientific Foresight Unit (STOA), Brussels, 2017.

[43] „How Bitcoin Works,“ Freedom Node Limited, [Tinkle]. Available: https://freedomnode.com/guides/17/how-bitcoin-works. [Kreiptasi 23 Balandžio 2019].

[44] B. Asolo, „Full Node and Lightweight Node,“ 21 Gruodžio 2017. [Tinkle]. Available: https://www.mycryptopedia.com/full-node-lightweight-node/. [Kreiptasi 7 Birželio 2018].

[45] „What is the difference between a Light Node and Full Node?,“ IOTA Foundation, [Tinkle]. Available: https://www.iota.org/get-started/faqs. [Kreiptasi 2 Gegužės 2019].

[46] A. Panarello, N. Tapas, G. Merlino, F. Longo ir A. Puliafito, „Blockchain and IoT Integration: A Systematic Survey,“ *Sensors,* t. 18, pp. 2575-2601, 2018.

[47] A. Bahga ir V. K. Madisetti, „Blockchain Platform for Industrial Internet of Things,“ *Journal of Software Engineering and Applications,* nr. 9, pp. 533-546, 2016.

[48] S. Yin, J. Bao, Y. Zhang ir X. Huang, „M2M Security Technology of CPS Based on Blockchains,“ *Symmetry,* t. 9, nr. 9, p. 193, 14 Rugsėjo 2017.

[49] J. Lin, Z. Shen ir C. Miao, „Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT,“ *Proceedings of the 2nd International Conference on Crowd Science and Engineering,* pp. 38-43, 2017.

[50] M. Valenta ir P. Sandner, „Comparison of Ethereum, Hyperledger Fabric and Corda,“ Birželis 2017. [Tinkle]. Available: http://explore-ip.com/2017_Comparison-of-Ethereum-Hyperledger-Corda.pdf. [Kreiptasi 14 Balandžio 2019].

[51] X. Liang, S. Shetty, J. Zhao, J. Liu ir D. Li, „Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications,“ įtraukta *28th International Symposium on Personal, Indoor and Mobile Radio Communication,* Montreal, 2017.

[52] „Nodes,“ R3 Limited, 2018. [Tinkle]. Available: https://docs.corda.net/key-concepts-node.html. [Kreiptasi 2 Gegužės 2019].

[53] „Peers,“ Hyperledger, 27 Kovo 2019. [Tinkle]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html. [Kreiptasi 27 Balandžio 2019].

[54] „Gossip data dissemination protocol,“ Hyperledger, 5 Sausio 2019. [Tinkle]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/gossip.html. [Kreiptasi 27 Balandžio 2019].

[55] „The Ordering Service,“ Hyperledger, 29 Kovo 2019. [Tinkle]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/orderer/ordering_service.html. [Kreiptasi 27 Balandžio 2019].

[56] „Identity,“ Hyperledger, 1 Rugpjūčio 2018. [Tinkle]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/identity/identity.html#certificate-authorities. [Kreiptasi 27 Balandžio 2019].

[57] „Fabric SDK,“ Hyperledger Internationalization Working Group, 9 Balandžio 2018. [Tinkle]. Available: https://hyperledgercn.github.io/hyperledgerDocs/sdk-design_zh/. [Kreiptasi 27 Balandžio 2019].

7. PRIEDAI

Šiame skyriuje pateikiama papildoma informacija ir dokumentai, neįeinantys į pagrindinį dokumento turinį.

7.1. Priedas. Tyrime realizuotas prototipas ir literatūra

Kompaktiniame diske pateikti tyrime realizuoto prototipo komponentai. Diske taip pat patalpinti straipsniai, kuriais buvo remtasi tyrimo metu.