KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

Mindaugas Banionis

Rijndael simetrinio šifravimo algoritmo tyrimas

Magistro darbas

Darbo vadovas

doc. dr. J. Toldinas

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

Mindaugas Banionis

Rijndael simetrinio šifravimo algoritmo tyrimas

Magistro darbas

Recenzentas

dr. A. Liutkevičius

2011-05-

Vadovas

doc. dr. J. Toldinas

2011-05-

Atliko

IFN-9/3 gr. stud.

Mindaugas Banionis

2011-05-25

Kaunas, 2011

TURINYS

1. ĮVADAS

Vystantis informacinėms technologijoms, šiuolaikinio žmogaus kasdienybė tampa neatsiejama nuo milžiniško informacijos srauto, kurio šaltinis yra internetas. Gerėjant interneto ryšio kokybei, pralaidumui bei prieigos taškų skaičiui, o taip pat, skaičiavimams taikant serverius, gebančius per sekundę įvykdyti trilijonus operacijų, vis daugiau paslaugų perkeliama į virtualią erdvę. Pastaroji dažnai apibūdinama terminu „debesų kompiuterija“ (angl. cloud computing), kas reiškia, bendrinamų (angl. shared) kompiuterinių resursų dalinimąsi tinklu, internetu. Tokia resursų paskirstymo technologija leidžia optimaliai išnaudoti turimus tinklo, kompiuterių skaičiavimų ir programinės įrangos išteklius. [1]

Šiuolaikiniai mobilūs įrenginiai, pvz., delniniai kompiuteriai, sumanieji telefonai, nešiojamieji kompiuteriai yra neatsiejama „debesų kompiuterijos“ dalis. Dėl neitin galingų techninių parametrų šie įrenginiai dažnai taikomi pradinių duomenų, reikalingų sudėtingiems skaičiavimams, įvedimui bei gautų rezultatų pateikimui. Informacijos apdorojimui dažniausiai taikoma „plono kliento“ (angl. *thin client*) infrastruktūra, kuomet naudotojo mobiliajame įrenginyje yra įdiegta tik grafinė sąsaja, pvz., interneto naršyklė, o reikalingi skaičiavimai atliekami serveryje. Tai supraprastina naujinimų diegimą bei sumažina grėsmių skaičių, pvz., virusų plitimą, nes naudotojui nereikia diegti papildomos programinės įrangos, kad galėtų atlikti darbus. [2]

Mobilumas ir funkcijų gausa yra pagrindiniai kriterijai, kurie lėmė mobiliųjų įrenginių paklausos augimą per keletą pastarųjų metų: 2008 metais jų paklausa ūgtelėjo 20%, o 2009 metais iki 35%. Šie įrenginiai sėkmingai taikomi medicinoje (pvz., realiu laiku pateikia paciento būklę), versle (pvz., informuoja apie pokyčius akcijų biržoje), mokslo srityse (pvz., atstoja įprastą skaičiuoklę) ir kitur. [2, 24]

Mobiliųjų įrenginių duomenų saugumui, vientisumui ir pasiekiamumui egzistuoja grėsmės, panašios į tas, kurios būdingos įprastiems stacionariems kompiuteriams. Tačiau dėl ribotų baterijos energijos išteklių, mobiliesiems įrenginiams taikomi saugos metodai turi būti itin kruopščiai ištirti ir įvertinti tiek saugos, tiek baterijos energijos sąnaudų aspektais.

Magistrinio darbo tyrimas yra orientuotas į mobiliųjų įrenginių, konkrečiai delninių kompiuterių, baterijos energijos sąnaudas duomenų apsaugai, panaudojant Rijndael simetrinio blokinio šifravimo algoritmą. Šis algoritmas pasirinktas tyrimo objektu, nes:

- Yra universalus, užtikrina itin aukštą duomenų apsaugą t.y. atsparus žinomoms kriptoanalizės atakoms, ir labai plačiai naudojamas ribotus kompiuterinius resursus turinčiuose įrenginiuose;
- Skirtingi parametrai – duomenų bloko dydis, rakto ilgis (nuo pastarojo priklauso kriptoalgoritmo stiprumas), tyrimo eigoje leis sudaryti duomenų apsaugos profilius, atsižvelgiant į baterijos energijos sąnaudas;

- Siekiama ištirti, kaip skirtingas duomenų bloko dydis įtakos delninio kompiuterio baterijos energijos eikvojimą, taikant skirtingą informacijos šifravimo stiprumą.

Esminė tyrimo problema – duomenų šifravimas stipriai apkrauna mobiliųjų įrenginių mikroprocesorių, ko pasekoje negausūs baterijos energijos ištekliai senka labai ženkliai, todėl svarbu rasti optimalias energijos sąnaudas, esant skirtingam duomenų šifravimo stiprumui.

Magistrinio darbo tikslas – pasitelkus Microsoft .NET Compact Framework platformos teikiamus programavimo sprendimus, ištirti Rijndael simetrinio blokinio šifravimo algoritmo stiprumo įtaką delninio kompiuterio baterijos energijos sąnaudoms.

Darbo uždaviniai:

- Atlikti mokslinių straipsnių, susijusių su magistro darbo tema, analizę.
- Suformuoti eksperimentui kūriamos programos (prototipo) funkcinius ir nefunkcinius reikalavimus.
- Pagal specifikaciją realizuoti Rijndael simetrinio blokinio kriptoalgoritmo veiksmais paremtą programą ir ją tinkamai susieti su energijos sąnaudas matuojančiu programiniu moduliu.
- Ištirti ir įvertinti tyriamo kriptoalgoritmo energijos suvartojimą, kai etaloninio (angl. *benchmark*) paveikslo užšifravimui ir iššifravimui taikomi skirtingi bloko ir rakto dydžiai.

Darbo struktūra:

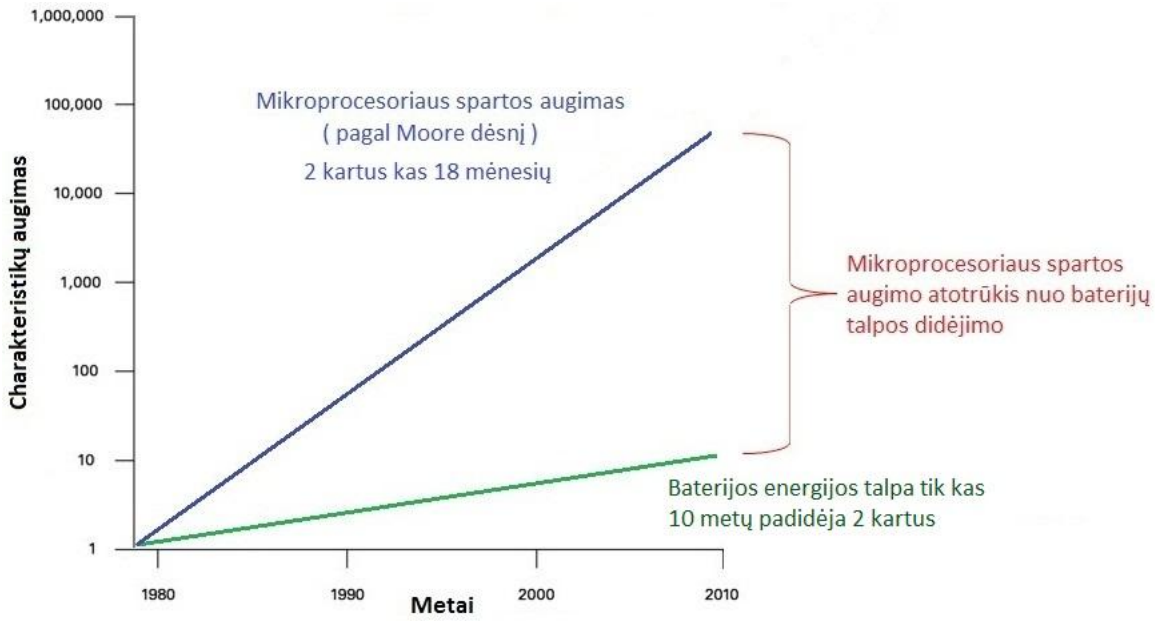
- Mokslinių straipsnių analizės dalyje pateiktos mobiliųjų įrenginių problemos, susijusios su nepakankama energijos talpa ir informacijos saugumu, taip pat, apžvelgti šiuose įrenginiuose informacijos apsaugai taikomi metodai, išskyriant duomenų šifravimą. Daugiausiai dėmesio skirta tyrimo objektui – Rijndael simetrinio blokinio algoritmo analizei: detalizuota struktūra, skirtumai nuo AES algoritmo, transformacijos, šifravimo režimai, kitų mokslininkų atlikti kriptoalgoritmų energijos sąnaudų tyrimai, rezultatų palyginimai.
- Projektavimo dalyje, aprašytas tyrimo eksperimentui suprogramuotas įrankis, aprašyti .NET Compact Framework platformos RijndaelManaged klasės kintamieji, metodai, pateikta klasių diagrama. Sudaryta panaudojimo atvejų (angl. *use case*) ir požymių diagrama, bei tyrimo programos blokinė schema. Pavaizduota eksperimento veiksmų sekos algoritmo schema.
- Tyrimo eksperimentinėje dalyje aprašytas Rijndael algoritmo energijos sąnaudų tyrimas: sudaryti taikymų variantai, įvardintos eksperimento failo – etaloninio (angl. *benchmark*)

paveikslo savybės, apibrėžti matavimų dydžiai, kurie saugomi rezultatų žurnale (angl. *log*), pateikti skaitiniai ir grafiniai tyrimo rezultatai.

- Pabaigoje pateiktos atlikto tyriamojo darbo pagrindinės išvados.

2. MOBILIŲJŲ ĮRENGINIŲ SAUGA IR ENERGIJOS SĄNAUDOS

Nors mikroprocesoriaus sparta jau eilę metų didėja pagal Moore dėsnį, deja, to paties negalime pasakyti apie mobiliųjų įrenginių baterijų energijos talpos augimą. [24] Šių dviejų komponentų charakteristikų vystymasis pavaizduotas 1 paveiksle.



1 pav. Mobiliųjų įrenginių mikroprocesoriaus ir baterijos energijos talpos vystymosi palyginimas.

Kaip matyti iš 1 paveikslo, mikroprocesoriaus sparta kas 18 mėnesių išauga du kartus, tuo tarpu baterijos energijos talpa tik per dekadą yra padvigubinama. Didesnė sparta, reikalauja didesnių energijos sąnaudų. Akivaizdus mikroprocesoriaus spartos augimo atotrūkis, mokslininkus verčia ieškoti įvairių būdų (tiek programinių, tiek techninių), padedančių taupyti energijos sąnaudas, nesukeliant vartotojui pastebimų paslaugų kokybės sutrikimų (angl. *Quality of service*). [18]

Atlikti mokslininkų tyrimai rodo, jog labiausiai baterijos energiją eikvoja LCD ekrano apšvietimas, mikroprocesorius bei bevielio ryšio įrenginys. Vidutiniškai mikroprocesorius, atmintis ir ekranas bendrai suvartoja apie 65% delninio kompiuterio baterijos energijos. Likusi energijos dalis – apie 35%, sueikvojama bevielio ryšio įrenginio. Tačiau, sistemai esant smarkiai apkrautai, pvz., duomenų šifravimo metu, mikroprocesorius gali sunaudoti iki 52% visos baterijos energijos. [16, 19]

1 lentelėje pateikti veiksniai, įtakojantys eksperimente naudojamų techninių delninuko komponentų energijos sąnaudas. [18, 20]

1 lentelė. Delninio kompiuterio pagrindinių komponentų energijos sąnaudas įtakojantys veiksniai.

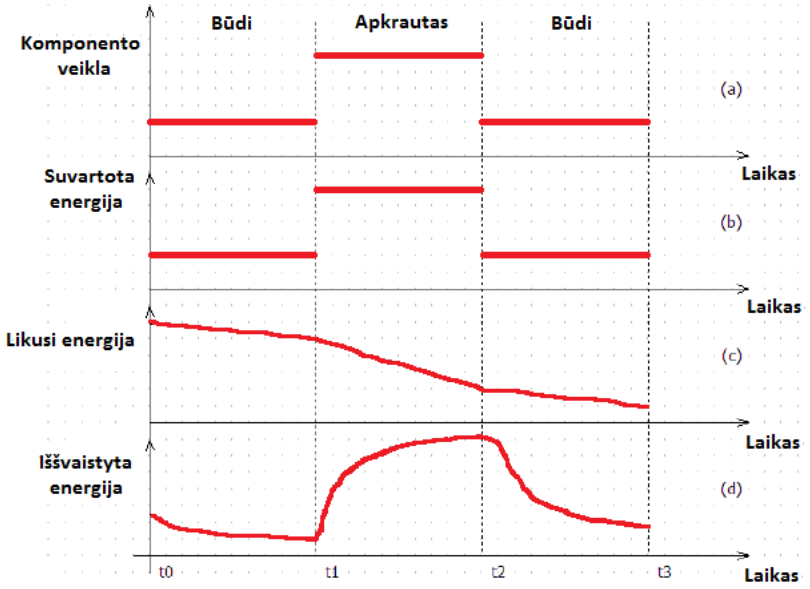
Komponentas	Veiksniai	Apibūdinimas
Mikroprocesorius	• Veikimo režimai	Priklausomai nuo operacinės sistemos ir mikroprocesoriaus (MP), galimi įvairūs veikimo režimai. Pvz, *sleep* – energijos sąnaudos yra minimalios, nes MP neatlieka jokių skaičiavimo operacijų; *idle* – atliekamos tik būtinos funkcijos t.y. MP yra būdėjimo režime; *max. load* – MP yra maksimaliai apkrautas.
Atmintis	• Kreipiniai į atmintį	Tyrimai rodo, jog atmintis didžiausias baterijos energijos sąnaudas pasiekia tada, kai į ją atliekami kreipiniai informacijai rasti arba ją išsaugant.
Ekranas	• Apšvietimo lygiai	Kuo didesnis apšvietimo lygis, kontrastas, tuo daugiau energijos sąnaudų suvartojama.
Bevielio ryšio įrenginys	• Duomenų gavimo / perdavimo sparta	Didesnei duomenų išsiuntimo (parsiuntimo) spartai pasiekti, reikalinga aukštesnė įtampa.
Baterija	• Senėjimas	Baterijai senstant, keičiasi jos cheminė sudėtis, to eigoje maksimalus įkrovimo lygis mažėja.
	• Aplinkos temperatūra	Netinkama temperatūra, kurioje veikia mobilusis įrenginys, skatina greitesnį energijos išsekimą.

Kai kurie techniniai komponentai tarpusavyje yra labai susiję, pvz., procesorius ir atmintis. Todėl, nagrinėjant energijos sąnaudų klausimus, paminėtų komponentų poros dažnai yra vertinamos bendrai. [18]

Bendra įrenginio suvartojama energija susideda iš kiekvieno techninio komponento. Vieno komponento energijos sąnaudų sumažinimas, dažnai lemia kito komponento energijos eikvojimo padidinimą. Pavyzdžiui, norint sumažinti bevielio ryšio įrenginio suvartojamą energiją, kuomet parsiunčiamų duomenų srautas yra sąlyginai didelis, reikia padidinti mikroprocesoriaus įtampą. Tokiu būdų, procesorius greičiau apdoros parsiunčiamų duomenų paketus, ko pasekoje sumažės bevielio ryšio įrenginio veikimo trukmė. [18]

Paprastai aparatiniai komponentai turi keletą profilių: veikimo ir energijos sunaudojimo. Energijos sunaudų profilis yra rinkinys energijos būsenų į kurias gali pereiti komponentas, tuo tarpu veikimo profilis apsprendžia skirtingus veikimo scenarijus (šablonus) kiekvienai energijos būsenai.

2 paveiksle, vaizduojamos teorinės energijos sąnaudos, kai tam tikro komponento (pvz. mikroprocesoriaus) veikimo būsena kinta laike. [18]



2 pav. Teorinės energijos sąnaudos, esant skirtingom komponento veikimo būsenom. [1]

2 paveiksle pateikti trys energijos sunaudojimo būviai: suvartota energija, likusi energija ir iššvaistyta energija. Kiekvienu atveju energijos pokytis laike yra įtakojamas komponento veikimo būsenų. Komponento veikimo būsenos, remiantis (a) dalimi, kinta: *būdi – apkrautas – būdi*. Dalis (b) rodo, kaip proporcingai veikimo būsenos kitimas įtakoja energijos sąnaudas. (c) atvejis, atspindi baterijos energijos eikvojimą. Dalyje (d) vaizduojama šilumos pavidalu iššvaistoma energija. Pastaroji įgauna staigų šuolį į viršų, kai komponentas yra aktyvus. [18]

Šiuolaikinių baterijų tvarkyklės (angl. *drivers*) leidžia sužinoti sekančius parametrus:

- Esamas baterijos energijos lygis (mWh);
- Maksimali energijos talpa (mWh);
- Įkrovimo / iškrovimo greitis (mW);
- Realiu laiku vykstantis baterijos energijos eikvojimas (mA);
- Likęs baterijos gyvavimo laikas iki visiško jos išsikrovimo (s);
- Baterijos temperatūra (Celcijais). [18]

Šios, programiškai išgaunamos, baterijų parametrų reikšmės gali būti naudojamos energijos sąnaudas mažinančiuose algoritmuose.

Mokslininkai Creighton T. R. Hager, Scott F. Midkiff, Jung-Min Park, Thomas L. Martin [28] atliko eksperimentą, kurio metu tyrė simetrinius blokinius algoritmus RC2, Blowfish, XTEA ir AES keliais aspektais:

- Įvertino šifravimo veiksmų energijos sąnaudas įvairiems failų dydžiams;
- Paskaičiavo apdorotų duomenų kiekį megabaitais vienam džiauliui (J).

Eksperimente buvo naudojamas HP iPAQ 4150 delninis kompiuteris su 400 MHz Intel PXA255 mikroprocesoriumi, MS PPC operacine sistema ir 1000 mAh talpos baterija. Matavimai

atlikti su skaitmeniniu multimetru, kuris per 1 sekundę geba pamatuoti 10,000 kartų, o taip pat, naudotas labai aukšto tikslumo rezistorius (0.025 Ω). [28]

Energijos sąnaudos t`uoju laiko momentu (P(t)) matuotos pagal sekančią išraišką:

$P(t) = V(t) \cdot V_{input} \,/\, R$, kur V(t) – įtampa t`uoju laiko momentu, V_{input} – įėjimo įtampa, kurios dydis pastovus ir lygus 4,1 voltui, R – rezistorius, kurio varža yra 0.025 omo (Ω).

Failo užšifravimo arba iššifravimo veiksmų sekai (darbui) sunaudota energija paskaičiuojama pagal formulę:

$$E_{darbui} = \sum_{i=0}^{n} [P(t_i) - P_{\text{Įrenginys ramyb ės b}\bar{\text{u}}\text{senoje}}] \cdot T$$

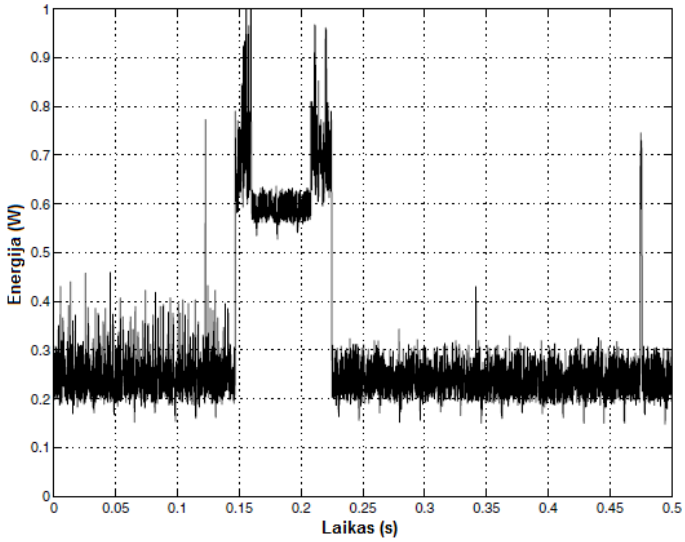
Užšifravimo arba išifravimo darbas pradedamas t_0 laiko momentu, kai matuojamos energijos lygis pastebimai padidėja – kriptoalgoritmas pradeda užšifruoti arba iššifruoti failą. Darbas baigiamas t_n laiko momentu, kai matuojamos energijos lygis ženkliai sumažėja – kriptoalgoritmas baigia užifruoti arba iššifruoti failą. Multimetras atlieka $n+1$ matavimą. Kiekviena pamatuota energijos reikšmė iššifravimo arba užšifravimo metu $P(t_i)$, $i = 0,...,n$, sumažinama dydžiu, kuris yra lygus vidutinei energijos reikšmei, kai įrenginys yra ramybės būsenoje $P_{Įrenginys\ ramybės\ būsenoje}$. Galiausiai, energijos suma, kuri yra gauta $t_0 - t_n$ laiko momentais, padauginama iš matavimų fiksavimo periodo (intervalo) $T = 100\ \mu s$, nes galima įvykdyti 10,000 matavimų per 1 sekundę. [28]

2 lentelėje pateikiamos energijos ir apdorotų duomenų vidutinės skaitinės vertės minėtiems simetriniams blokiniams algoritmams.

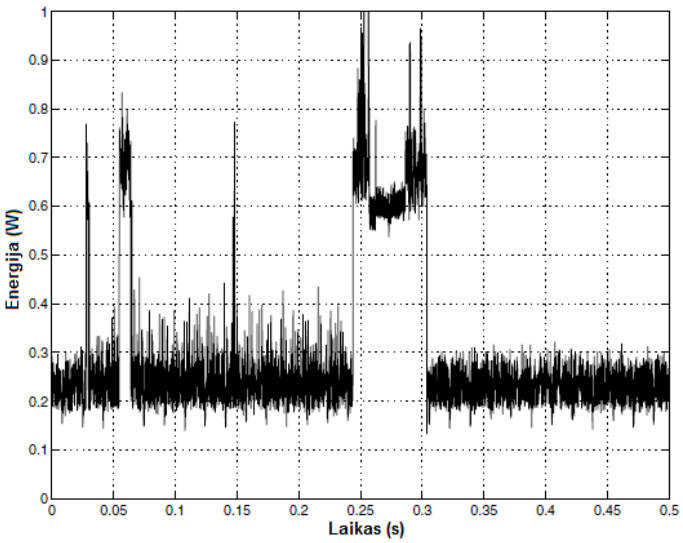
2 lentelė. Simetrinių algoritmų energijos vidutinės sąnaudos baitui bei megabaitų srautas vienam džiauliui .

Algoritmas	**Rakto/Bloko dydis (bitai)**	**Ciklų skaičius**	**Failo dydis (baitai)**	**Užšifravimas (µJ/Baitui)**	**Iššifravimas (µJ/Baitui)**	**Užšifravimas (Megabaitai/J)**	**Iššifravimas (Megabaitai/J)**
RC2	40 / 64	18	2^{10}	0.633	0.623	1.507	1.531
			2^{12}	0.497	0.506	1.919	1.886
			2^{15}	0.401	0.420	2.381	2.269
			2^{17}	0.387	0.403	2.463	2.366
			2^{20}	0.385	0.401	2.480	2.378
Blowfish	448 / 64	16	2^{10}	1.858	1.883	0.513	0.506
			2^{12}	0.643	0.624	1.484	1.529
			2^{15}	0.276	0.276	3.451	3.457
			2^{17}	0.244	0.239	3.909	3.986
			2^{20}	0.237	0.236	4.027	4.046
XTEA	128 / 64	64	2^{10}	1.003	1.071	0.951	0.890
			2^{12}	0.875	0.789	1.113	1.208
			2^{15}	0.912	0.946	1.046	1.009
			2^{17}	0.887	0.881	1.075	1.083
			2^{20}	0.902	0.931	1.057	1.024
AES	256 / 128	14	2^{10}	2.829	2.879	0.337	0.331
			2^{12}	2.599	2.531	0.367	0.377
			2^{15}	2.524	2.505	0.378	0.381
			2^{17}	2.506	2.483	0.381	0.384
			2^{20}	2.594	2.582	0.368	0.369

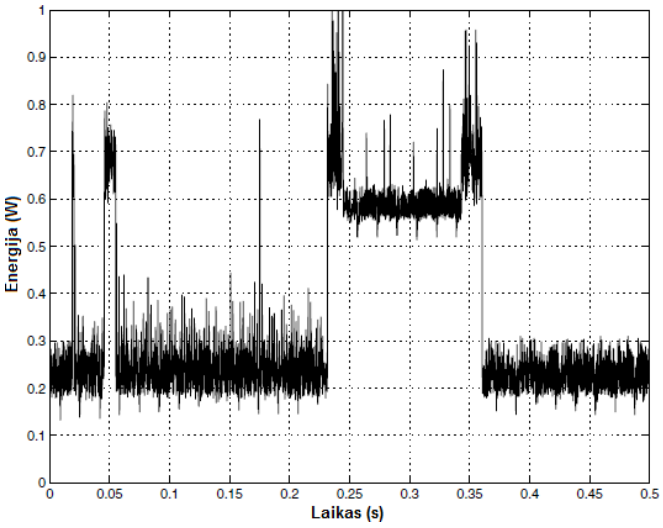
Paveiksluose 3-6 vaizduojamos kriptoalgoritmų energijos sąnaudos, užšifruojant failą (energijos lygis tuo metu pakilęs ir laikosi apie 0.6 vato), kurio dydis 2^{15} baitai t.y. 32 KB. Energijos šuoliai užšifravimo proceso pradžioje ir pabaigoje atsiranda dėl vietos atmintyje išskyrimo ir atlaisvinimo. Energijos šuolius, kurie matyti dar prieš failo užšifravimą (4-5 paveikslai), iššaukė tekstinio failo įkrovimas į atmintį. [28]



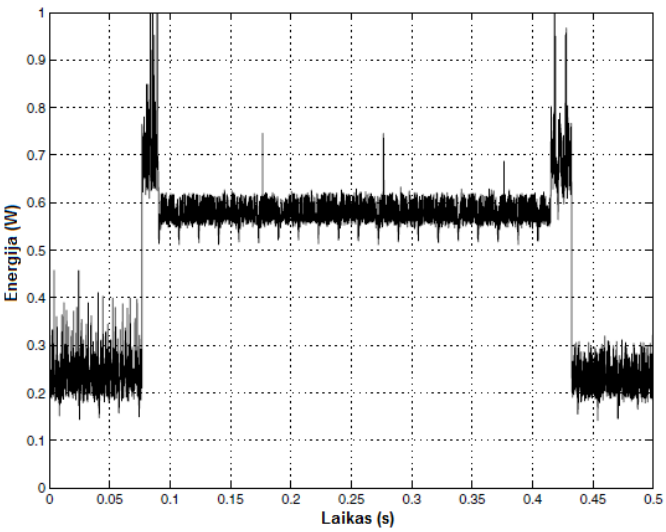
3 pav. RC2 energijos sąnaudos, užšifruojant 32 KB tekstinį failą. [2]



4 pav. Blowfish energijos sąnaudos, užšifruojant 32 KB tekstinį failą. [2]



5 pav. XTEA energijos sąnaudos, užšifruojant 32 KB tekstinį failą. [2]



6 pav. AES energijos sąnaudos, užšifruojant 32 KB tekstinį failą.[2]

[28] straipsnio autoriai po tyrimo priėmė tokias išvadas:

- Dėl būklių matricoje (S-box) ir sukeitimų matricoje (P-box) eilės susietų matematinių operacijų, kurios suteikia stiprias kriptografines savybes, AES kriptoalgoritmui prireikė daugiau atminties, kompiuterinių skaičiavimų, bei laiko, nei bet kuriam kitam šiame eksperimente tirtam algoritmui.

- Užšifruojant ir iššifruojant didelius failus Blowfish algoritmas suvartoja mažiausiai energijos, lyginant su tirtais algoritmais, tačiau energijos sąnaudos išauga, kai atvyro teksto failo dydis mažėja (žiūrėti 2 lentelę).
- Visi tirti kriptoalgoritmai užšifravimo proceso metu suvartoja panašų kiekį energijos, tačiau greitesni algoritmai, kurių operacijoms reikia mažiau skaičiavimų, vykdomi trumpesnį laiką, todėl energijos suvartoja mažiau.

2.1. Grėsmės duomenų saugumui

Kenkėjiškų programų, skirtų mobiliesiems įrenginiams, kūrimo ir atsiradimo viešumoje ženklus pagausėjimas buvo 2004 – 2006 metai. Per šį laikotarpį mobiliųjų įrenginių vartotojai kentėjo nuo tokių programų, kaip: virusai, kirmėlės (angl. *worms*), trojos arkliai (angl. *trojans*), įvairios šnipinėjimo, nepageidaujamos reklamos programos. [26]

Laikotarpyje tarp 2006 – 2009 metų, šių kenkėjiškų programų tik daugėjo, pagal „Kaspersky Lab“ duomenis:

- 2009 metais iš viso buvo žinomos 106 šių kenkėjiškų programų šeimos, kai 2006 metais buvo žinoma tik 31. Taigi jų skaičius išaugo 3,4 karto.
- 2009 metais buvo žinoma 514 kenkėjiškų programų modifikacijų, tuo tarpu 2006 metais 170. Skaičius padidėjo 3 kartus. [26]

Įmonės, norėdamos išlikti konkurencingos, darbuotojus aprūpina delniniais kompiuteriais, kad šie galėtų dirbti iš bet kur. Šiuolaikiniai mobilieji įrenginiai gali talpinti gigabaitus informacijos, bet ne visada susirūpinama dėl šių duomenų saugumo. „iAnywhere“ apklausė 104 IT profesionalus, iš kurių net 78% savo mobiliuose įrenginiuose talpino konfidencialią informaciją (pvz., laiškus, klientų duomenis), bet tik 68% duomenims apsaugoti naudojo šifravimo metodus ar slaptažodžius. [4]

Sumanieji mobilūs įrenginiai turėdami bevielio ryšio įtaisus gali keistis duomenimis per bevielio ryšio priegos taškus. Daugelis viešuose vietose laisvai prieinamų bevielio ryšio priegos taškų nereikalauja prisijungimo duomenų, todėl perduodami internetu duomenys nėra šifruojami, apsaugomi. Taigi, vien faktas, jog mobilus įrenginys atsidūrė už sąlyginai saugaus įmonės privataus tinklo perimetro, verčia susimąstyti apie galimą duomenų praradimą, pavogimą. [4]

Neapsaugotiems duomenims išgauti tinklu, internetu, programišiai gali taikyti, pvz., tinklo srauto analizatorių (angl. *sniffers*) programinę įrangą arba sekančias atakas:

- „Žmogus viduryje“ ataka (angl. *Man in the middle*), skirta, pvz., „Bluetooth“ įrenginiais perduodamų duomenų perėmimui, jų sugadinimui.
- Įrenginio aptikimo ataka (angl. *Device discovery*) - atakos zonoje esančių „Bluetooth“ įrenginių adresams sužinoti. Išgautus duomenis galima panaudoti „Bluesnarfing“ atakai,

kuomet per „Bluetooth“ susijungimą sudaroma neautorizuota prieiga prie kalendoriaus, kontaktų sąrašo, elektroninio pašto ir pan.

- Informacijos vagystės (angl. *Information theft*) atakos metu mobilieji įrenginiai užkrėčiami kenkėjiškomis programomis, kurios iš atminties išgauna konfidencialią informaciją ir ją išsiunčia įsibrovėliui. [5, 6]

Be to, dažnas mobiliuosius įrenginius pametame, neapsižiūrėję paliekame stotyse, traukiniuose. Anot „iAnywhere“, taip yra nutikę 57% jos apklaustų respondentų, tiesa, net 49% pavyko šiuos įrenginius atgauti. Bet net ir atgavus įrenginį, jei duomenys nebuvo užšifruoti ar kitaip apsaugoti, išlieka didelė tikimybė, jog svarbi informacija sėkmingai pasisavinta svetimų asmenų. [4]

Kadangi grėsmių skaičius mobiliesiems įrenginiams nuolat didėja, techninės ir programinės įrangos gamintojai saugumui skiria itin daug dėmesio t.y. sauga užima aukštą poziciją tarp šių įrenginių teikiamų galimybių. Žemiau pateikiami pagrindiniai į mobiliuosius įrenginius orientuoti saugumo sprendimai ir problemos, kurias jie sprendžia:

- Vartotojo identifikacija (angl. *user identification*) užtikrina, jog tik autorizuoti naudotojai gali naudotis prietaisu.
- Saugi laikmena (angl. *secure storage*), skirta saugoti jautriai informacijai, pvz., slaptažodžiams, PIN kodams, kriptoalgoritmų raktams, sertifikatams ir pan. Šia laikmena gali būti, pvz., įrenginio „Flash“ atmintis.
- Saugi programinės įrangos vykdymo aplinka (angl. *secure software execution environmenti*) yra būtina, norint užkirsti kenkėjiškų programų atakas.
- Klastojimams atsparių sistemų diegimas (angl. *tamper-resistant system implementation*), kad užtikrinti techninės įrangos saugumą nuo įvairių fizinių ir elektrinių atakų.
- Saugus prisijungimas prie tinklo (angl. *secure network access*) užtikrina, jog tik autorizuoti įrenginiai gali prisijungti prie tinklo ar naudotis paslaugomis.
- Saugūs apsikeitimai duomenimis (angl. *secure data communications*) apsprendžia duomenų perdavimo privatumą ir integralumą (vientisumą) iš/į mobilųjį įrenginį.
- Turinio saugumas (angl. *content security*) orientuotas į tai, kaip užtikrinti, jog turinys, kuris yra parsisiųstas į mobilųjį įrenginį, būtų naudojamas pagal turinio autoriaus pateiktas sąlygas (pvz., tik skaityti, bet ne kopijuoti). [3]

2.2. Saugūs duomenys mobiliajame įrenginyje

Parašyta žinutė, kurią gali suprasti bet kuris stebėtojas yra vadinama atviru tekstu (angl. *plaintext*). Tam, kad žinutę galėtų perskaityti tik konkretus asmuo (asmenys), naudojami du pagrindiniai procesai:

- Užšifravimo (angl. *encryption*) proceso meto atvyras tekstas paverčiamas į formą, kuri paslėpia žinutės reikšmę nuo visų asmenų, kuriems ji nėra skirta. Tokia žinutė vadinama užšifruota.
- Iššifravimo (angl. *decryption*) proceso metu vyksta atvirkštiniai užšifravimui veiksmai t.y. žinutė iš užšifruotos formos atverčiama į atvyrą tekstą. [7]

Šie du procesai yra parametrizuoti raktu, kuris idealiu atveju yra žinomas tik šią teisę turintiems komunikavimo dalyviams. Kadangi raktas yra paslaptis, todėl šiai paslapčiai išsaugoti turi būti imtąsi visų įmanomų saugumo priemonių.

Taisyklės, apibrėžiančios veiksmus, kuriuos turi vykdyti kelios komunikuojančios pusės, kad duomenys, kuriais apsikeičiama, išliktų saugūs, vadinamos saugos protokolais (angl. *security protocols*). Saugos protokolų taisyklės yra paremtos keturiais saugos tikslais:

- Konfidencialumas (angl. *confidentiality*) užtikrina, jog niekas kitas, kaip tik teisėti dalyviai, negalės suprasti komunikacijos kanalais perduodamos ar atmintyje saugomos informacijos. Šiam tikslui pasiekti, dažniausiai naudojami kriptoalgoritmai.
- Autentifikacija (angl. *authentification*) sprendžia problemą, jog žinutės siuntėjas yra tas, kuo ir dedasi.
- Vientisumas (angl. *integrity*) žinutės gavėjui suteikia galimybę patikrinti, jog žinutės turinys, jos siuntimo metu, nebuvo pakeistas įsibrovėlio.
- Neatsisakymas (angl. *nonrepudiation*) neleidžia žinutės siuntėjui paneigti fakto, jog būtent jis šią žinutę išsiuntė. [7]

Šie keturi saugos tikslai užtikriną pasitikėjimą tarp kelių komunikuojančių dalyvių. Šis pasitikėjimas realybėje išreiškiamas, kaip bendravimas akis į akį (angl. *face to face*), o internete, kur dominuoja beveidis (angl. *faceless*) bendravimas, pasitikėjimas užtikrinamas pasitelkus kriptografijos algoritmus. Priklausomai nuo charakteristikų kriptoalgoritmai skirstomi į tris kategorijas:

- Simetriniai (angl. *simmetric*) algoritmai duomenų užšifravimui ir iššifravimui naudoja tą patį slaptą raktą. Šių algoritmų sąvybės paremtos „painiavos ir sklaidos“ (angl. *confusion and diffusion*) sąvokomis. Daugiausiai naudojami duomenų konfidencialumui užtikrinti.

- Asimetriniai (angl. *asymmetric*) algoritmai naudoja skirtingus raktus (viešuosius ir privačiuosius) užšifravimo ir iššifravimo metodams. Šie algoritmai sukonstruoti iš matematinių abstrakcijų - vienos krypties funkcijų (angl. *one way functions*), kurios paremtos sunkiai sprendžiamomis skaičiavimų problemomis, pvz., sveiko skaičiaus faktorizacija, diskretiniai logaritmai. Naudojami autentifikacijos ir neatsisakymo tikslais. Be to, asimetriniai algoritmai yra beveik 1000 kartų lėtesnį, nei simetriniai algoritmai, nes reikalauja kur kas didesnių skaičiavimų.
- Maišos (angl. *hash*) algoritmai iš apibrėžto ilgio žinutės sudaro fiksuoto ilgio skaičių, atitinkantį žinutę. Net mažiausi originalios žinutės pakeitimai įtakos maišos funkcijos reikšmę. Naudojami apsikeistų duomenų vientisumui patikrinti. [7]

Priklausomai nuo duomenų saugumo tikslų turi būti atidžiai parenkami tinkami kriptoalgoritmai, leidžiantys realizuoti reikalingus saugos protokolus, pvz., WPA (Wi-Fi Protected Access) protokolą, naudojamą šifruojamų duomenų perdavimui bevieliu tinklu.

2.2.1. Simetrinio blokinio Rijndael (AES) kriptoalgoritmo analizė

Simetrinis blokinis Rijndael algoritmas (tariamas „Reign Dahl“, „Rain Doll“ arba „Rhine Dahl“) sukurtas dviejų Belgijos mokslininkų: Dr. Vincent Rijmen ir Dr. Joan Daemen. Dabar plačiai paplitęs Advanced Encryption Standard (AES) algoritmas yra sudarytas Rijndael algoritmo pagrindu. AES algoritmą rekomenduoja JAV Nacionalinis standartų institutas ir Technologijos institutas (NIST), siekiant apsaugoti jautriai ir neklasifikuotai vyriausybės informacijai. 2001 metais AES buvo patvirtintas Federalinės informacijos tvarkymo standartu (FIPS 197). [8]

Pagrindiniai kriterijai, lėmę Rijndael kriptoalgoritmo pasirinkimą, kaip AES yra sekantys:

- Saugumas – šiai dienai šis algoritmas yra laikomas saugiu t.y. viešai nėra paskelbta apie sėkmingai įvykdytą ataką prieš šį algoritmą;
- Efektyvumas, veiksmingumas bei lankstumas įgalina šį algoritmą pritaikyti plačiame skirtingų techninės ir programinės įrangos platformų diapazone;
- Struktūros dizaino paprastumas.

Esminiai skirtumai tarp Rijndael ir AES kriptoalgoritmų yra bloko dydis ir galimi raktų ilgiai. Rijndael duomenų bloko (duomenų bitų grupė, kuriais algoritmas manipuliuoja) ir rakto (paslapties) ilgio savybės moksliniuose straipsniuose apibrėžiamos taip:

- Tiek blokas, tiek raktas skaičiavimuose gali būti parenkamas kas 32 bitus, pradedant nuo minimalios 128 bitų reikšmės;
- Bloko maksimalus dydis yra 256 bitai;

- Rakto maksimalus ilgis praktikoje naudojamas 256 bitų, tačiau teoriškai maksimali rakto ilgio vertė nėra apibrėžta.

Tuo tarpu AES standartas griežtai apibrėžia bloko dydį lygiu 128 bitams, o rakto ilgis gali turėti tik tris vertes t.y. 128, 192 arba 256 bitus. [8]

2.2.1.1. Rijndael (AES) kriptoalgoritmo struktūra

Kai kurios Rijndael kriptoalgoritmo operacijos yra apibrėžtos baitų lygyje, o baitus atstovaujantys elementai yra išdėstyti baigtiniame sveikų skaičių lauke GF(2^8) (pavadintu mokslininko Evariste Galois garbei). Kitos operacijos apibrėžtos keturių baitų ilgio žodžiais. [9]

Kiekvienam pirminio skaičiaus laipsniui egzistuoja vieta baigtinių skaičių lauke, taigi, visos skaičių reikšmės, kurios yra gaunamos pirminį skaičių pakėlus laipsniais iš lauko GF(2^8), yra izomorfinės. [9]

Baitas *b*, susidedantis iš bitų $b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$, yra laikomas polinomu su koeficientu iš erdvės {0,1}:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x^i$$

Pvz., šešioliktainė reikšmė "57" (dvejetainė 01010111) atitinka polinomą:

$$x^6 + x^4 + x^2 + x + 1$$

Rijndael algoritmą sudaro ciklo (angl. *Round*) transformacija, kuri susideda iš keturių, viena nuo kitos nepriklausomų ir pastovių transformacijų (kiekvienam būsenos masyvo bitui taikomi tie patys veiksmai), dar vadinamų sluoksniais. Nepriklausomi sluoksniai yra pasirinkti, remiantis „Plataus tako strategija“ (angl. *Wide trail strategy*), kuri užtikrina pasipriešinimą prieš linijinę ir diferencialinę kriptoanalizę. Pagal „Plataus tako strategiją“, kiekvienas sluoksnis turi savo funkciją:

- Linijinis maišymo sluoksnis (angl. *Linear mixing layer*) garantuoja aukštą difuziją vos po kelių ciklų;
- Nelinijinis sluoksnis (angl. *Non-linear layer*) lygiagrečiai taiko sukeitimų lentelę (angl. *Substitution box*), kurios tikslas paslėpti ryšius tarp rakto ir užšifruoto teksto, dėl to S-box pasižymi optimaliai blogiausiomis tiesiškumo savybėmis;
- Rakto papildymo sluoksnyje (angl. *Key addition layer*) kiekvienos būklės baitas skaičiavimų eigoje yra apjungiamas su ciklo raktu, panaudojant loginę išraišką EXOR. [9]

Algoritmo aprašytos transformacijos (sluoksniai) dirba su tarpiniais užšifravimo (iššifravimo) procedūros rezultatų duomenimis, kurie sudaro būseną (angl. *State*). Būsena gali būti apibrėžiama kaip dvimatis stačiakampis baitų masyvas. Masyvą sudaro keturios eilutės ir stulpeliai (toliau žymimi Nb), kurių skaičius priklauso nuo bloko dydžio, pastarąjį dalinant iš 32. [9]

Šifro raktas taip pat gali būti atvaizduojamas dvimačiame masyve, kurį sudaro keturios eilutės, o stulpelių skaičius (toliau žymimas Nk) priklauso nuo rakto ilgio, pastarąjį dalinant iš 32.

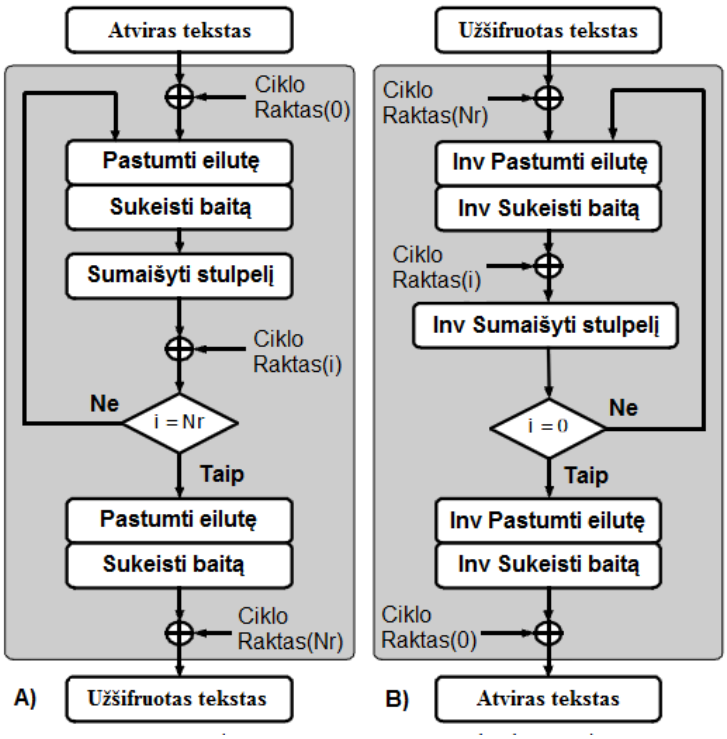
Rijndael algoritmo įėjime (išėjime) yra paduodami (gaunami) vienmačiai 8 bitų masyvai, kurie sudaro baitus, numeruojamus nuo 0 iki 4*Nb-1. Tokie baitų blokai, priklausomai nuo bloko dydžio, sudaromi iš 16, 24 ir 32 baitų. Lygiai taip pat yra ir su šifro raktu, tik čia vietoj Nb imama Nk reikšmė.

Algoritmo transformacijų vykdymo ciklų skaičius (Nr) priklauso nuo Nb ir Nk ir yra pateiktas 3 lentelėje. Pastaba, 32 bitų architektūros kompiuteriuose Nb ir Nk yra išreiškiami 4 baitų ilgio dvigubu žodžiu (angl. *Double Word*), pvz., Nb = 4 DWORD = 4 baitai * 4 baitai * 8 bitai = 128 bitai. [9]

3 lentelė. Transformacijų ciklų skaičius (Nr), priklausantis nuo bloko dydžio ir rakto ilgio.

Nr	**128 bitai, Nb = 4**	**192 bitai, Nb = 6**	**256 bitai, Nb = 8**
128 bitai, Nk = 4	10	12	14
192 bitai, Nk = 6	12	12	14
256 bitai, Nk = 8	14	14	14

Paveiksle 7 pateiktas Rijndael (AES) užšifravimo (A) ir iššifravimo (B) algoritmas. Šiuose algoritmuose taikomos tos pačios transformacijos, tik iššifravimo atveju, transformacijų veiksmai vykdomi atvirkščia tvarka.



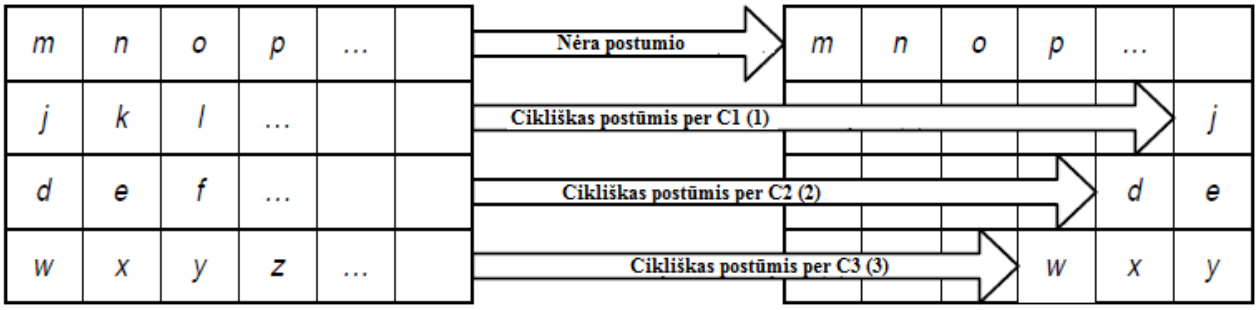
7 pav. Rijndael(AES) užšifravimo (A) ir iššifravimo (B) algoritmas. [3]

Eilutės postūmio (angl. *Shift row*) transformacija būklių matricos tris paskutines eilutes pastumia skirtingais postūmiais (C1, C2, C3), kurie priklauso nuo bloko dydžio Nb = bloko dydis bitais / 32 ir yra pateikti 2 lentelėje. [9, 10]

4 lentelė. Būklių matricos eilučių postūmiai, priklausantys nuo bloko dydžio.

	Postūmiai (baitais)		
Nb	**C1**	**C2**	**C3**
4	1	2	3
6	1	2	3
8	1	3	4

Pirmoji eilutė nėra pastumiama, antroji pastumiama per C1 baitų, trečioji per C2 baitų ir ketvirtoji per C3 baitų. Ši transformacija vaizdžiai pateikta 8 paveiksle.



8 pav. Eilutės postūmio transformacija vaizdžiai.[4]

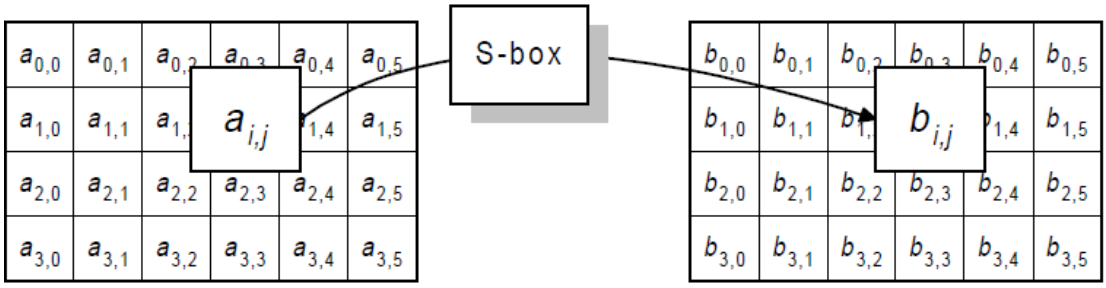
Eilutės postūmio inversijos transformacija tris apatines eilutes pastūmia per Nb – C1, Nb – C2 ir Nb – C3 baitų, taip, kad baitas, esantis eilutės i pozicijoje j, pasislenka į poziciją $(j + Nb - C_i)\ mod\ Nb$. [9]

Baitų sukeitimo (angl. *Byte substitution*) transformacija yra netiesinė operacija, operuojanti su kiekvienu būklės matricos baitu nepriklausomai. Sukeitimų lentelė yra paversta (angl. *invertible*) ir formuojama iš dviejų transformacijų kompozicijos:

- Paimama polinomo, pvz., $x^6 + x^4 + x^2 + x + 1$ (aprašyto kiek aukščiau) priešingybė (inversija);
- Pritaikoma afinioji (angl. *affine*) transformacija:

  $b_i = a_i \oplus a_{(i+4)mod\,8} \oplus a_{(i+5)mod\,8} \oplus a_{(i+6)mod\,8} \oplus a_{(i+7)mod\,8} \oplus c_i$, kiekvienam $0 \le i \le 8$, kur a_i yra i-tasis baito a bitas, o c_i i-tasis baito c, kurio dvejetainė reikšmė {01010111}, bitas.

  Baitų sukeitimo transformacijos poveikis, vaizduojamas 9 paveiksle.



9 pav. Baitų sukeitimo transformacija vaizdžiai. [4]

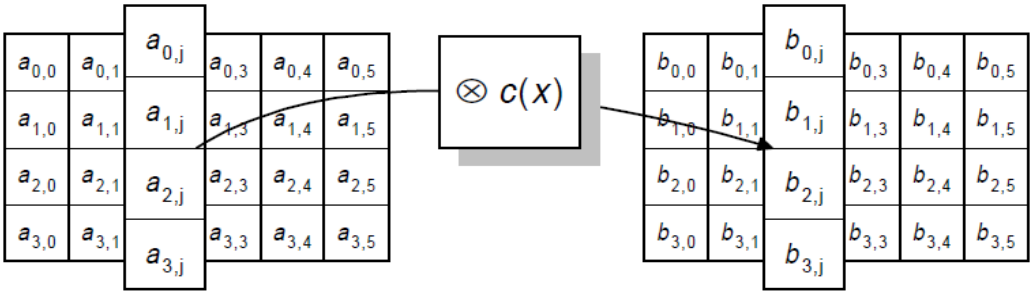
Baitų sukeitimo inversijos transformacija atliekama, pritaikant atvirkštinę afiniąją transformaciją. [9]

Stulpelių maišos (angl. *Mix column*) transformacijoje kiekvienas būklės matricos stulpelis yra laikomas GF(2^8) grupės polinomu ir padaugintas iš $modx^4 + 1$ su statiniu (nekintančios išraiškos) polinomu $c(x) = `03`x^3 + `01`x^2 + `01`x + `02`$. Pastarasis polinomas yra santykinai pirminis (angl. *coprime*) išraiškai $x^4 + 1$, todėl invertuojamas. Tai galima išreikšti matricų daugyba. Tegul $b(x) = c(x) \oplus a(x)$ t.y. [9]

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

10 pav. $b(x) = c(x) \oplus a(x)$ *matricų daugyba. [4]*

Transformacijos iliustravimas pateiktas žemiau:



11 pav. Stulpelių maišos transformacija vaizdžiai.[4]

Ciklo rakto pridėjimo (angl. *Round key addition*) transformacijoje ciklo raktas sudedamas su esama būsena, panaudojant loginę XOR operaciją. Rakto ilgis yra lygus bloko Nb dydžiui.

12 pav. Ciklo rakto sudėjimas su esama būsena, panaudojant loginę XOR operaciją. [4]

Ciklo rakto pridėjimo inversijos transformacija turi grįžtamąjį ryšį pati su savimi t.y. antrą kartą panaudojus šią transformaciją, gaunamas atvirkštinis rezultatas. [9, 10]

2.2.1.2. Rijndael (AES) kriptoalgoritmo transformacijų tyrimas

Mokslininkai M. Razvi Doomun, KM Sunjiv Soyjaudah ir Devesh Bundhoo [11] apskaičiavimo, kiek Rijndael kriptoalgoritme panaudojama loginių operacijų (OR, AND) bei baitų postūmių (angl. *Shift bytes*), esant įvairiems rakto ilgiams ir bloko dydžiams. Lentelėje 5 pateikiami šių operacijų skaičiai, kai duomenys užšifruojami ir iššifruojami.

5 lentelė. Būklių matricos eilučių postūmiai, priklausantys nuo bloko dydžio.

Rakto ilgis / bloko dydis	Rijndael užšifravimas				Rijndael iššifravimas		
	OR	AND	Baitų postūmis		OR	AND	Baitų postūmis
128/128	1268	1720	408		3860	5176	1272
128/192	1540	2088	496		4708	6312	1552
128/256	1812	2456	584		5556	7448	1832
192/128	2310	3132	744		7062	9468	2328
192/192	2310	3132	744		7062	9468	2328
192/256	2718	3684	876		8334	11172	2748
256/128	3624	4912	1168		11112	14896	3664
256/192	3624	4912	1168		11112	14896	3664
256/256	3624	4912	1168		11112	14896	3664

Pavadinkime **Nb** = bloko dydis/32, **R** – transformacijų ciklų skaičius, **Ta** – AND skaičius, **To** – OR skaičius, **Ts** – baitų postūmių skaičius. Tuomet, operacijų skaičių, reikalinga vieno duomenų bloko užšifravimui Rijndael(AES) kriptoalgoritmu, galima išreikšti sekančia išraiška:

$$Užšifravimas_{Operacijų\ skaičius} = (46N_bR - 30N_b)\,T_a + [31N_bR + 12(R-1) - 20N_b]\,T_o + \\ + [64N_bN_r + 96(R-1) - 61N_b]T_s$$

Stulpelių maišos inversijos transformacijai reikia $96N_bT_a + 72N_bT_o - 32N_bT_s$ daugiau operacijų, nei stulpelių maišos transformacijai, todėl vieno duomenų bloko iššifravimui reikalingų operacijų skaičių galima rasti, pagal išraišką:

$$Iššifravimas_{Operacijų\ skaičius} = Užšifravimas_{Operacijų\ skaičius} + \\ + \{[96N_bT_a + 72N_bT_o - 32N_bT_s] \times (R-1)\}$$

6 lentelėje įvertinamas įvairioms transformacijoms reikalingas AND, OR ir baitų postūmio operacijų skaičius per visą vieno 128 bitų bloko užšifravimo procesą, kai rakto ilgis 128, o skaičiavimų ciklų skaičius = 10. [11]

6 lentelė. Užšifruojant transformacijose panaudojamos operacijos .

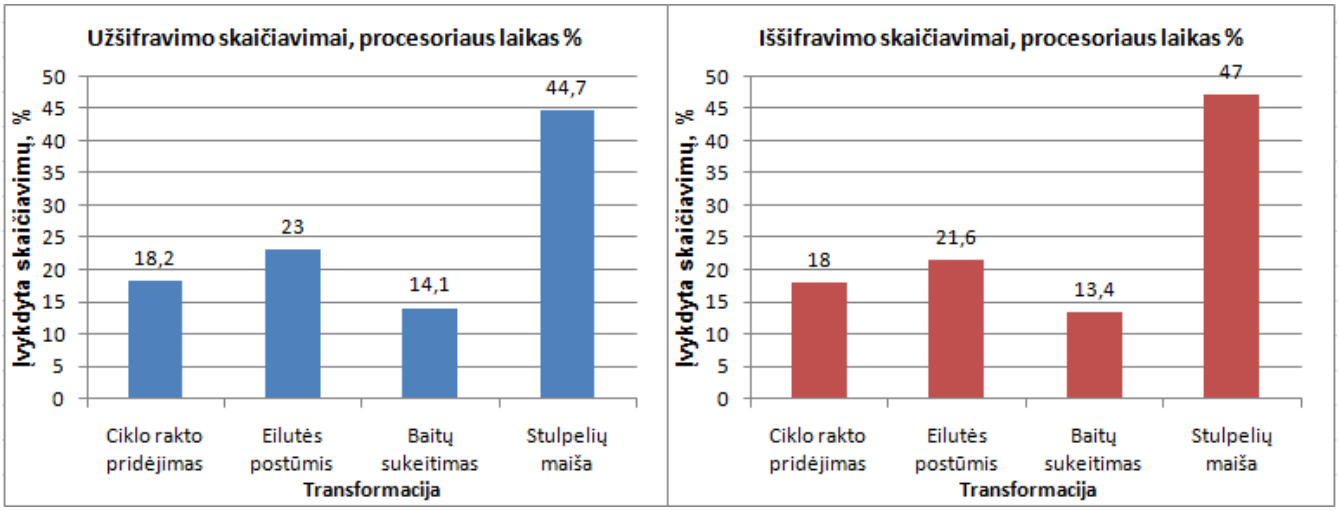
Rijndael(AES), bloko dydis: 128, rakto ilgis: 128, ciklų skaičius: 10	**AND**	**OR**	**Baitų postūmis**	**Kiekvienos transformacijos skaičiavimų apkrovos vidurkis**
Ciklo rakto pridėjimas	320 Ta (17.7%)	160 To (12.0%)	0 (0.0%)	9.9%
Eilutės postūmis	0 (0.0%)	120 To (9.0%)	120 Ts (5.0%)	5%
Baitų sukeitimas	120 Ta (6.6%)	80 To (6.0%)	0 (0.0%)	4%
Stulpelių maiša	1368 Ta (75.7%)	972 To (73.0%)	2304 Ts (95.0%)	81.1%

Kaip matyti iš 6 lentelės, stulpelių maišos transformacija išsiskyria minėtų operacijų poreikiu, todėl jos baterijos energijos sąnaudos turėtų būti didžiausios iš visų lentelėje įvardintų transformacijų.

Visa duomenų saugai užtikrinti sunaudojama energija, skaičiavimų resursai susideda iš sekančių veiksnių:

- Duomenų užšifravimo/iššifravimo;
- Duomenų autentifikavimo;
- Duomenų persiuntimo, priėmimo;
- Sesijos metu, kai atnaujinama sesija, raktai. [11]

13 paveiksle atvaizduojamas transformacijoms sunaudotas mikroprocesoriaus skaičiavimų laikas (%), kai užšifruojamas ir iššifruojamas vienas 128 bitų duomenų blokas, esant 128 bitų rakto ilgiui ir 10 ciklų.



13 pav. Mikroprocesoriaus skaičiavimų laiko sąnaudos (%) užšifravimo ir iššifravimo transformacijoms.

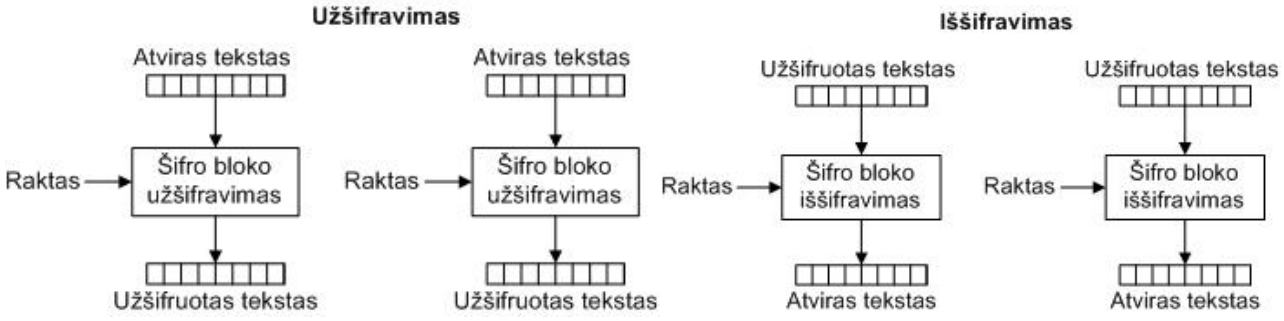
Remiantis [11] tyrimo rezultatais, galima daryti tokias išvadas:

- Eilutės postūmio operacijoms reikia daugiau energijos (procesoriaus skaičiavimo laiko), nei loginėms operacijoms AND, OR;
- Tiek užšifravimo, tiek iššifravimo procese, stulpelių maišos transformacijai reikia bent du kartus daugiau įvykdyti instrukcijų (operacijų), nei bet kuriai kitai transformacijai.
- Iššifravimo proceso metu mikroprocesoriaus skaičiavimų kiekis išauga 20-30% vien dėl to, kad stulpelių maišos inversijos transformacija atlieka keturis (užšifravimo metu tik du) daugybos veiksmus kiekvienam būklės matricos baitui. [11]

2.1.1.3. Rijndael (AES) kriptoalgoritmo režimai

Užšifravimo (iššifravimo) algoritmas įvestyje priima vieną duomenų bloką ir raktą, kuriuo užšifruoja (iššifruoja) duomenis ir juos pateikia išvestyje. Tiek įvestyje, tiek išvestyje duomenų dydis yra toks pats. Taisyklės, aprašančios duomenų užšifravimą (iššifravimą) duomenų blokais, vadinamos režimais. Pavadinkime užšifruotą bloką - „C“, atvyrą tekstą - „P“, raktą - „K“, užšifravimo procesą - „E“, iššifravimo procesą - „D“, o bloko eilės indeksą - „i“.

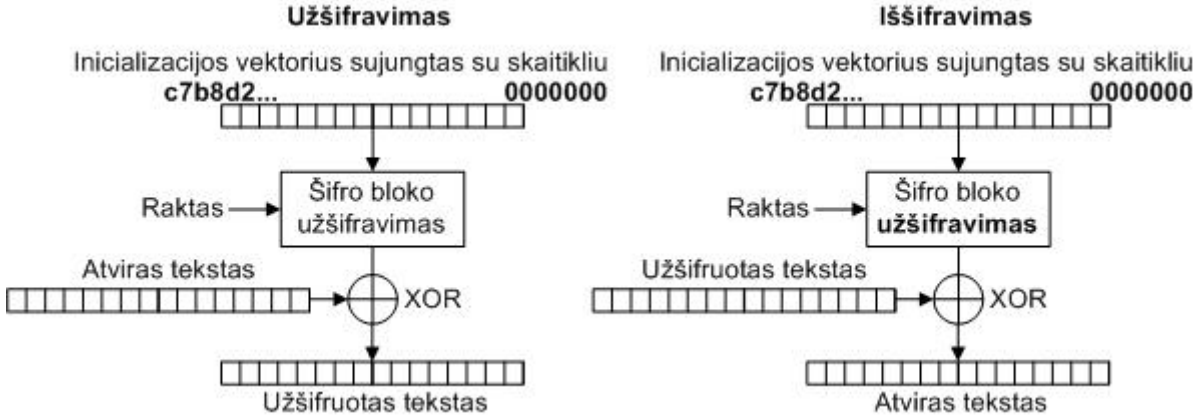
Elektroninio kodo knygos režimas (angl. *Electronic CodeBook mode*) ECB veikia žodyno principu t.y. kiekvienas įvestyje įeinantis atvyro teksto (angl. *plaintext*) duomenų blokas, panaudojant bendrą visiems blokams raktą, užšifruojamas individualiai. Tokia struktūra leidžia įgyvendinti lygiagretų apdorojimą, kurį, iš žemiau minimų režimų, dar geba CTR. Silpnoji vieta - blokų užšifravimas nėra paremtas tarpusavio priklausomybe, todėl įsibrovėlis gali sėkmingai pakeisti bet kurį bloką su anksčiau apdorotu ir nulaužti užšifruotą žinutę net nežinodamas slapto rakto. Ši saugumo spraga dar yra vadinama „blokų nepriklausomybės problema“. Be to, identiško atvyro teksto blokai užšifruojami į identiškus užšifruoto teksto blokus, o, pvz., triukšmo, pažeistas bent vienas bitas, duomenų perdavimo kanale, gali būti sugadintų duomenų priežastimi. [12, 13]



14 pav. ECB režimo veikimo principas užšifruojant ir iššifruojant duomenų bloką.

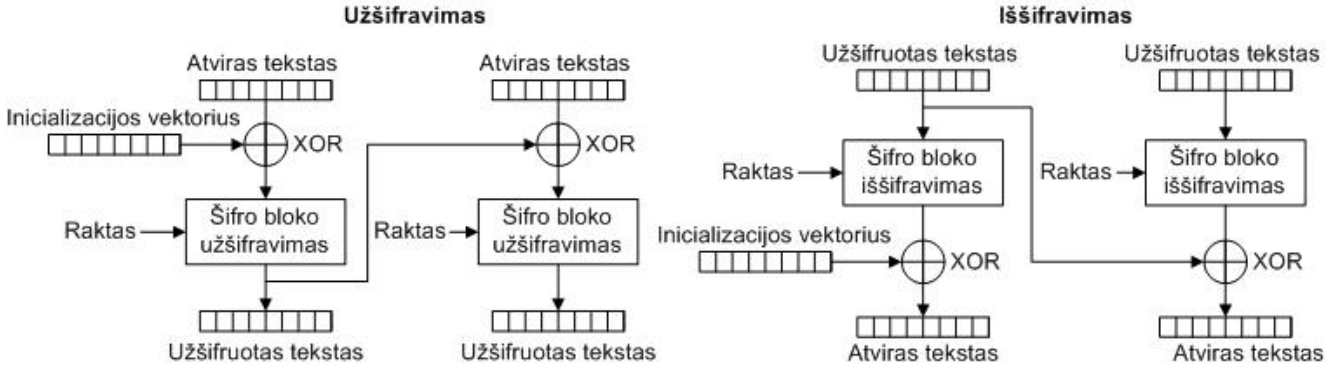
Skaitiklio (angl. *CounTeR*) CTR režime inicializacijos vektorius (IV) įvedamas į duomenų bloką bet kurioje bitų pozicijoje [0.. N], tada šis blokas užšifruojamas bendru raktu. Tokius blokus apjungus su atvyro teksto blokais, panaudojant loginę operaciją XOR, gaunamas užšifruotas tekstas

ir atvirkščiai, tokius blokus apjungus su užšifruoto teksto blokais, gaunami atvyro teksto duomenų blokai. Inicializacijos vektorius sumažina duomenų blokų atkartojimo riziką, kuri yra būdinga ECB režimui, kadangi visi blokai užšifruojami skirtingai, priklausomai nuo IV pozicijos atitinkamame bloke. Blokai gali būti apdorojami lygiagrečiai, o duomenų užšifravimui ir iššifravimui naudojama ta pati techninė įranga. Vieno užšifruoto bloko iškraipymas lemia atitinkamo atvyro teksto bloko iškraipymą. [12]



15 pav. CTR režimo veikimo principas užšifruojant ir iššifruojant duomenų bloką.

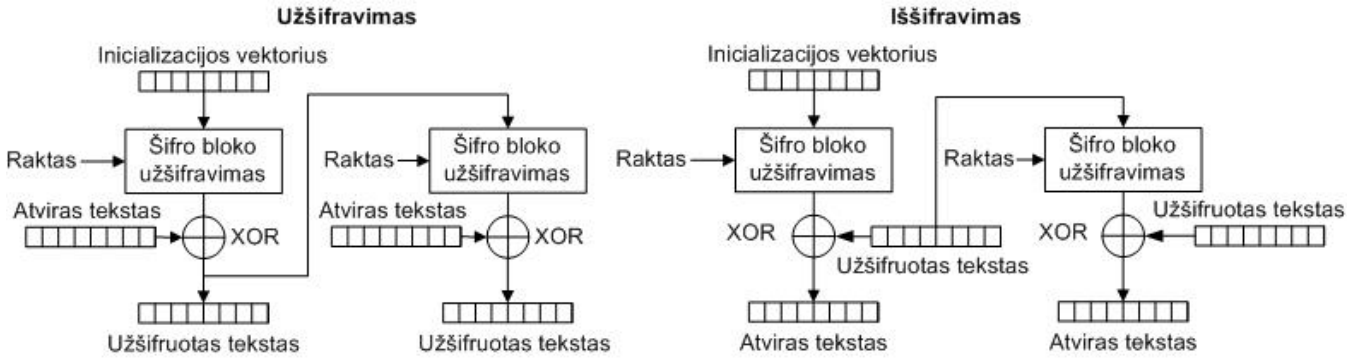
Šifro blokų grandinės režimas (angl. *Cipher block chaining mode*) CBC įveda atsakomąją reakciją (angl. *feedback*), kuri išreiškiama priklausomybe tarp duomenų blokų t.y. prieš kiekvieno atvyro teksto bloko užšifravimą, panaudojant loginę operaciją XOR bitų lygyje, atliekamas šio atvyro teksto bloko apjungimas su prieš tai užšifruotu duomenų bloku. Tai užtikrina, jog net jei ir tekstas susideda iš daug identiškų duomenų blokų, jie visi bus užšifruoti į skirtingo teksto blokus. Šiame režime inicializacijos vektorius yra apjungiamas su pirmu atvyro teksto bloku, panaudojant loginę operaciją XOR bitų lygyje, dar prieš atliekant bloko užšifravimo veiksmus. Pažeistas bitas užkoduoto teksto bloke lemia atitinkamo atvyro teksto bloko sugadinimą. [12, 13]



16 pav. CBC režimo veikimo principas užšifruojant ir iššifruojant duomenų bloką.

Jei bloko indeksas „i“ = 1, tai matematiškai užšifravimą CBC režimui galima išreikšti sekančiai: $C_i = E_K(P_i \oplus C_{i-1})$ atitinkamai, iššifravimas: $P_i = D_K(C_i) \oplus C_{i-1}, kur\ C_0 - inicializacijos\ vektorius.$

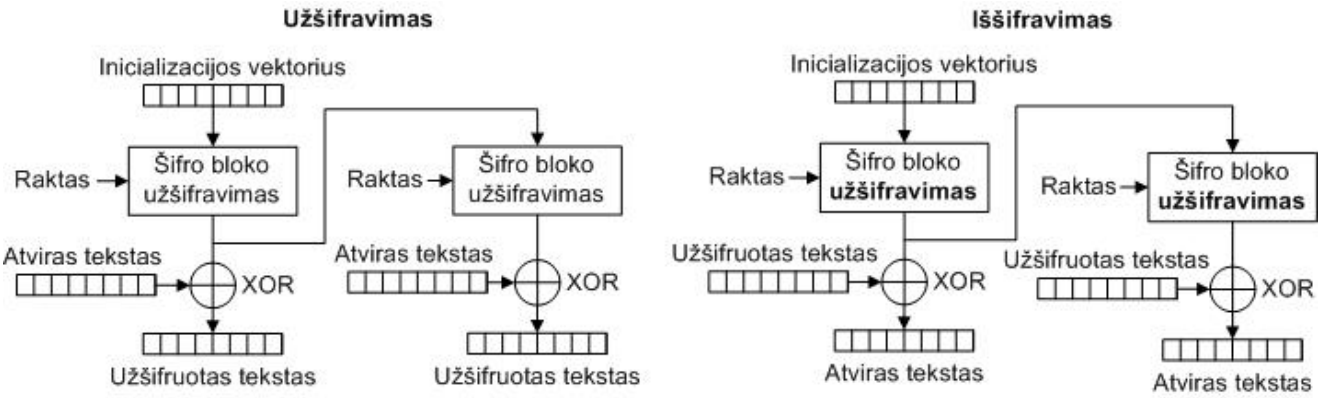
Šifro atsakomosios reakcijos (angl. *Cipher Feedback*) CFB režime vienam užšifravimo proceso ciklui perduodami mažesni duomenų kiekiai, nei vienas duomenų blokas. Kiekvienas užšifruotas duomenų blokas yra priklausomas nuo anksčiau užšifruotų blokų. CFB naudoja poslinkio (angl. *shift*) registrą, kurio ilgis yra vieno bloko ilgio, tačiau padalintas į dalis (skyrius). Pvz., jei bloko ilgis 16 baitų, o vienu metu apdorojamas tik vienas baitas, tuomet poslinkio registras yra padalinamas į šešioliką dalių t.y. vėliausiai užšifruoti kairiausi 8 bitai, panaudojus loginę operaciją XOR, apjungiami su pirmaisias atvyro teksto 8 bitais. Jeigu užšifruotame tekste vienas bitas yra iškraipytas, tuomet iššifravimo procese vienas atvyro teksto bitas taip pat bus iškraipytas, o poslinkio registras taps sugadintu. Tai lems keletos atvyro teksto dalių iškraipymus, kol blogas bitas bus pašalintas iš poslinkio registro. Duomenų blokai užšifruojami, panaudojant vieną slaptą raktą. [12]



17 pav. CFB režimo veikimo principas užšifruojant ir iššifruojant duomenų bloką.

Jei bloko indeksas „i“ = 1, tai CFB bloko užšifravimą (1) ir iššifravimą (2) galima užrašyti taip: (1): $C_i = E_K(C_{i-1}) \oplus P_i$, (2): $P_i = E_K(C_{i-1}) \oplus C_i, kur\ C_0 - inic.vektorius.$

Išeities atsakomosios reakcijos (angl. *Output Feedback*) OFB režimas veikia panašiu principu kaip ir CFB, bet nereikalauja užšifruoto duomenų bloko proceso įvestyje. Čia poslinkio registras (kartu su inicializacijos vektoriumi) užšifruojamas, panaudojant raktą, o gautas rezultatas yra paslenkamas ir naudojamas kaip įvestis sekančiam proceso etapui. Jeigu kiekvieno etapo rezultatas yra apdorojamas ir saugomas, tuomet, panaudojus loginę operaciją XOR, šiuos apdorotus rezultatus galima apjungti su atvyru (užšifruotu) tekstu ir išgauti užšifruotą (atvyrą) tekstą. Suprantama, jei vienas užšifruotų duomenų blokas yra prarandamas, tuomet visi sekantys blokai yra paveikiami, nes apjungiami su klaidingais rezultatų blokais. Vis dėl to, vienas pažeistas užšifruotas duomenų blokas „pagamina“ tik vieną sugadintą žinutės atvyro teksto duomenų bloką. [12]



18 pav. OFB režimo veikimo principas užšifruojant ir iššifruojant duomenų bloką.

7 lentelėje pateiktos mokslininkų [7] išmatuotos Rijndael (AES) visų minėtų, išskyrus CTR, režimų energijos sąnaudos.

7 lentelė. Rijndael-AES režimų energijos sąnaudos, nustatant raktą ir apdorojant vieną bloką duomenų.

Rakto dydis	**Rakto nustatymas (*µJ*)**	**ECB (*µJ/Blokui*)**	**CBC (*µJ/Blokui*)**	**CFB (*µJ/Blokui*)**	**OFB (*µJ/Blokui*)**
128	7.83	1.21	1.62	1.91	1.62
192	7.87	1.42	2.08	2.30	1.83
256	9.92	1.64	2.29	2.31	2.05

Remiantis 7 lentalės duomenimis, galime teigti:

- Energijos sąnaudos išauga atitinkamo rakto nustatymo etape, didėjant rakto ilgiui;
- ECB režimas mažiausiai implus energijos sąnaudoms, tačiau jis yra lengvai pažeidžiamas kriptoanalizės atakų;
- Atsparesni kriptoanalizės atakoms režimai – CBC, OFB, CFB suvartoja gerokai daugiau, daugiausiai CFB. [7]

8 lentelėje pateikiamas aprašytų režimų sugadinamų duomenų kiekis, įsivėlus vieno bito klaidai užšifravimo arba persiuntimo procese. [15]

8 lentelė. Rijndael-AES režimų klaidų toleravimas, duomenų bloke atsiradus vieno bito klaidai.

Klaidos atsiradimo sritis	**Režimas ir sugadinamų duomenų kiekis, esant sugadintam vienam bitui**				
	ECB	**CBC**	**OFB**	**CFB**	**CTR**
Užšifravimas	Vienas blokas	Vienas blokas	Visi duomenys po klaidos atsiradimo	Vienas blokas	Vienas blokas
Duomenų persiuntimas	Vienas blokas	Du blokai	Klaidos netoleruojamos	Du blokai	Klaidos netoreluojamos

Atsižvelgiant į saugumo lygį, energijos sąnaudas ir klaidų toleravimą, optimaliausiu režimu galima laikyti CBC.

2.3. Išvados

- Mobiliųjų įrenginių mikroprocesoriaus sparta, atminties kiekis didėja pagal Moore dėsnį, tačiau baterijos energijos talpai plečiantis nepakankamai sparčiai, naudotojas verčiamas rinktis: ilgesnį įrenginio veikimo laiką ir neapsaugotus duomenis ar saugius duomenis, kurių apsaugai taikomi energijos sąnaudoms imlūs kriptografiniai sprendimai.
- Grėsmės duomenų saugumui dažnai lieka neįvertinamos, anot „iAnywhere“, net apie 30% apklaustų IT profesionalų nešifravo mobiliuosiuose įrenginiuose esančių slaptų duomenų.
- Daugiausiai mikroprocesoriaus skaičiavimų, o tuo pačiu ir baterijos energijos sąnaudų, reikia Rijndael kriptoalgoritmo stulpelių maišos transformacijai.
- Atsižvelgiant į saugumo lygį, energijos sąnaudas ir klaidų toleravimą, optimaliausiu režimu galima laikyti CBC.
- Išanalizavus pagrindinių techninių komponentų veiklos įtaką baterijos energijos sąnaudoms, nuspręsta, jog tikslesniems tyrimo rezultatams gauti, būtina išjungti energijos tiekimą eksperimento metu nenaudojamiems delninio kompiuterio techniniams komponentams.
- Analizuojant mokslininkų straipsnį [28], kuriame pateikti RC2, Blowfish, XTEA ir AES simetrinių blokinių kriptoalgoritmų energijos sąnaudų palyginimai, paaiškėjo, jog dėl būklių matricoje ir sukeitimų matricoje eilės susietų matematinių operacijų, kurios suteikia stiprias kriptografines savybes, AES kriptoalgoritmui prireikė daugiausiai atminties, kompiuterinių skaičiavimų bei energijos sąnaudų.

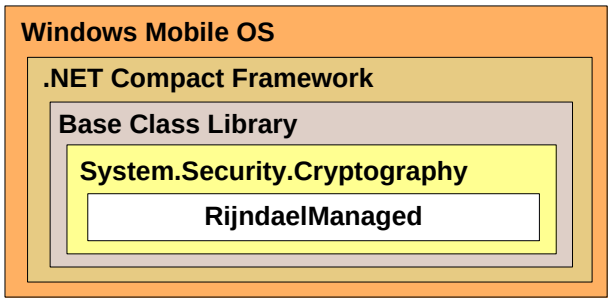
3. RIJNDAEL ENERGIJOS SĄNAUDAS ĮVERTINANČIOS PROGRAMINĖS ĮRANGOS PROJEKTAVIMAS

Tyrimo eksperimentui atlikti yra projektuojama, programuojama taikomoji programa. Numatyta, jog programa atliks etaloninio (angl. *benchmark*) failo (paveikslo) užšifravimo ir iššifravimo procedūras, pagal Rijndael kriptoalgoritmą. Pastarosios procedūros bus įvertintos delninio kompiuterio baterijos energijos sąnaudomis.

3.1. Simetrinio blokinio Rijndael (AES) kriptoalgoritmo stiprumo įtakos energijos suvartojimui tyrimo motyvacija

Šiuolaikiniuose Windows Mobile CE operacinę sistemą turinčiuose delniniuose kompiuteriuose Rijndael (AES) kriptoalgoritmas su 128 bitų bloko dydžiu ir 128 bitų rakto ilgiu yra numatytąsis atminties kortelėje esančių duomenų šifravimo algoritmas. Šis faktas motyvuoja atlikti detalų šio kriptoalgoritmo baterijos energijos sąnaudų tyrimą.

Moderni Microsoft .NET Compact Framework platforma pasižymi efektyviu resursų išnaudojimu, pvz., RAM atmintis nepasiekiama, kol programa nėra įgalinama. Ši platforma puikiai tinka kurti taikomąsias programas, ribotus energijos išteklius ir skaičiavimų resursus turintiems įrenginiams. .NET Compact Framework ir Microsoft Development Network (MSDN) – viešai internete prieinama programavimo sprendimų biblioteka, suteikia galimybę realizuoti tyrimo tikslą. Mūsų tyrime naudojamas Rijndael kriptoalgoritmas MSDN bibliotekoje aprašytas RijndaelManaged skiltyje. Komponentų sluoksnių lygyje RijndaelManaged klasė pasiekiama keliaujant iš išorės į vidų, kaip pateikta 19 paveiksle. [29]



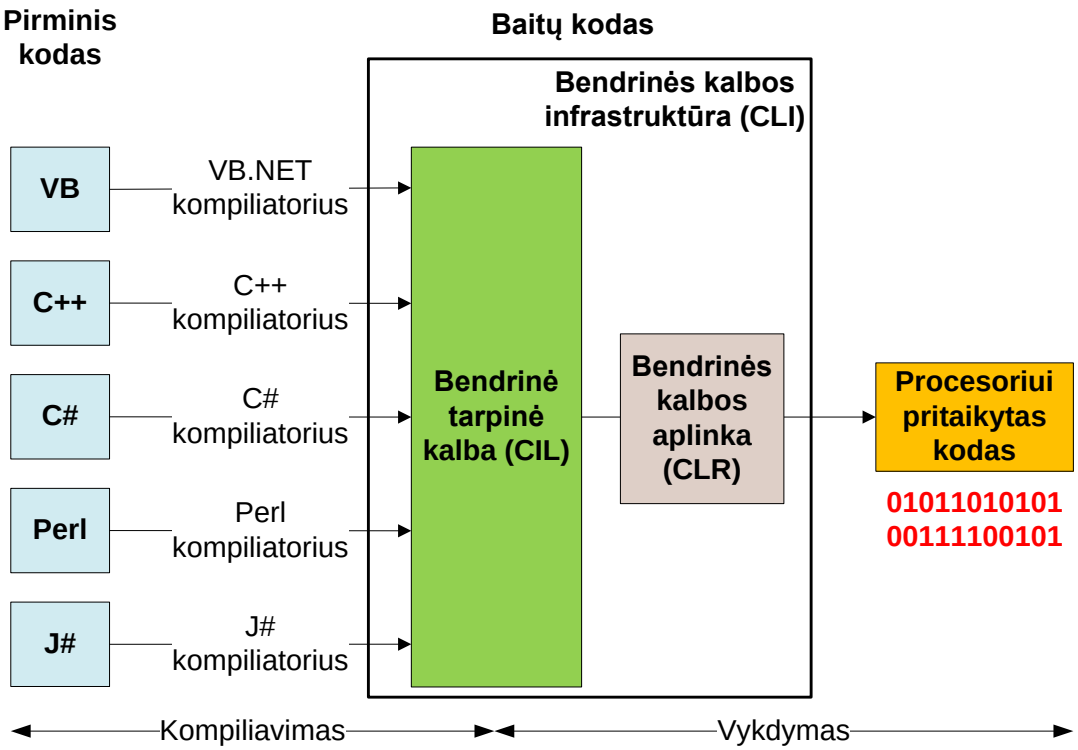
19 pav. Programinių komponentų sluoksniai iki pasiekiama RijndaelManaged klasė.

Programinė įranga, sukurta .NET Compact Framework platformą palaikančiais programavimo įrankiais, pvz., Visual Studio 2008, vykdoma programinėje aplinkoje, kuri tvarko vykdomosios aplinkos (angl. *runtime*) kompiuterinių skaičiavimų reikalavimus. Minėtoji programinė aplinka yra .NET Compact Framework dalis ir yra vadinama „Bendrinės kalbos aplinka“ (angl. *Common language runtime (CLR)*). CLR suteikia programinės įrangos vykdymui virtualią aplinką, kuri programos kūrėjui leidžia nesirūpinti procesoriaus galimybėmis, atminties

paskirstymu ir kitais operacinės sistemos servisais. CLR taip pat apima saugos servisus, pvz., kodo prieigos apsaugą (angl. *code access security)*, patvirtinimą (angl. *validation*), patikrinimą (angl. *verification*), bei klaidų išimčių (angl. *Error exceptions*) apdorojimą.

Bazinės klasės biblioteka (angl. *Base Class Library (BCL)*) yra standartinė biblioteka visoms programavimo kalboms, pvz., C#, Visual Basic, J#, C++, C, kurios naudojamos .NET Compact Framework platformoje. BCL sudaro įvairios funkcijos (metodai), palengvinančios programuotojo darbą, pvz., skaitymas iš failo arba rašymas į jį, grafikos apdorojimas, komunikavimas su duomenų bazių serveriais, manipuliacijos su XML (Extensible Markup Language) dokumentais, o taip pat, kriptografijos mechanizmas.

20 paveikslas vaizduoja .NET Compact Framework platformos sudedamąsias dalis (įvairios programavimo kalbos, kodo kompiliatoriai, bendrinės kalbos infrastruktūra (CLI)), o taip pat, eigą, kurios metu taikomosios programos pirminis kodas apdorojamas iki mikroprocesoriui suprantamo dvejetainio kodo.



20 pav. .NET Compact Framework struktūra ir pirminio kodo apdorojimas.

Minėtoji eiga, susideda iš dviejų etapų:

- Kompiliavimo, kurio metu su .NET suderinamų programavimo kalbų pirminis kodas paverčiamas (kompiliuojamas) į antrinę, nuo platformos nepriklausomą bendrinę tarpinę kalbą (angl. *Common Intermediate Language (CIL)*);

- Vykdymo, kurio metu bendrinės kalbos aplinka (CLR), būdama priklausoma nuo konkrečios kompiuterinės įrangos platformos, kompiliuoja programos CIL kodą į įrenginio mikroprocesoriui suprantamas ir įvykdomas komandas dvejetainio kodo pavidalu.

.NET Compact Framework platforma su Windows Mobile CE operacine sistema užtikrina šiuos suderinamumo kriterijus:

- Suderinamumą su vietiniu (angl. *native*) saugumu;
- Pilną integraciją su įrenginio OS diegimo (angl. *setup*) programomis;
- Sąveiką su įrenginio kodu, panaudojant COM Interop ir kreipinius į platformos DLL (dynamic link library) bibliotekas, iškviečiant reikalingas funkcijas. [29]

Programavimo kalba pasirinkta C#, nes:

- Stipriai paremta tipais, todėl kodo kompiliatorius gali tikrinti kokias reikšmes mūsų kodas priskiria kintamiesiems ir esant nekorektiškoms reikšmėms, apie tai informuoti. Pvz., kompiliatorius neleis, jog sveiko skaičiaus tipui (Integer) būtų priskirtas tekstas.
- Objektiškai orientuota, kas leidžia atlikti duomenų manipuliacijas objektų lygyje.
- Programavimo kalba sukurta taip, kad būtų lengvai suprantama ir optimaliai eksploatuojama, atsižvelgiant į kompiuterinių resursų išteklius. [30]

3.2. Programinė įranga

Programinė įranga yra pasirinkta „Microsoft“ kompanijos, dėl keletos priežasčių:

- Ši kompanija teikia plačią programinės įrangos produkcijos gamą, kuri yra nemokama KTU informatikos fakulteto studentams;
- Laboratorijoje turimas delninukas, skirtas atlikti realiems eksperimentams, turi įdiegtą šios kompanijos operacinę sistemą Windows CE. Ši OS yra gan populiari, todėl atliekamas tyrimas yra pakankamai aktualus;
- Visual Studio 2008 progessional programinis paketas, su kuriuo numatoma realizuoti programą, imituojančią užkardos veiksmus, turi integruotą objektinio programavimo kalbą C#, bei palaiko .NET Compact Framework aplinką;
- Windows Mobile SDK priedas Visual Studio programinį paketą papildo mobiliųjų įrenginių taikomosioms programoms kurti skirtomis bibliotekomis;
- Gautų matavimų rezultatų atvaizdavimui naudojamas Microsoft Office Excel programinis paketas.

Taigi, iš vieno tiekėjo gauname praktiškai visą reikalingą programinę įrangą. Žemiau detaliau aprašomi esminiai programiniai komponentai.

Windows Mobile CE operacinė sistema

Kaip teigiama [31] statistikoje, šios operacinės sistemos 2011 metais užimama rinka yra 5.6%, tačiau 2015 metais jai prognozuojama net 19.5% rinkos dalis. Tai realaus laiko sistema, kuri yra skirta sąlyginai silpnus techninius parametrus (lyginant su nešiojamaisiais kompiuteriais) turintiems mobiliesiems įrenginiams. [17]

Šios operacinės sistemos savybės, susijusios su energijos valdymu:

- Energijos valdymo funkcijas atlieka integruotas programinis modulis - energijos valdytojas (angl. *power manager*);
- Galima atlikti energijos būklės pakeitimus tiek aparatinės įrangos lygmenyje, pvz. sumažinama procesoriaus įtampą, tiek operacinės sistemos sluoksnyje, pvz., sistema pereina į miego būseną;
- Vartotojas informuojamas apie įvykius, susijusius su energijos pokyčiais, pvz., baterija netrukus išsikraus;
- Yra keturios sistemos būsenos: *On, UserIdle, SystemIdle* ir *Suspend*;
- Puikiai suderinama su eksperimento programai kurti skirtais įrankiais.

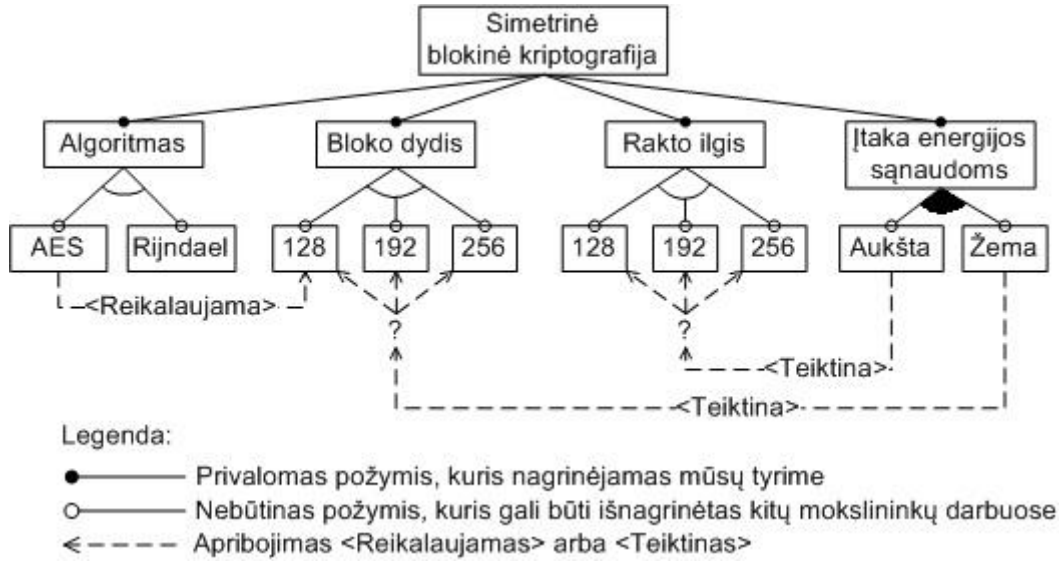
Taikymo sritys: brūkšninio kodo skaneriai, žaidimų konsolės, GPS įrenginiai, delninukai, sumanieji mobilieji telefonai ir t.t.

Visual Studio 2008 Professional

Visual Studio 2008 Professional programinis paketas suteikia galimybę dirbti keliomis objektinio programavimo kalbomis: C#, C++, Visual Basic. Juo galima kurti komandinės eilutės konsoles, interaktyvias vartotojo sąsajas Windows operacinės sistemos taikomosioms programoms, web tinklalapius, web servisus. Taip pat, šis paketas gali būti pildomas įvairiais programiniais priedais, pvz., Windows Mobile Software Development Kit (SDK). Šis priedas Visual Studio paketą papildo dokumentacija, bibliotekomis, kodo pavyzdžiais, įrankiais, skirtais kurti taikomąsias programas Windows Mobile operacinei sistemai. Didžiausias šio priedo privalumas yra tas, kad sukurtas taikomąsias programas, galima testuoti virtualiame mobiliajame įrenginyje - simuliavimo įrankyje, turinčiame Windows Mobile operacinės sistemos aplinką.

3.3. Problemos formulavimas

Siekiant geriau suformuluoti tyrimo problemą, 21 paveiksle sudaryta simetrinių blokinių Rijndael ir AES kriptoalgoritmų požymių (angl. *feature*) diagrama.



21 pav. AES ir Rijndael požymių diagrama. [5]

Pirmiausia, rodyklės su užspalvintais juodais skrituliais nurodo būtinus simetrinės blokinės kriptografijos komponentus ir tuo pačiu savybę, jog komponentų – bloko dydžio ir rakto ilgio daroma įtaka energijos sąnaudoms labai skiriasi.

Detalizuojant komponentus, galima teigti:

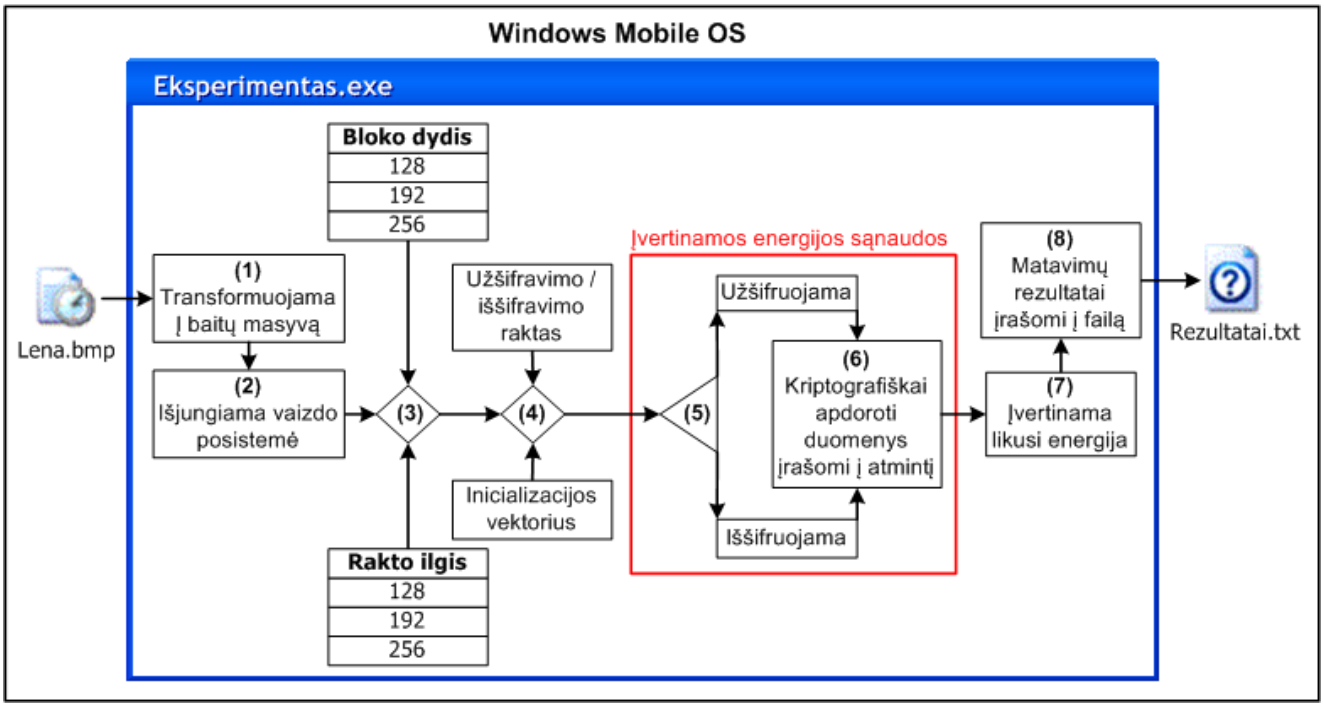
- Alternatyvos tyrimo algoritmui pasirinkti gali būti Rijndael arba AES, tačiau AES, būdamas Rijndael algoritmo poaibiu (FIPS 197 standartas AES algoritmui numato tik vieną 128 bitų duomenų bloko dyd), riboja mūsų siekį – įvertinti sąryšį tarp skirtingų rakto ilgių ir bloko dydžių, energijos sąnaudų atžvilgiu.
- Duomenų, kuriais manipuliuojama, bloko dydžio alternatyvios reikšmės tyrime išskiriamos trys. Galima teigti, jog energijos sąnaudoms šis dydis neturi didelės įtakos t.y. energijos sąnaudos yra mažos, nes skaičiavimai nėra vykdomi, o tik išskyriama vieta atmintyje.
- Tikimąsi, jog delninio kompiuterio baterijos energijos sąnaudas labai įtakos parenkamas rakto ilgis t.y. jį didinant, pastebimai turėtų kristi esamas baterijos energijos lygis.

Žinant, jog rakto ilgis apsprendžia duomenų saugos stiprumą, mūsų tyriamam Rijndael kriptoalgoritmui, galima išskirti tris saugos lygius (profilius): žemas (128 bitai), vidutinis (192 bitai) ir aukštas (256).

Logiška, jog kuo aukštesnis saugos lygis, tuo daugiau prireiks mikroprocesoriaus skaičiavimų ir energijos sąnaudų, tačiau be detalaus tyrimo negalime pagrįstai ir tiksliai nusakyti

kiek procentų baterijos suvartos skirtingi raktų ilgiai, esant skirtingiems bloko dydžiams – štai kur esminė tyrimo problema.

Sprendžiant įvardintą problemą, 22 paveiksle vaizduojama sudaryta eksperimento veiksmų seka ir kriptoalgoritmo energijos suvartojimo įvertinimo schema operacinės sistemos lygmenyje.



22 pav. Eksperimento veiksmų seka ir algoritmo energijos suvartojimo įvertinimo schema OS lygmenyje.

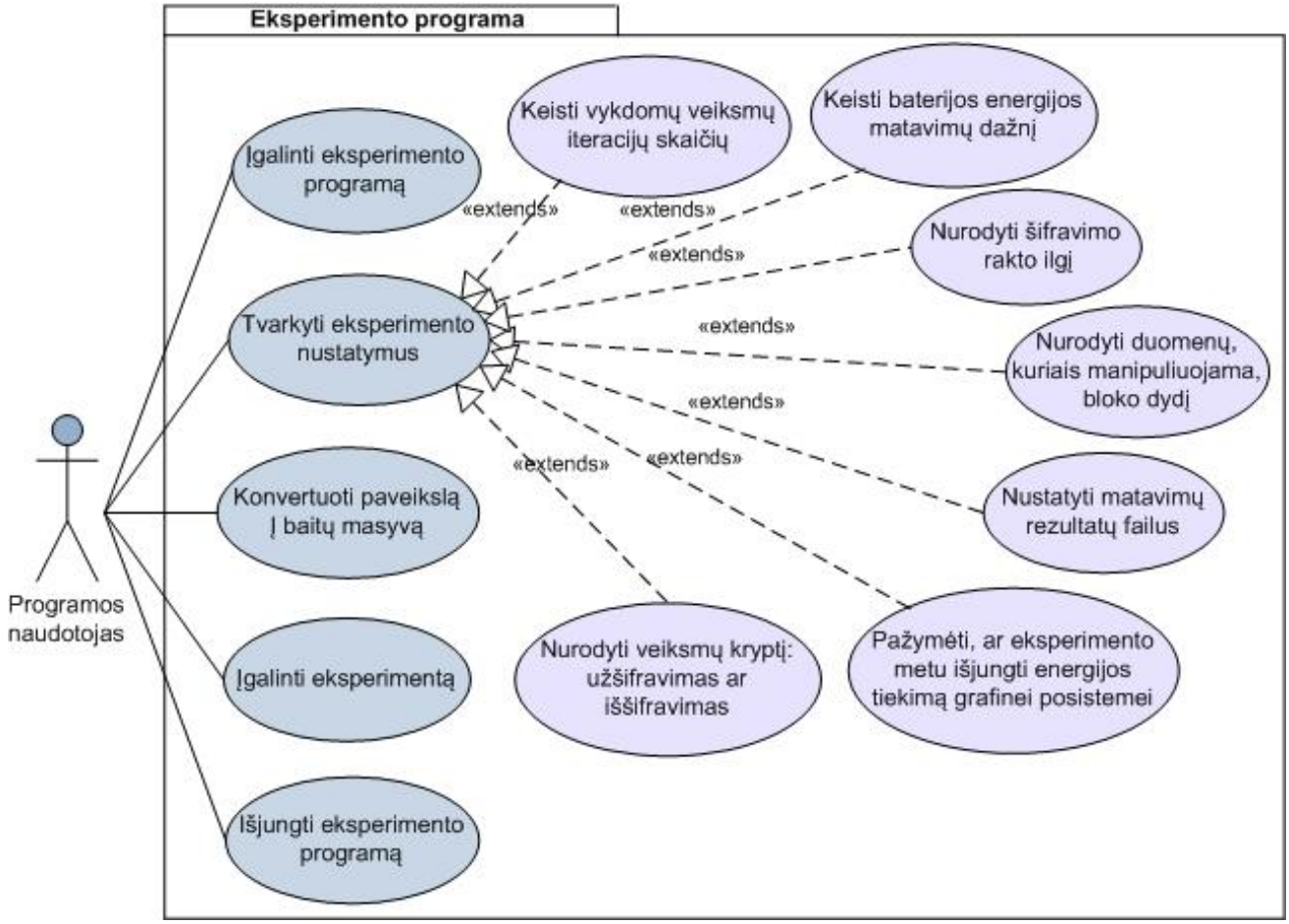
Kaip matyti iš 22 paveikslo, bendru atveju eksperimento veiksmus sudaro 8 etapai:

1. Eksperimento failas „Lena.bmp", tam, kad būtų galima su juo atlikti kriptografinius veiksmus, pirmiausiai transformuojamas į baitų masyvą;
2. Toliau, delninio kompiuterio klaviatūros ir ekrano apšvietimą valdantis modulis išjungia vaizdo posistemę;
3. Parenkamas bloko dydis ir rakto ilgis bitais;
4. Programa pagal parinktą bloko ir rakto dydį automatiškai sugeneruoja užšifravimo/iššifravimo raktą bei inicializacijos vektorių;
5. Priklausomai nuo veiksmų krypties baitų masyvas užšifruojamas arba iššifruojamas;
6. Kriptografiškai apdoroti duomenys įrašomi į delninio kompiuterio vidinę atmintį;
7. Veiksmų, kurie įvykdomi 5. ir 6. etapuose, įtaka delninio kompiuterio baterijos energijos sąnaudoms įvertinama matavimų modulio, kuris integruotas į eksperimento programą dll (dynamic link library) bibliotekos pavidalu;
8. Energijos sąnaudų matavimų rezultatai išvedami į tekstinį failą.

Energijos sąnaudos įvertinamos tik 5. ir 6. etapų, nes mūsų siekis, problemos sprendimas - ištirti kriptoalgoritmo veiksmų (užšifravimo ir iššifravimo) įtaką energijos sąnaudoms bei identifikuoti, kiek svarbu yra įvertinti bloko dydžio ir rakto ilgio tarpusavio priklausomybę kriptografiniuose taikymuose. 1 – 4 etapai yra skirti pasiruošimui, o 7 ir 8 etapas rezultatų fiksavimui.

3.4. Funkciniai reikalavimai sprendimui

Šioje dalyje iškelti reikalavimai eksperimento taikomąjai programai (sprendimui). Įvardinta kokias funkcijas (galimybes) ji turi vykdyti, kad Rijndael šifravimo kriptoalgoritmo energijos sąnaudos būtų pagrįstai įvertintos.



23 pav. Naudojimo atvejų (angl. use case) diagrama programos naudotojo atžvilgiu.

23 paveiksle pavaizduotu atveju, veikėjas yra eksperimento programos naudotojas. Eksperimentui atlikti, iš pradžiu taikomąją programą reikia įgalinti delniniame kompiuteryje.

Paleistoje programoje, turi būti galima keisti, pagal nutylėjimą iš anksto nustatytus eksperimento vykdymui skirtus parametrus:

- Vykdomų veiksmų iteracijų skaičių;

- Nurodyti baterijos energijos matavimų dažnį, šis parametras apsprendžia kiek kartų bus pamatuotas baterijos energijos likutis įgalinus eksperimento vykdymą;
- Nustatyti duomenų bloko dydį bei parinkti tinkamą rakto ilgį;
- Įrašyti matavimams išvesti skirtų failų pavadinimus;
- Pažymėti varnele, ar eksperimento vykdymo metu išjungti grafinės posistemės (ekrano apšvietimą) maitinimą energija – tai duos tikslesnius rezultatus, nes dirbs tik procesorius ir atmintis;
- Nurodyti, kuri Rijndael kriptoalgoritmo veiksmų kryptis (užšifravimas ar iššifravimas) turi būti vykdoma.

Dar prieš eksperimento įgalinimą, etaloninis paveikslas turi būti paverstas (konvertuotas) į baitų masyvą, kadangi blokinis simetrinis Rijndael algoritmas operuoja, pvz., 128 bitų t.y. 16 baitų dydžio duomenų blokais.

Po to, kai nustatymai tenkina, galima paspausti eksperimento vykdymą pradedantį mygtuką. Verta pastebėti, jog eksperimentas automatiškai užsibaigia, po to, kai įvykdo užduotą kiekį iteracinių veiksmų.

Baigęs eksperimentą, naudotojas paspaudžia eksperimento programos išjungimo mygtuką.

3.5. Nefunkciniai reikalavimai sprendimui

Šiame skyrelyje iškeliami minimalūs reikalavimai techninei įrangai, kurie yra būtini, kad eksperimentas būtų įvykdytas sėkmingai. Taip pat, išvardijami reikalavimai ir eksperimento programai.

3.5.1. Reikalavimai techninei įrangai

Tyrime naudojamo delninio kompiuterio procesoriaus spartai, atminties kiekiui ypatingų reikalavimų nėra, nes laikas, sugaištas veiksmų įvykdymui, tyrimui įtakos nedaro. Beabejo, spartesnis procesorius veiksmus įvykdo greičiau, o didesnis atminties (RAM) kiekis, leidžia greičiau atlikti duomenų manipuliacijas.

Delninuke turi būti įdiegta Windows Mobile operacinė sistema (bent 6.0 versija), palaikanti .Net Compact Framework 3.5 aplinką.

Turi būti galimybė prijungti delninį kompiuterį prie stacionaraus kompiuterio per DMA (*direct memory access*) susijungimą. Per jį į delninuką bus įkelta taikomoji programa ir etaloninis lena.bmp failas. Pastarąjį numatyta naudoti, kaip pradinius Rijndael kriptoalgoritmo duomenis eksperimento metu.

Pagrindiniai delninukui keliami reikalavimai yra susiję su baterijos informacijos išgavimu ir energijos valdymų. Tam, kad tyrimas būtų tinkamai vykdomas, delninukas turi tenkinti šias savybes:

- Baterija, neturi būti paveikta senėjimo proceso, dėl ko matavimų rodmenys gali būti iškreipti;
- Palaikomos bent kelios aparatinių komponentu (procesorius, grafinė posistemė) veikimo ir energijos sunaudojimo būsenos (veikimo, būdėjimo, miego, išjungtas);
- Baterijų tvarkyklės, turi leisti programiškai išgauti įvairius parametrus: esamas baterijos energijos lygis, maksimali energijos talpa, likęs baterijos gyvavimo laikas iki visiško jos išsikrovimo ir panašiai.

Tai yra minimalūs reikalavimai delniniam kompiuteriui, tam, kad eksperimentas būtų vykdomas korektiškai.

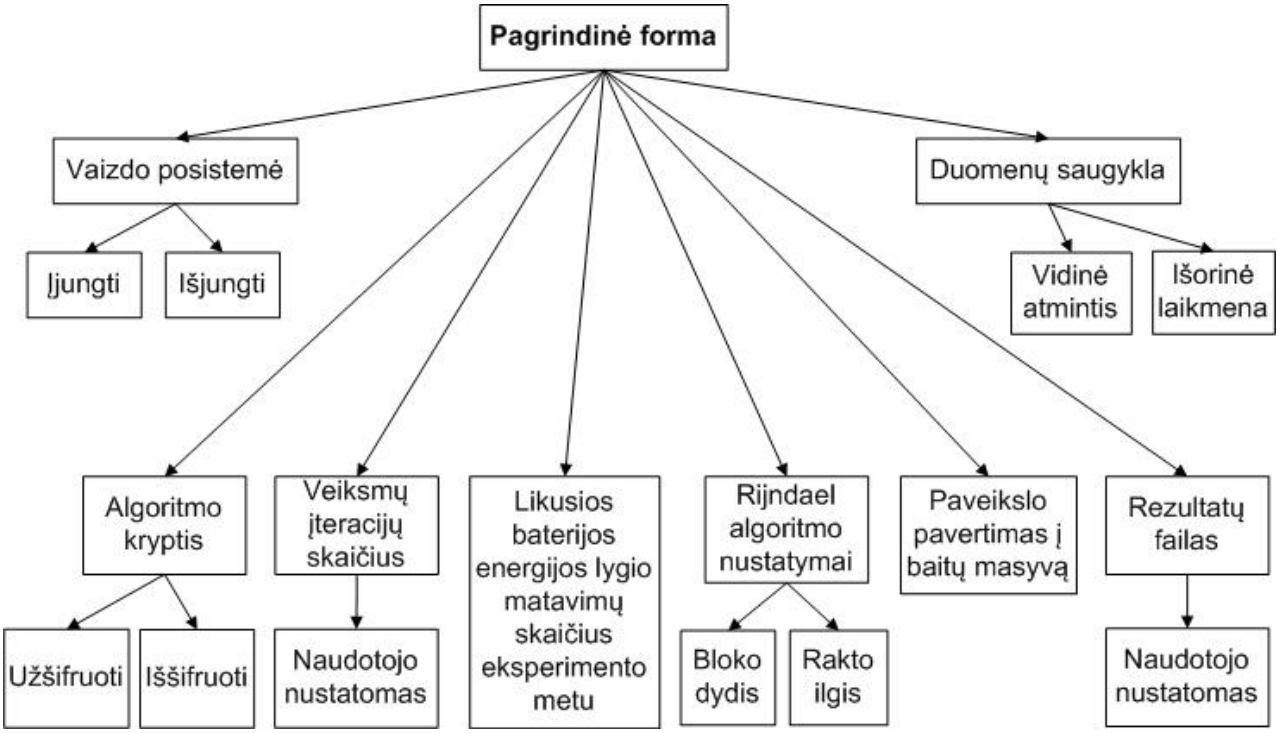
3.5.2. Reikalavimai eksperimento programai

Kuriant eksperimente taikomąją programą, siekiama, jog ji būtų patogi ir suprantama ja besinaudojančiam naudotojui. Ši programa turėtų tenkinti sekančius reikalavimus:

- Turėti aiškų eksperimentui skirtų pradinių duomenų, nustatymų įvedimo langą (sritį);
- Programoje turėtų būti galimybė patikrinti, ar eksperimento programa veikia korektiškai, pvz., etaloninis *lena.bmp* failas sėkmingai paverstas į baitų masyvą;
- Vartotojas turėtų galėti lengvai manipuliuoti įrašais (ištrinti, papildyti), esančiais eksperimento metu nadojamuose failuose;
- Programa turėtų būti apsaugota nuo nekorektiškai įvestų programos naudotojo duomenų, pavyzdžiui, netinkamo formato duomenų įvedimas;
- Programa turėtų būti apsaugota nuo „pakibimo“, dėl eksperimento vykdymo metu atsiradusių klaidų, pvz., buferio perpildymas, kai duomenų kiekis viršija išskirtos atminties kiekį ir iki tol atmintyje buvuios informacijos tam tikra dalis yra sugadinama;
- Vartotojo grafinė sąsaja, turėtų efektingai „bendrauti“ t.y. atsiradus klaidoms, vartotojas turėtų būti informuojamas klaidų pranešimais su paaiškinimais (nuorodomis) kas iššaukė atitinkamą klaidą.

3.6. Tyrimo programos prototipas

Tyrimo programos prototipo struktūros blokinė schema, kurioje matyti programos sudedamosios dalys, pavaizduota 24 paveiksle.

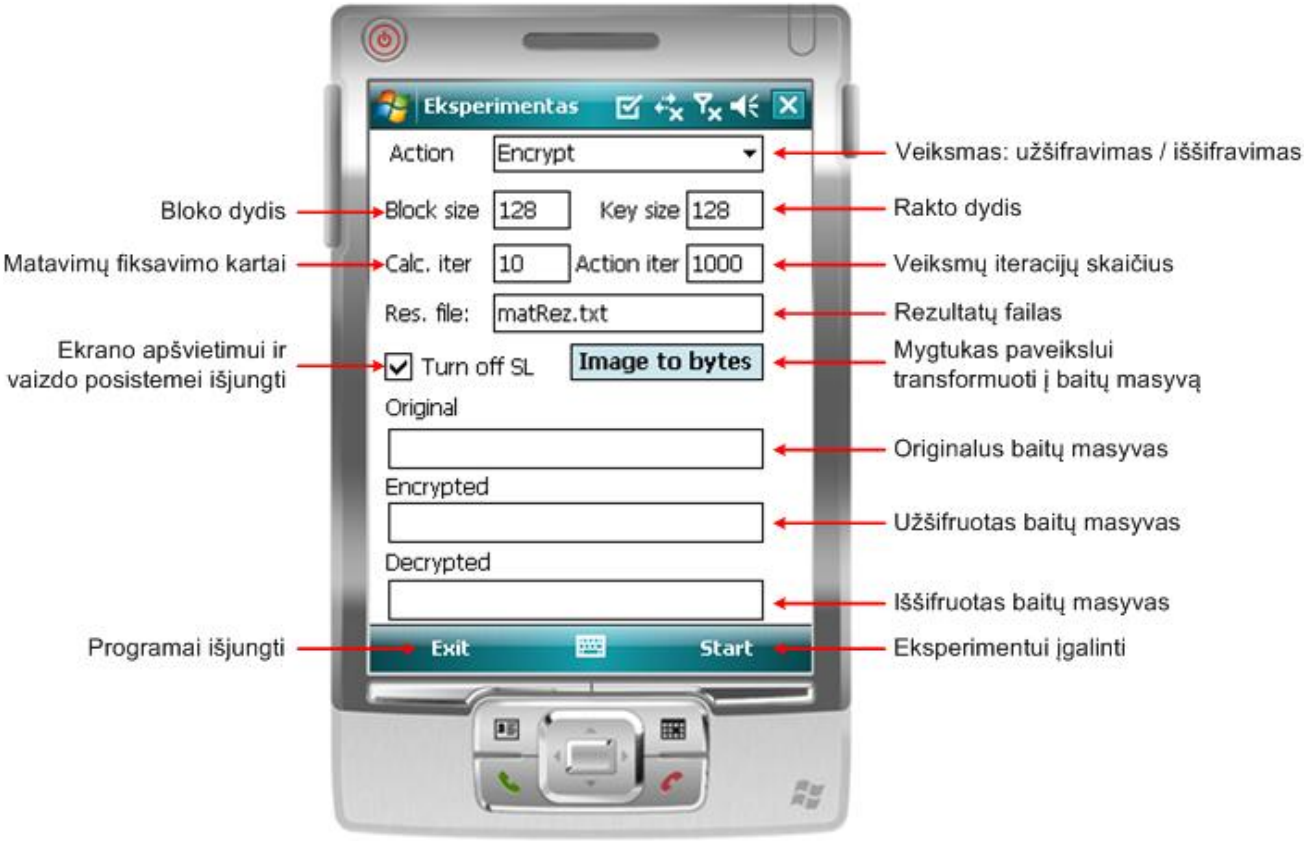


24 pav. Tyrimo programos prototipo blokinė schema.

Eksperimento įrankis (prototipas) suprogramuotas taip, kad visi nustatymai, reikalingi eksperimentui, būtų išdėstyti patogiai ir aiškiai. Atsižvelgiant į tai, jog eksperimento veiksmams vykdyti reikalingas tik mobilaus įrenginio mikroprocesorius bei atmintis, panaudotos papildomos programinės priemonės, įgalinančios programos naudotoją nustatyti, jog eksperimento starto metu būtų išjungiamas energijos tiekimas visai grafinei posistemei. Eksperimentui pasibaigus, grafinei posistemei baterijos energijos tiekimas atstatomas automatiškai. Tai yra būtina tam, kad gauti rezultatai būtų kuo mažiau įtakoti pašalinių, visiškai su tiesioginiais skaičiavimais nesusijusių veiksnių. Įrankis susideda iš sekančių modulių (komponentų):

- Vaizdo grafinės sąsajos apšvietimą valdantis modulis;
- Mobilaus įrenginio likutinės energijos informaciją (procentais, %) nuskaitantis modulis, kuris papildomai fiksuoja matavimų datą (sekundžių tikslumu) ir visą šią informaciją išveda į rezultatų failą;
- Rijndael (AES) kriptoalgoritmo skaičiavimus atliekanti, bei vartotojo grafinę sąsają pateikianti taikomoji programa.

Įrankio išvaizda su reikalingų parametrų įvedimo laukais pateikta 16 paveiksle.



25 pav. Eksperimento programa - prototipas.

Toliau detaliau apibūdinamos 25 paveiksle pateikto įrankio laukų, mygtukų paskirtys.

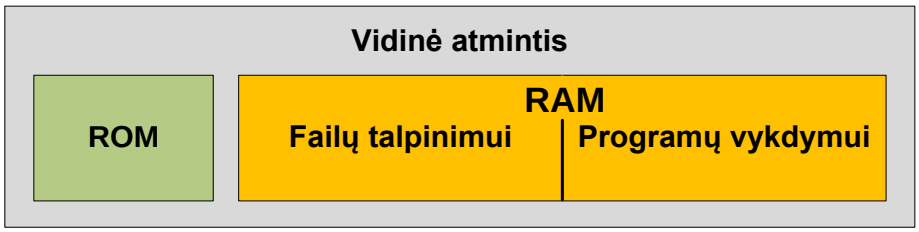
- Prieš atliekant eksperimentą reikia nurodyti veiksmų kryptį t.y. duomenys bus užšifruojami ar iššifruojami (laukas ,,Action");
- Laukuose ,,Block size" ir ,,Key size", nurodomi AES ir Rijndael kriptoalgoritmų parametrai: atitinkamai bloko dydis nurodo, kokio fiksuoto ilgio naudoti ,,String" tipo bitų eilutę, bei rakto dydis bitais, nuo kurio ilgio labai priklauso kriptoalgoritmo atsparumas įvairioms atakoms, pvz., brutalios jėgos atakai (angl. brute force attack);
- ,,Calc. iter" apsprendžia, kiek kartų eksperimento metu bus kreipiamasi į įrenginio baterijos tvarkykles, kad šios pateiktų likusios energijos kiekį, kuris kartu su kitais duomenimis (pvz., data, iteracijų skaičiumi) išvedamas į rezultatų failą;
- ,,Action iter" lauke įrašoma skaitinė reikšmė, kuri nurodo, kiek kartų reikės programai užšifruoti / šifruoti baitų masyvą, kol bus nuskaitoma likusi įrenginio energija. Pavyzdžiui, jei ,,Action iter" reikšmė yra 1000, o „Calc. iter" reikšmė 10, tai kas kiekvieną 1000 iteracinių užšifravimo ar iššifravimo veiksmų atliekamas likusios energijos fiksavimas ir taip daroma 10 kartų t.y. iš viso bus atlikta, pvz., 10 000 iteracinių užšifravimo veiksmų;
- ,,Res. file" leidžia naudotojui nurodyti rezultatų failo pavadinimą skirtingiems matavimams saugoti, rezultatų failas saugomas mobilaus įrenginio vidinėje atmintyje;

- Kadangi šiame eksperimente kriptoalgoritmų operacijos atliekamos su baitų masyvais, mygtuko „Image to bytes" paspaudimas iškviečia funkciją, kuri paveikslą transformuoja į šios struktūros duomenis. Šie veiksmai atliekami dar prieš eksperimentą, kad nedarytų įtakos ieškomiems eksperimento rezultatams;
- „Turn off SL" žymė prieš eksperimentą išsiunčia „0" (nuimta varnelė) arba „1" (uždėta varnelė) reikšmę vaizdo apšvietimą valdančiam moduliui, kuris gavęs reikšmę „1" nutraukia energijos tiekimą į vaizduoklį bei klaviatūrą;
- Trys laukai „Original", „Encrypted" ir „Decrypted" yra informacinio pobūdžio;
- Mygtukas „Start" įgalina eksperimentą, o „Exit" išjungia eksperimento programą.

Toliau paaiškinamas sprendimas, programoje taikyti užšifravimo ir iššifravimo veiksmų iteracijas.

Vidinė delninio kompiuterio atmintis susideda iš ROM (Read Only Memory) ir RAM (Random Access Memory) atminties. ROM atmintis yra pastovioji t.y. į ją negalima rašyti duomenų, galima tik skaityti iš jos, be to, duomenų nuskaitymui nėra būtinas energijos šaltinis. Šioje atmintyje paprastai talpinama delninuko operacinė sistema ir keletas pagrindinių taikomųjų programų, kurios ten įdiegiamos ROM atminties gaminimo metu. [32]

26 paveiksle pateikta delninio kompiuterio vidinės atminties blokinė schema.



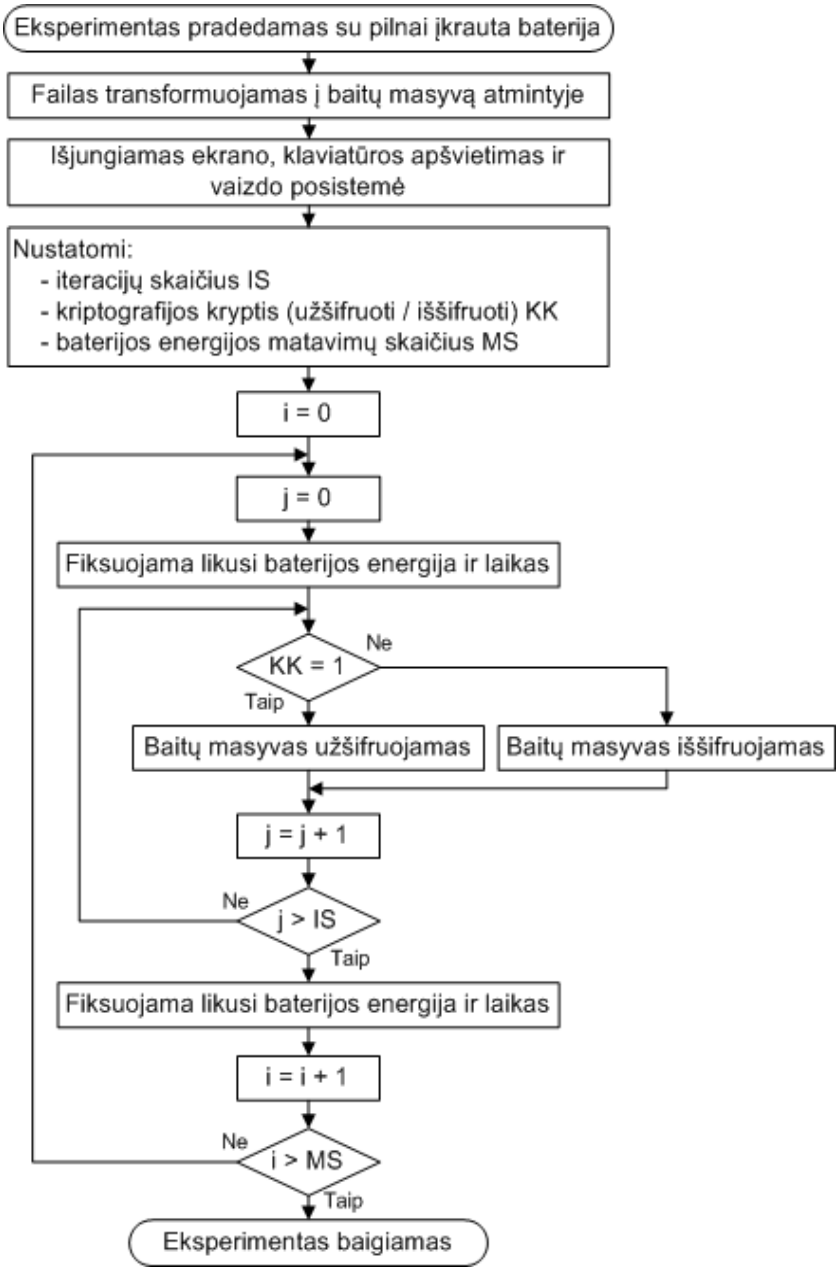
26 pav. Delninio kompiuterio vidinės atminties blokinė schema.

Verta pabrėžti, jog delninio kompiuterio vidinė RAM atmintis yra naudojama ne tik failams talpinti, bet ir programų vykdymui. Tad, kai dokumentuose deklaruojama, jog delninis kompiuteris turi, pvz., 256 MB vidinės RAM atminties, ištikrųjų, asmeninių failų, programų talpinimui jos bus skiriama kur kas mažiau. Pagrindiniai trys faktoriai, lemiantys RAM atminties kiekio sumažėjimą delniniuose kompiuteriuose:

- Operacinė sistema gali užimti dalį RAM atminties savo tikslais, pvz., kodo sekimui (angl. *code shadowing*);
- Net ir visiškai naujas įrenginys RAM atmintyje talpina tam tikrą informaciją, pvz., dokumentų šablonus;
- Kuo daugiau įgalinama programų, tuo daugiau atminties reikia jų vykdymui, to pasekoje dinamiškai sumažėja failų talpinimui skirtos laisvos atminties kiekis. [32]

Beabejo, duomenų talpinimui skirtos atminties kiekį galima praplėsti į įrenginį įdėjus FLASH atminties kortelę, tačiau mūsų eksperimento įvykdymui, net ir jos gali nepakakti. Tam tikslui, eksperimento programoje naudojamas parametras “Action iter”, saugantis mūsų nustatytą užšifravimo ir iššifravimo veiksmų iteracijų skaičių. Veiksmus kartodami kelis tūkstančius kartų su etaloniniu 768 kilobaitų dydžio paveikslu, galime apdoroti kelių gigabaitų dydžio duomenis, turėdami vos kelis šimtus megabaitų atminties.

Detalus eksperimento veiksmų algoritmas vienam taikymui pateiktas 27 paveiksle.



27 pav. Eksperimento veiksmų ir matavimų algoritmo blokinė schema.

Eksperimento vykdymo metu gauti rezultatai saugomi delninio kompiuterio vidinėje atmintyje, tekstiniuose (*.txt), CSV tipo, failuose. Šio tipo failai lengvai importuojami į Microsoft Excel paketą, kuriame skaitiniai duomenys atvaizduojami grafiškai.

9 lentelėje pateikiamos eksperimento metu rezultatų faile įrašomi matuojami dydžiai, informaciniai laukai.

9 lentelė. Rezultatų žurnale (angl. log) pateikiamų dydžių apibrėžimai .

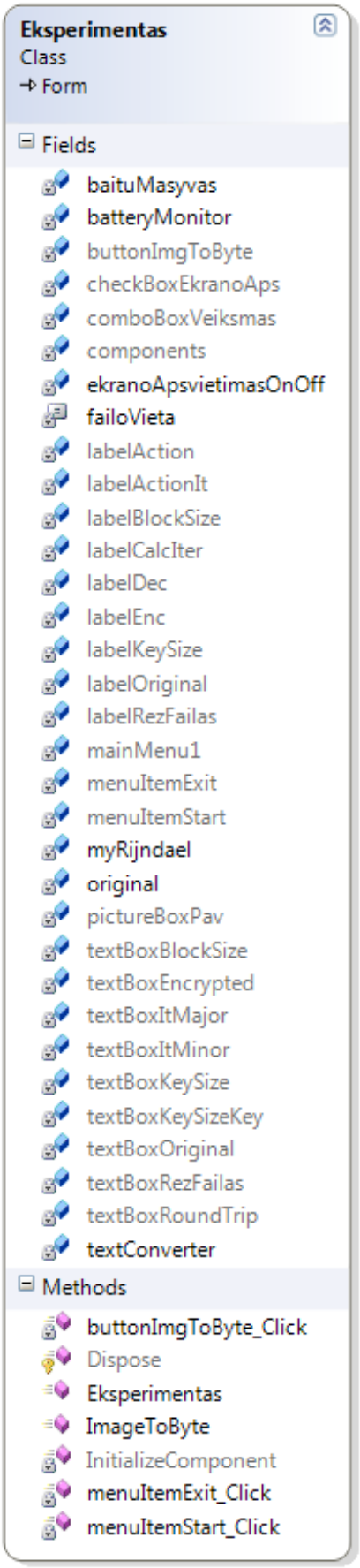
Eil. Nr.	**Kintamojo pavadinimas**	**Aprašymas**
1.	H1	Pateikia iteracinio energijos matavimo eilės numerį
2.	H2	Nurodo įvykusių užšifravimo / iššifravimo veiksmų skaičių
3.	StartDate	Data prieš pamatuojant baterijos energijos lygį (yy:mm:dd)
4.	StartTime	Laikas prieš pamatuojant baterijos energijos lygį (hh:mm:ss)
5.	ACLineStatus	Baterijos būklė (įkraunama/iškraunama)
6.	BatteryLifePercentage	Likęs baterijos energijos lygis (%)
7.	EndDate	Data po baterijos energijos lygio pamatavimo (yy:mm:dd)
8.	EndTime	Laikas po baterijos energijos lygio pamatavimo (hh:mm:ss)

Eksperimento taikymo, kai duomenys iššifruojami, metu suformuoto rezultatų failo pavyzdys pateiktas 28 paveiksle.

```
H1;H2;StartDate;StartTime;ACLineStatus;BatteryLifePercentage;EndDate;EndTime;
Decrypt / mat pries eks.;Baitu: 786486; Ks: 256 , Bs: 128;10.11.06;00:57:18;Offline;100;10.11.06;00:57:18;
1 mat.;Ivyko 600 iter.;/;10.11.06;01:15:06;Offline;93;10.11.06;01:15:06;
2 mat.;Ivyko 1200 iter.;/;10.11.06;01:32:54;Offline;86;10.11.06;01:32:54;
3 mat.;Ivyko 1800 iter.;/;10.11.06;01:50:42;Offline;78;10.11.06;01:50:42;
4 mat.;Ivyko 2400 iter.;/;10.11.06;02:08:30;Offline;71;10.11.06;02:08:30;
5 mat.;Ivyko 3000 iter.;/;10.11.06;02:26:18;Offline;63;10.11.06;02:26:18;
6 mat.;Ivyko 3600 iter.;/;10.11.06;02:44:05;Offline;57;10.11.06;02:44:05;
7 mat.;Ivyko 4200 iter.;/;10.11.06;03:01:54;Offline;49;10.11.06;03:01:54;
8 mat.;Ivyko 4800 iter.;/;10.11.06;03:19:41;Offline;42;10.11.06;03:19:41;
9 mat.;Ivyko 5400 iter.;/;10.11.06;03:37:29;Offline;35;10.11.06;03:37:29;
10 mat.;Ivyko 6000 iter.;/;10.11.06;03:55:17;Offline;28;10.11.06;03:55:17;
```

28 pav. Eksperimento taikymo metu suformuoto rezultatų failo pavyzdys.

Eksperimento programos kintamieji ir metodai yra aprašyti Visual Studio 2008 professional projekto formoje `public partial class Eksperimentas : Form`. Ekrano apšvietimo valdymo (VideoPower.dll) bei baterijos energijos matuoklio (wb.BatteryMonitor.dll) kintamieji ir metodai yra suimportuoti į programos projektą bibliotekų pavidalu. Formos „Eksperimentas“ elementai pateikti klasių diagramoje, 29 paveiksle.



29 pav. Klasės „Eksperimentas“ kintamieji ir metodai.

Detalesnei esminių kintamųjų informacijai pateikti, sudaryta 10 lentelė.

10 lentelė. Rezultatų žurnale (angl. log) pateikiamų dydžių apibrėžimai .

Eil. Nr.	Kintamojo pavadinimas	Tipas	Klasė	Aprašymas
1.	baituMasyvas	Baitas	Eksperimentas	Saugomas konvertuoto etaloninio lena.bmp paveikslo baitų masyvas.
2.	batteryMonitor	Objektas	wb.BatteryMonitor	Turi įsisavinęs visus reikalingus metodus, baterijos parametrams išgauti iš delninio kompiuterio, bei rezultatams pateikti csv (.txt) tipo faile.
3.	ekranoApsvietmasOnOff	Objektas	VideoPower	Objekto metodai VideoPowerOn() ir VideoPowerOff() naudojami įrenginio ekrano posistemei energijos tiekimą įjungti ir išjungti eksperimento metu.
4.	failoVieta	String (baigtinė smbolių seka)	Eksperimentas	Saugomas katalogo adresas, kuriame talpinami eksperimento rezultatai.
5.	myRijndael	Objektas	RijndaelManaged	Saugomi Rijndael kriptoalgoritmo metodai, skirti, pvz., raktui generuoti, bloko dydžiui, rakto ilgiui nustatyti, duomenims šifruoti/iššifruoti ir kiti.
6.	textConverter	Objektas	ASCIIEncoding	Turi metodus, skirtus manipuliuoti su duomenimis, juos paverčiant iš baitų masyvų į string tipo simbolių eilutes ir atvirkščiai.

Eksperimento klasių diagramoje (29 paveikslas) iš įvardintų metodų, galima išskirti kelis:

- `ImageToByte(Image img)` – lena.bmp paveikslą paverčia į baitų masyvą ir jį grąžina aprašytam kintamajam baituMasyvas.
- `menuItemStart_Click(object sender, EventArgs e)` – įgalina eksperimentą, kurio metu sukuriami eksperimento reikšmių saugojimui bei skaičiavimams reikalingi kintamieji, jiems suteikiamos pradinės reikšmės; iškviečiami ekrano apšvietimo posistemei energijos tiekimą išjungiantys (eksperimento pradžioje) ir įjungiantys (eksperimentui pasibaigus), Rijndael kriptoalgoritmo veiksmus vykdantys ir baterijos energijos sąnaudas įvertinantys, bei gautus matavimus į rezultatų failą įrašantys metodai.

Eksperimento metu gautų energijos sąnaudų vs (versus) kriptoalgoritmo šifravimo stiprumas matavimų geriausiems sprendiniams rasti, bus taikomas „Pareto optimal“ metodas. Tegul E galimų pasirinkimų rinkinys, kur $e_{ij} = f(b_i, k_j), e_{ij} \epsilon E$ yra pasirinkimas, priklausantys nuo dviejų kriterijų: b_i (duomenų bloko dydžio) ir k_j (rakto ilgio), nuo kurių dydžio priklauso Rijndael (AES) algoritmo šifravimo stiprumas bei energijos sąnaudos. Tegul Y yra E poaibis, kur $y_j = \min_{b_i} f(b_i, k_j), y_j \epsilon Y$. Tuomet Y yra „Pareto optimal“ sprendinių rinkinys iš rinkinio E. [33]

3.7. Išvados

- Kadangi eksperimente naudojamo delninio kompiuterio operacinė sistema yra Windows Mobile, tyrimo programai realizuoti pasirinkti Microsoft kompanijos sprendimai ir įrankiai, leidžiantys pasiekti puikų programinį suderinamumą.
- Požymių diagramoje suformuluota tyrimo problema, išryškino tris Rijndael algoritmo saugos lygius (profilius), priklausančius nuo rakto ilgio: žemas (128 bitai), vidutinis (192 bitai) ir aukštas (256).
- Suformuluoti funkciniai ir nefunkciniai reikalavimai eksperimento programai ir nefunkciniai reikalavimai techninei įrangai, apibrėžė funkcionalumą programai, nustatė, kokius reikalavimus turi tenkinti techninė įranga, kad eksperimento metu gauti rezultatai būtų korektiški, neiškraipyti.
- Sukurtas eksperimento programos prototipas, detalizuotos klasės, metodai, rezultatų kintamieji bei eksperimento taikymų algoritmas, įgalina atlikti tyrimo eksperimentą.
- Aprašytas „Pareto optimal“ metodas, kuriuo remiantis, po tyrimo eksperimento bus rastas geriausias santykis tarp energijos sąnaudų ir Rijndael kriptoalgoritmo stiprumo.

4. EKSPERIMENTINIS DELNINIO KOMPIUTERIO ENERGIJOS SĄNAUDŲ TYRIMAS

Šioje skiltyje pristatomas eksperimentui suprogramuotas įrankis, aprašoma veiksmų eiga (strategija), taikoma AES (Advanced Encryption Standard) ir Rijndael kriptoalgoritmų energijos suvartojimui išmatuoti. Reikia pastebėti, jog didelis dėmesys skiriamas kriptoalgoritmų parametrų (bloko dydis, rakto ilgis) tarpusavio priklausomybės identifikavimui.

Eksperimente naudojamas ASUS P750 delninis kompiuteris, pasižymintis sekančiomis charakteristikomis: procesorius Intel PXA270 520 MHz CPU, atmintis 256 MB RAM, operacinė sistema Windows Mobile © 6 Professional CE OS, baterija Li-Ion, kurios talpa 1300 mAh.

4.1. Tyrimo metodika

Tyrimui pasirinktas failas – etaloninis (angl. *benchmark*) paveikslas, kuris pasiekiamas [23] literatūros šaltinio adresu ir atvaizduotas 30 paveiksle.

30 pav. Eksperimento etaloninis failas (paveikslas) lena.bmp.

Eksperimentas pradedamas su pilnai įkrauta įrenginio baterija, o kriptoalgoritmo veiksmai atliekami dviem kryptimis – užšifravimo ir iššifravimo. Prieš tyrimo eksperimentą atlikti preliminarūs matavimai:

- Eksperimento programos veiksmų, nesusijusių su duomenų užšifravimu (iššifravimu), energijos sąnaudų matavimai. Gautuose rezultatuose baterijos energijos lygis nesumažėjo nė vienu procentu, vadinasi, Rijndael kriptoalgoritmo tyrimo rezultatai yra korektiški, nes pašaliniai veiksniai nedaro pastebimos įtakos.
- Matavimai, kurie padėjo nustatyti optimalų užšifravimo ir iššifravimo veiksmų iteracijų skaičių, kuriam esant įrenginio baterijos energijos lygmuo buvo arti, bet dar nesiekė kritinės ribos.

Eksperimento metu labai svarbu nepasiekti baterijos energijos kritinės ribos, nes ją pasiekus, operacinė sistema įgalina energijos taupymo režimą, kuris įtakoja procesoriaus spartą. To pasekoje matavimuose atsiranda nekorektiškų duomenų.

11 lentelėje pateiktos eksperimento failo savybės, optimalus iteracijų skaičius bei nurodytas eksperimento metu apdorojamas duomenų kiekis su kiekvienu galimu taikymų variantu.

11 lentelė. Eksperimento failo savybės, veiksmų iteracijų skaičiu, apdorojamas duomenų kiekis.

Pavadinimas	**Dydis**	**Užšifravimų / iššifravimų iteracijų skaičius**	**Apdorojamas duomenų kiekis su kiekvienu taikymų variantu**
Lena.bmp	786486 B = 768 KB	6000 (10x600)	768 * 6000 = 4500 MB = 4,3945 GB

Reikia paminėti, jog 11 lentelėje pateiktas iteracijų skaičius gaunamas 11 kartų fiksuojant baterijos energijos likutį. Viskas daroma tokia eiga:

- Dar prieš atliekant kriptografinius veiksmus, pirmiausiai pamatuojamas ir fiksuojamas baterijos energijos lygis – tai, tarsi, atskaitos taškas.
- Toliau, kas 600 iteracinių kriptografijos algoritmo užšifravimo arba iššifravimo procedūrų, 10 kartų fiksuojami matavimų duomenys. Taip gaunamas bendras iteracijų skaičius 6000.

Eksperimento taikymų variantai, priklausantys nuo kriptoalgoritmo krypties, duomenų, kuriais operuojama, bloko dydžio bei rakto ilgio, pateikti 9 lentelėje.

12 lentelė. Eksperimento taikymų variantai.

Eil. Nr.	**Rakto ilgis, bitai**	**Bloko dydis, bitai**	**Kryptis**
1.	128	128	Užšifruoti
2.	192		
3.	256		
4.	128	192	
5.	192		
6.	256		
7.	128	256	
8.	192		
9.	256		
10.	128	128	Iššifruoti
11.	192		
12.	256		
13.	128	192	
14.	192		
15.	256		
16.	128	256	
17.	192		
18.	256		

4.2. Tyrimo rezultatai

Eksperimento įrankiu išmatuotos baterijos energijos sąnaudų vertės, priklausančios nuo kriptoalgoritmo parametrų dydžių, pateiktos skaitiniu ir grafiniu pavidalu. Eksperimentas buvo atliktas du kartus, todėl gautos skaitinės reikšmės suvidurkintos.

4.2.1. Skaitiniai rezultatai

Eksperimento metu gauti skaitiniai tyrimo duomenys pateikti 13-14 lentelėse.

13 lentelė. Rijndael kriptoalgoritmo užšifravimo tyrimo duomenys.

Bloko dydis, bitai	Rakto ilgis, bitai	Sugaištas laikas (visoms veiksmų iteracijoms) hh:mm:ss	Suvartota baterijos energija, %	Apdorotas duomenų kiekis vienam % energijos, MB
128	**128**	**01:59:57**	**55**	**81,81**
128	192	02:18:03	63	71,43
128	256	02:37:07	71	63,38
192	128	02:18:32	60	75
192	**192**	**02:18:37**	**60**	**75**
192	256	02:20:21	68	66,17
256	128	02:18:27	66	68,18
256	192	02:25:55	66	68,18
256	**256**	**02:25:47**	**66**	**68,18**

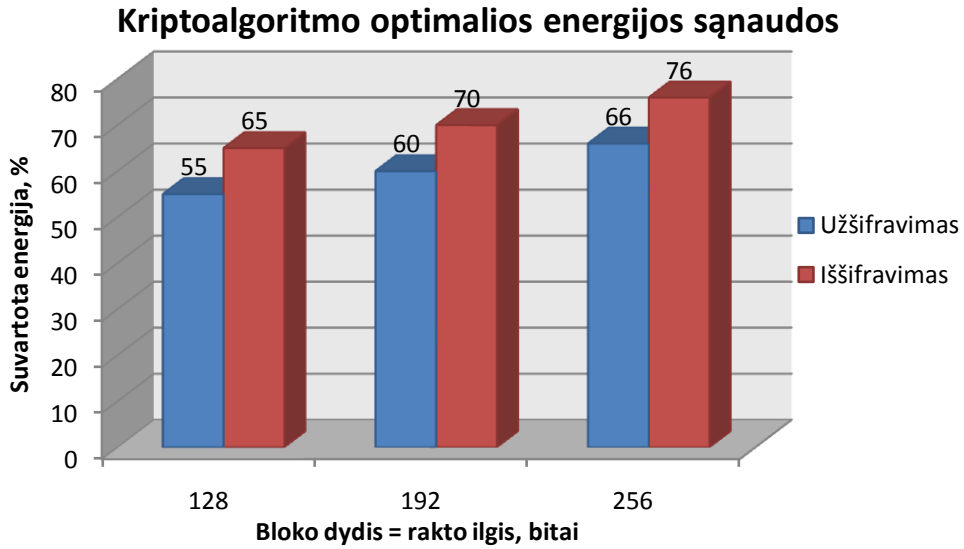
14 lentelė. Rijndael kriptoalgoritmo iššifravimo tyrimo duomenys.

Bloko dydis, bitai	Rakto ilgis, bitai	Sugaištas laikas (visoms veiksmų iteracijoms) hh:mm:ss	Suvartota baterijos energija, %	Apdorotas duomenų kiekis vienam % energijos, MB
128	**128**	**02:25:17**	**65**	**69,23**
128	192	02:39:16	72	62,5
128	256	02:57:48	81	55,55
192	128	02:34:38	70	64,28
192	**192**	**02:34:21**	**70**	**64,28**
192	256	02:53:55	79	56,96
256	128	02:48:04	76	59,21
256	192	02:48:14	76	59,21
256	**256**	**02:48:06**	**76**	**59,21**

Tyrimo problemos sprendimui – rasti optimalias energijos sąnaudas kriptoalgoritmo bloko dydžio ir rakto ilgio atžvilgiu, taikome „Pareto optimal“ metodą (žiūr. sk. 3.6.). Iš gautų skaitinių rezultatų matyti, jog stiprinant kriptoalgoritmą (pvz., taikant didesnį rakto ilgį) labiau sekinama įrenginio baterija t.y. vieno parametro charakteristikų gerinimas reiškia kito parametro charakteristikų bloginimą. „Pareto optimal“ sprendiniais galima laikyti pilka spalva nuspalvintų eilučių duomenis.

4.2.2. Grafiniai rezultatai

31 paveiksle palyginamos energijos sąnaudos taikymams, kurie atitinka „Pareto optimal” metodo sprendinius – 13 ir 14 letelėje šie sprendiniai užspalvinti pilka spalva. Visų trijų taikymų energijos suvartojimo skirtumas tarp užšifravimo ir iššifravimo yra apie 10%.

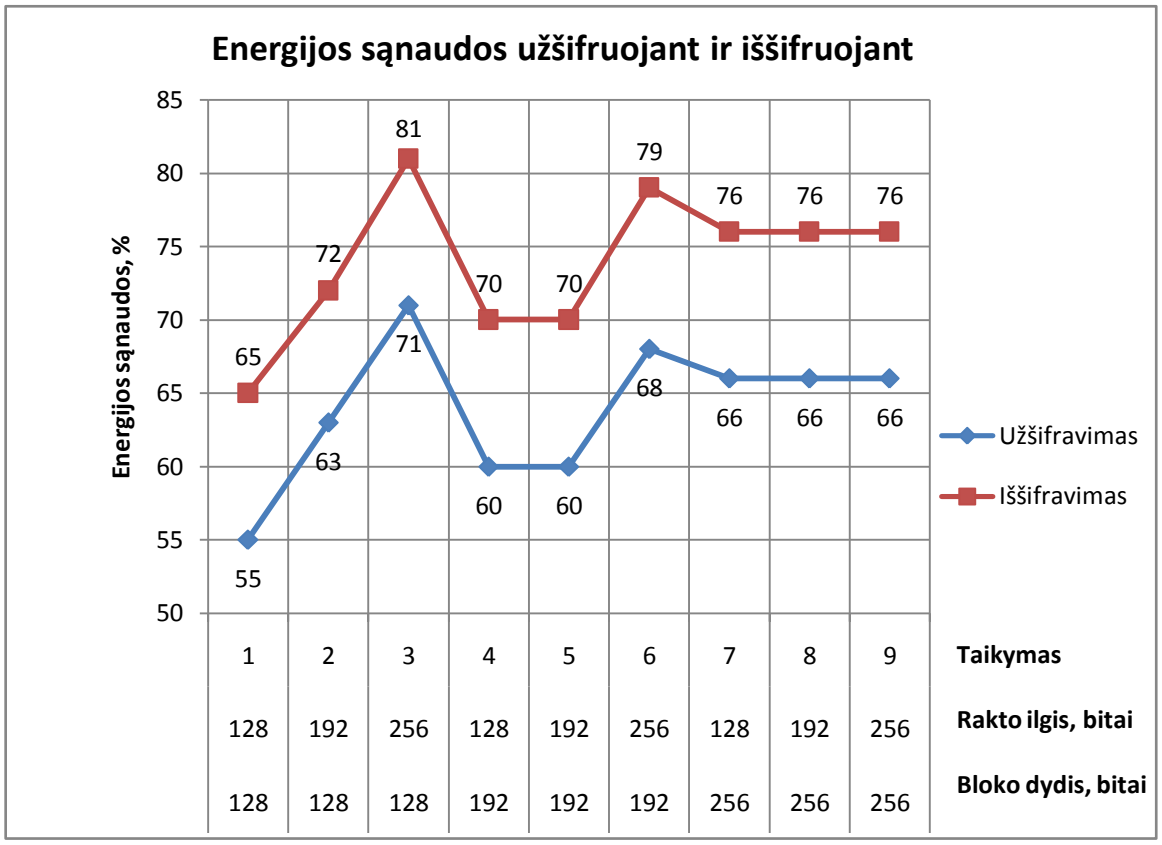


31 pav. Kriptoalgoritmo optimalios energijos sąnaudos, remiantis „Pareto optimal“ metodu.

Pagal energijos sąnaudas ir šifravimo stiprumą, remiantis 31 pav. ir „Pareto optimal” metodu, galima išskirti tris delninio kompiuterio naudotojo profilius:

- Mažos energijos sąnaudos/maža duomenų apsauga, kai rakto ir bloko dydis yra 128 bitai – iki šiol profilis laikomas saugiu, tačiau teoriškai kriptoanalizė gali būti įvykdyta;
- Vidutinės energijos sąnaudos/vidutinė duomenų apsauga, kai rakto ir bloko dydis yra 192 bitai – profilis tinkamas itin aukšto saugumo informacijos apsaugai; duomenų užšifravimo ir iššifravimo veiksmams bendrai suvartojama ~10% energijos daugiau, nei esant žemo saugumo profiliui;
- Aukštos energijos sąnaudos/aukšta duomenų apsauga, kai rakto ir bloko dydis yra 256 bitai – profilis tinkamas aukščiausio saugumo informacijos apsaugai; suvartojama ~12% energijos daugiau, nei esant vidutinio saugumo profiliui.

Kaip matyti 32 paveiksle, daugiausiai energijos suvartojo (užšifravimui 71%, iššifravimui 81%) taikymas nr. 3. Mažiausiai energijos suvartojo taikymas nr. 1, su 128 bitų bloko ir rakto dydžiais - užšifravimui 55%, iššifravimui 65%. Reikia pabrėžti, jog bendrai eksperimente energijos sąnaudų skirtumas tarp užšifravimo ir iššifravimo veiksmų svyruoja nuo 9% iki 11%.



32 pav. Visų taikymų energijos sąnaudos.

Daugiausiai energijos suvartojęs taikymas nr. 3, logiška, buvo ilgiausiai vykdomas – 2 valandas 37 minutes ir 7 sekundes užifruojant ir beveik 3 valandas iššifruojant duomenis (33 paveikslas). Mažiausiai laiko prireikė 1 taikymui, taigi, jis buvo ir sparčiausias: 1 valanda 59 minutės ir 57 sekundės užšifruojant ir 2 valandos 25 minutės ir 17 sekundžių iššifruojant duomenis.

Vykdymo laiko skirtumas tarp užšifravimo ir iššifravimo procedūrų:

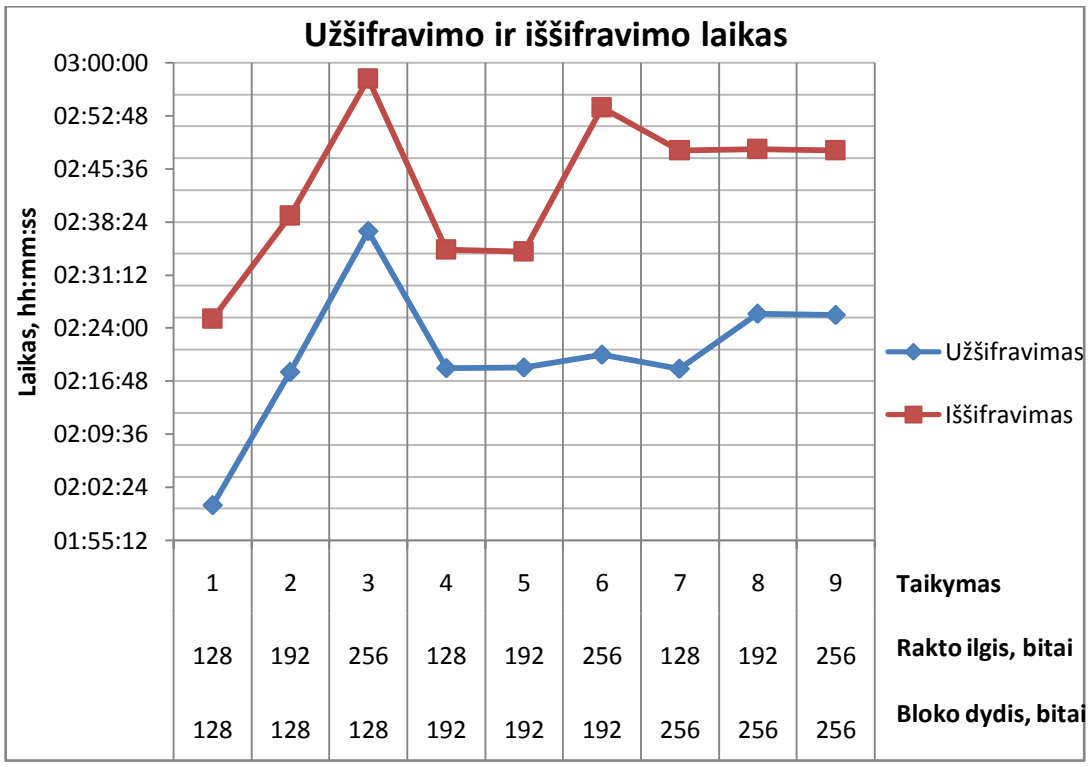
- Didžiausias yra taikymui nr. 6, skirtumas 33 minutės ir 34 sekundės.
- Mažiausias taikymų, kurių numeriai yra 4, 5 – skirtumas apie ~16 minučių. Įdomu ir tai, jog užšifravimo bei iššifravimo laikai yra labai panašūs tarp taikymų 4 ir 5 bei 8 ir 9.

Didžiausias laiko kreivės šuolis:

- Užšifruojant - praktiškai identiškas laiko šuolis tarp 1 ir 2 bei 2 ir 3 taikymų.
- Iššifruojant - didžiausias šuolis tarp 5 ir 6 taikymo.

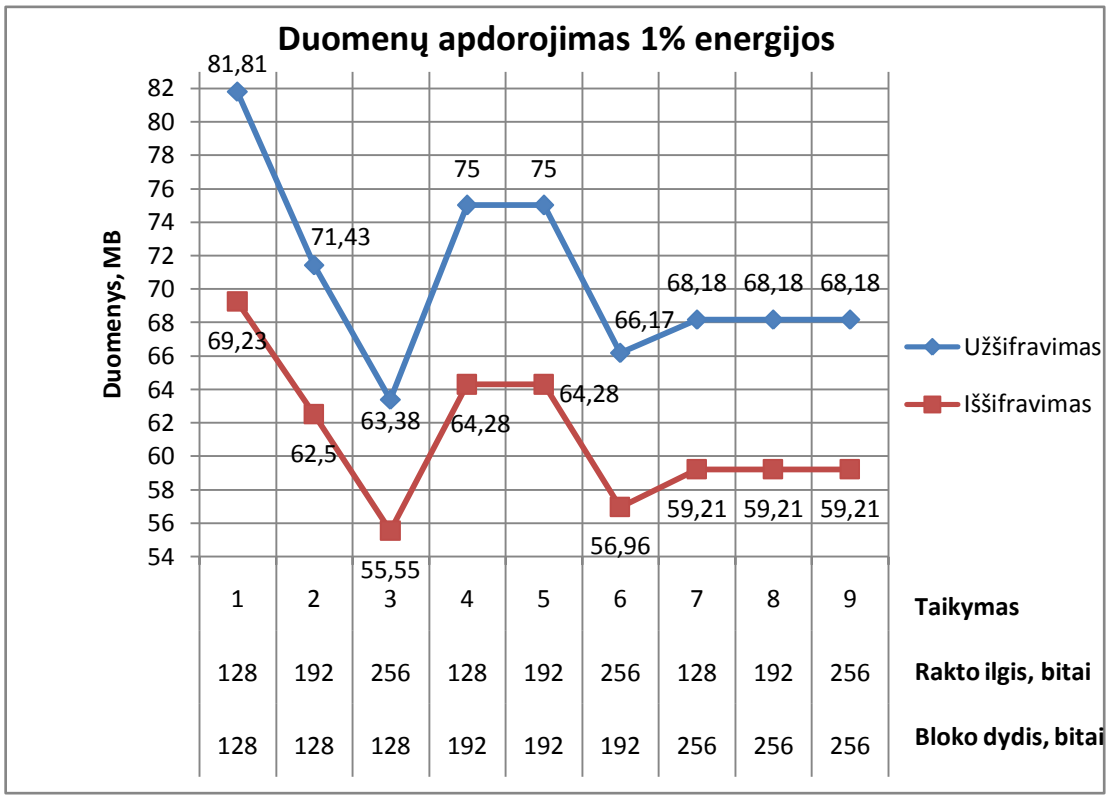
Didžiausias laiko kreivės nuolydis:

- Tiek užšifruojant, tiek iššifruojant - tarp 3 ir 4 taikymo.



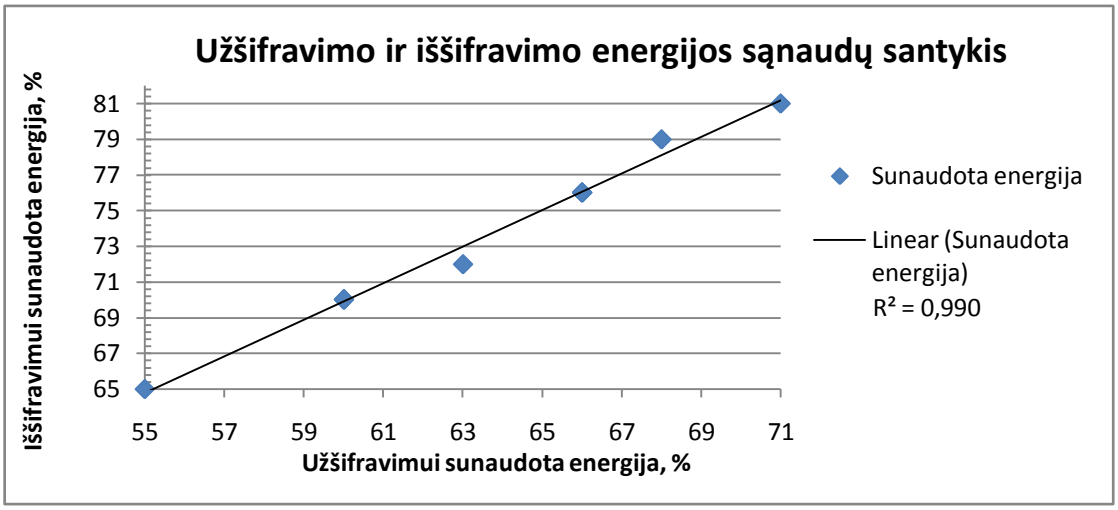
33 pav. Taikymų užšifravimo ir iššifravimo laikas.

34 paveiksle parodytame grafike matyti, jog daugiausiai duomenų apdorojo vienam % baterijos energijos taikymas nr. 1 (užšifravimo metu 81,81 MB, iššifravimo 69,23 MB). Mažiausiai - taikymas nr. 3, šis užšifravimo metu apdorojo 18,43 MB (22,53%), o iššifravimo metu 13,68 MB (19,76%) mažiau duomenų, nei taikymas nr. 1.



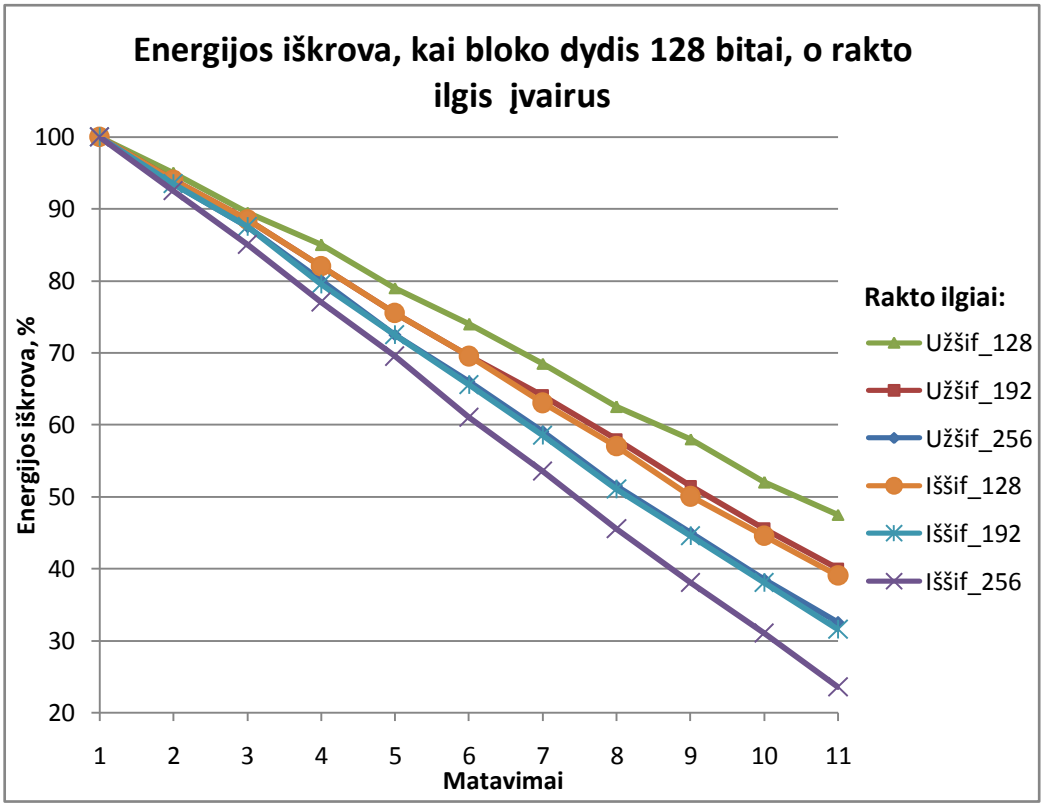
34 pav. Užšifravimo ir iššifravimo taikymuose apdorotų duomenų kiekis vienam % energijos.

Kaip matyti 35 paveiksle, energijos sąnaudų santykis tarp užšifravimo ir iššifravimo yra tiesinis. Tikimybė nuspėti, pvz., iššifravimo energijos sąnaudas konkrečiam taikymui, žinant bent kelių taikymų užšifravimo reikšmes, yra 0.99 (99%).



35 pav. Užšifravimo ir iššifravimo energijos sąnaudos tarpusavio santykis.

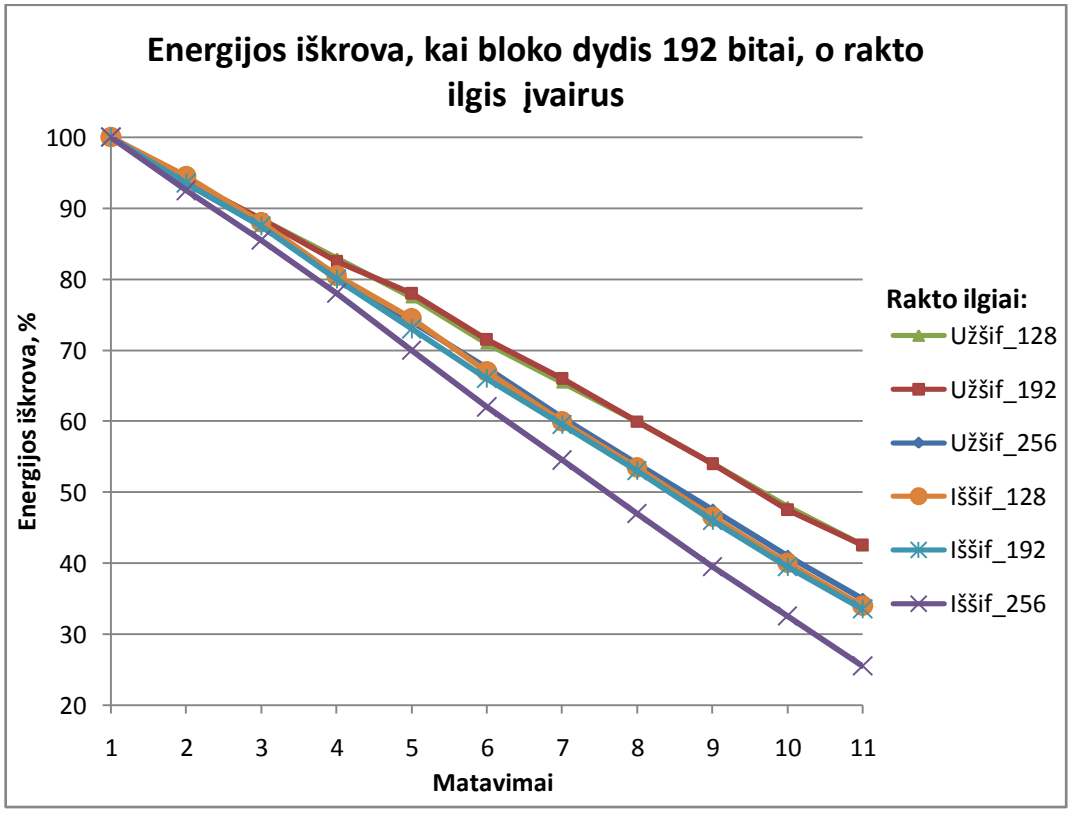
Sekančiuose 36-37 paveiksluose pateikiamos įrenginio baterijos energijos išsikrovimo vertės, matuotos kas 600 duomenų užšifravimo ir iššifravimo iteracinių veiksmų. Kiekviename grafike vaizduojamos matavimų reikšmės yra suskirstytos pagal kriptoalgoritmo bloko dydį.



36 pav. Energijos iškrova duomenis užšifruojant ir iššifruojant, kai bloko dydis 128 bitai.

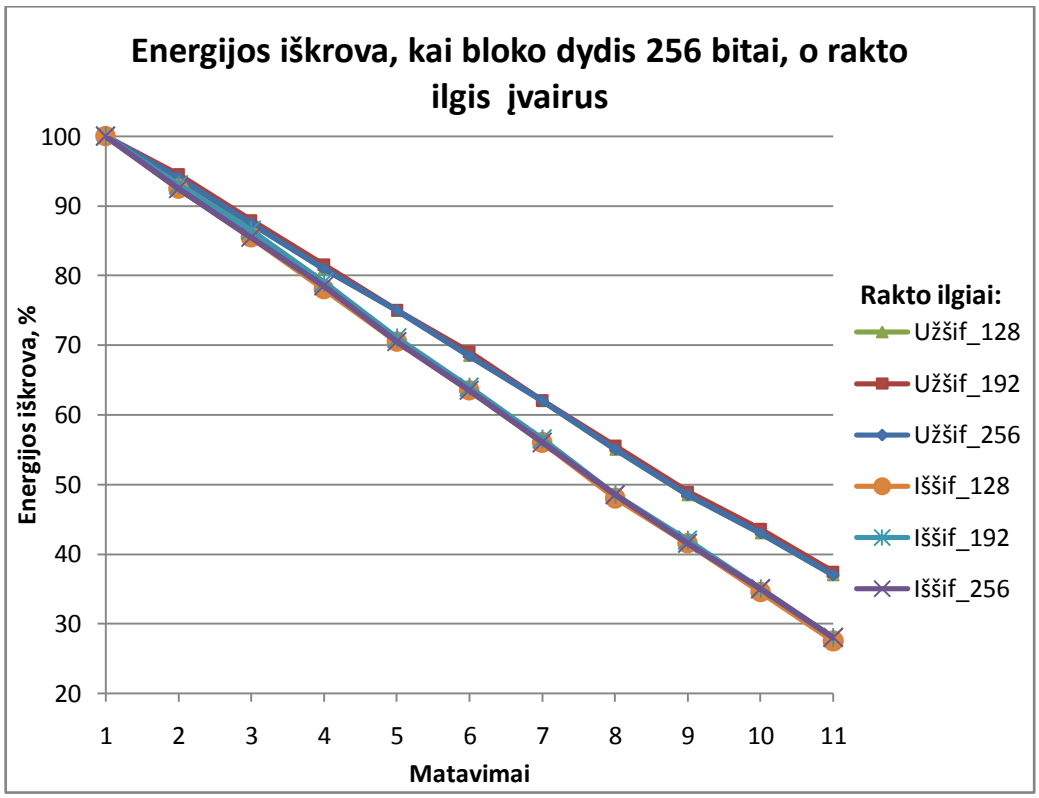
Esant kriptoalgoritmo bloko dydžiui 128 bitams (36 pav.), energijos suvartojimas tarp kai kurių kriptoalgoritmo taikymų labai supanašėja:

- Tarp užšifravimo (rakto dydis 192) ir iššifravimo (rakto ilgis 128).
- Tarp užšifravimo (rakto dydis 256) ir iššifravimo (rakto ilgis 192).



37 pav. Energijos iškrova duomenis užšifruojant ir iššifruojant, kai bloko dydis 192 bitai.

Kai kriptoalgoritmo bloko dydis 192 bitai (37 pav.), eksperimento pabaigoje iš šešių energijos suvartojimo kreivių, aiškiai išsiskyria vos trys. Vienoje kreivėje susijungia užšifravimo taikymai su 128 ir 192 bitų raktais, antroje užšifravimo taikymas su 256 bitų raktu ir iššifravimo taikymai su 128, 192 bitų raktais. Vienintelė atskirta kreivė lieka iššifravimo taikymo su 256 bitų raktu.



38 pav. Energijos iškrova duomenis užšifruojant ir iššifruojant, kai bloko dydis 256 bitai.

Galiausiai, kai kriptoalgoritmo bloko dydis yra 256 bitai, iš šešių energijos suvartojimą atvaizduojančių kreivių, išsiskyria tik dvi (38 pav.). Taip pat matyti, jog kreivės, rodančios užšifravimo (iššifravimo) veiksmų įtaką energijos sąnaudoms, susijungia į bendras kreives pagal kriptografijos veiksmų vykdymo kryptį (užšifravimas/iššifravimas).

4.3. Išvados

- Užšifravimo/iššifravimo metu baterijos energijos sąnaudos didžiausios, kai duomenų bloko dydis 128 bitai, o rakto ilgis 256 bitai, tuo tarpu, mažiausios energijos sąnaudos pasiektos, esant 128 bitų bloko dydžiui ir tokiam pačiam rakto ilgiui.
- „Pareto optimal" metodo sprendiniai t.y. šifravimo algoritmo suvartotos energijos kiekis optimalus šifravimo stiprumui, tų taikymų, kurių bloko dydis lygus rakto ilgiui.
- AES/Rijndael kriptoalgoritmo energijos sąnaudų santykis tarp užšifravimo ir iššifravimo yra tiesinis, todėl, žinant, pvz., duomenų užšifravimui suvartotą energiją, galima su 99% tikimybę nuspėti, kiek baterijos energijos prireiks duomenų iššifravimui.
- Energijos sąnaudos skiriasi priklausomai nuo Rijndael algoritmo krypties – iššifravimo metu suvartojama 10% energijos daugiau, nei vykdant užšifravimo operacijas.
- Didinant duomenų bloką nuo 128 bitų iki 256 bitų, energijos iškrova įrenginyje tiesiogiai vis mažiau priklauso nuo rakto ilgio t.y. energijos sąnaudos yra tokios pat taikymams, kurių bloko dydis 256 bitai ir rakto ilgiai 128, 192 ir 256 bitai. Šių taikymų baterijos energijos sąnaudos užšifravimui yra 66%, o iššifravimui - 76%.
- Kai rakto ilgis 256 bitai ir bloko dydis 128 bitai, energijos sąnaudos užšifravimo ir iššifravimo operacijoms įvykdyti yra 5% didesnės, negu naudojant 256 bitų raktą ir tokio paties dydžio duomenų bloką. Vadinasi, duomenų šifravimui pasirinkus 256 bitų rakto ilgį, norint sutaupyti energijos, reikėtų naudoti tokio paties dydžio duomenų bloką.
- Delninio kompiuterio naudotojas, kai duomenų saugumui užtikrinti naudojamas Rijndael kriptoalgoritmas, pagal suvartojamos energijos kiekį ir šifravimo stiprumą, gali pasirinkti vieną iš trijų profilių: žema apsauga / mažos energijos sąnaudos, vidutinė apsauga / normalios energijos sąnaudos bei aukšta apsauga / didelės energijos sąnaudos.

5. IŠVADOS

- Šiai dienai baterijos energijos nepakankama talpa mobiliuose įrenginiuose vis dar labai aktuali problema, todėl būtina įvertinti duomenų apsaugai naudojamų kriptoalgoritmų skirtingo stiprumo režimų poveikį baterijos energijos eikvojimui.
- Iš tyrimo eksperimento matavimų matyti, jog geriausi baterijos energijos sąnaudų ir Rijndael kriptoalgoritmo stiprumų santykiai („Pareto optimal“ sprendiniai) tada, kai rakto ilgis lygus duomenų bloko dydžiui.
- Turėdami tyrimo eksperimento rezultatus ir žinodami Rijndael kriptoalgoritmo informacijos saugumo lygius, mobiliųjų įrenginių naudotojams pasiūlėme tris energijos sąnaudų / duomenų apsaugos stiprumo profilius.
- Iš Rijndael algoritmo užšifravimo energijos sąnaudų galima patikimai prognozuoti iššifravimo operacijų energijos suvartojimą, pastarajam reikia ~10% daugiau energijos, nei užšifravimui.
- Priklausomai nuo veiksmų krypties (užšifravimas ar iššifravimas) ir esant įvairiems algoritmo rakto ilgiams, didžiausi energijos sąnaudų svyravimai tuose taikymuose, kurių bloko dydis 128 bitai, tačiau duomenų blokui esant 256 bitų dydžio, su skirtingais rakto ilgiais energijos suvartojimo kiekis praktiškai išlieka toks pats.
- Rijndael algoritmo didžiausio stiprumo ir mažiausio stiprumo profilių raktų ilgiai skiriasi du kartus, tačiau energijos sąnaudos, tiek užšifruojant, tiek iššifruojant skiriasi ne daugiau 11%.
- Magistrinio darbo tematika parašytas mokslinis straipsnis, kuris išspausdintas žurnale „ELEKTRONIKA IR ELEKTROTECHNIKA“ 2011 m. Nr. 2(108) [33], bei perskaitytas pranešimas tarptautinėje konferencijoje „ELEKTRONIKA 2011“, kuri įvyko 2011-05-18 Kaune.

6. LITERATŪRA

1. **Mahbub Habib S., Ries S., Mühlhäuser M.** // Clound Computing Landscape and Research Challenges regarding Trust and Reputation. Technishe Universitüt Darmstadt, Mornewegstr. 32, Germany, 2010.
2. **Pohjonen H., Ross P., G. Blickman J., Kamman R.** // Pervasive Access to Images and Data – The Use of Computing Grids and Mobile/Wireless Devices Across Healthcare Enterprises. // IEEE Transactions on information technology in biomedicine, Vol 11, No. 1, January 2007.
3. **Raghunathan A., Ravi S., Hattangady S., Quisquater J.** // Securing mobile Appliances: New Challenges for the System Designer. // Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2003.
4. **Furnell S.** // Securing mobile devices: technology and attitude. Network Research Group, University of Plymouth, Network Security p. 9-13, 2006.
5. **Haataja K., Toivanen P.** // Two Practical Man-In-The-Middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures. // IEEE Transactions on wireless communications, Vol. 9, No. 1, January 2010.
6. **R. Moyers B., P. Dunning J., K. Buennemeyer T., C. Marchany R., G. Tront J.** // Battery-Sensing Intrusion Protection System Validation Using Enhanced Wi-Fi and Bluetooth Attack Correlation. // Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institue and State University, Blacksburg, Virginia, 2009.
7. **R. Potlapally N., Ravi S., Raghunathan A., K. Jha N.** // A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols. // IEEE Transactions on mobile computing, Vol. 5, No. 2, February 2006.
8. **Jamil T.** // The Rijndael algorithm. // IEEE Potentials p. 36-38, IEEE April/May 2004.
9. **Daemen J., Rijmen V.** // The Rijndael block cipher. // Proton World Int. - Brussel, Katholieke Universiteit Leuven - Heverlee, Belgium, 2003.
10. **Federal Information Processing Standards Publications (FIPS 197)** // Advanced Encryption Standard (AES) // November 26, 2001.
11. **Razvi Doomun M., Sunjiv Soyjaudah KM, Bundhoo D.** // Energy Consumption and Computational Analysis of Rijndael-AES // IEEE 2007.
12. **F. Elashry I., S. Farag Allah O., M. Abbas A., El-Rabaie S.** // A New Diffusion Mechanism for Data Encryption in The ECB Mode // IEEE 2009.
13. **Steidel J.** // Educational software for Cryptographic Library: Benchmarking. // George Mason University, ECE 646 Cryptography & Network Security, Fall 2001.
14. **Doomun R., Doma J., Tengur S.** // AES-CBC Software Execution Optimization. // Computer Science and Engineering University of Mauritius, IEEE 2008.
15. **Banu R., Vladimirova T.** // Fault-Tolerant Encryption for Space Applications. // IEEE Transactions on Aerospace and Electronic Systems, Vol. 45, No. 1, January 2009.
16. **Shanq-Jang R., Kun-Lin T., Wen-Yew L.** // A Study on Battery Life Tradeoff between Deep Sleep and Sleep Modes on an Actual PDA. // Dept. of Electronic Engineering National Taiwan University of

Science and Technology, No. 43, Sec. 4, Keelung Rd., Taipei 106 Taiwan & Dept. of CSIE National Taipei University of Technology, No. 1, Sec. 3, Chung-hsiao E. Rd., Taipei, Taiwan, IEEE 2009.

17. **K. Buennemeyer T., M. Nelson T., M. Clagett L., P. Dunning J., C. Marchany R., G. Tront J.** // Mobile Device Profiling and Intrusion Detection using Smart Batteries. // Proceedings of the 41st International Conference on System Sciences, Hawaii, 2008.
18. **Fei Y., Zhong L., Jha N. K. //** An Energy-Aware Framework for Dynamic Software Management in Mobile Computing Systems. // Plaza, New York, NY, USA, 2008.
19. **Min. J., Cha H., P. Sirini V. //** An Efficient Power Management Mechanism for WiFi-based Handheld System. // IEEE 2006.
20. **Toldinas J., Štuikys V., Damaševičius R., Ziberkas G.** // Application-Level Energy Consumption In Communication Models For Handhelds. // Electronics and Electrical Engineering journal No. 6(94), Computer department and Software Engineering department, Kaunas University of Technology, Kaunas, 2009.
21. C++ ir C# programavimo kalbų savybių palyginimas. [Žiūrėta: 2010-04-20]. Prieiga per internetą: < http://www.davidlenihan.com/2009/05/c_vs_c.html>
22. .Net Compact Framework apibūdinimas, savybės ir pan. [Žiūrėta: 2011-04-20]. Priega per internetą: < http://msdn.microsoft.com/library/w6ah6cw1.aspx>
23. Eksperimento failas - „benchmark“ paveikslas. [Žiūrėta: 2011-01-15]. Priega per internetą: < http://sipi.usc.edu/database/database.php?volume=misc&image=12#top >
24. Mobiliųjų įrenginių paklausos augimas. [Žiūrėta: 2010-01-28]. Prieiga per internetą: <http://www.xbitlabs.com/news/networking/display/20091223234148_Wi_Fi_Hotspot_Usage_by_Handhelds_Increasing_Research_Firm.html >
25. Mobiliųjų įrenginių baterijų energijos talpos kytimas vs mikroprocesorių spartos augimas. [Žiūrėta: 2010-01-25]. Prieiga per internetą: <http://www.mobilehandsetdesignline.com/howto/207100001 >
26. Kenkėjiškos programos, jų tipai. [Žiūrėta: 2010-01-26]. Prieiga per internetą: < http://www.viruslist.com/en/analysis?pubid=204792080 >
27. Mobiliųjų įrenginių operacinių sistemų rinkos pasiskirstymas. [Žiūrėta: 2011-04-09]. Prieiga per internetą: < http://www.gartner.com/it/page.jsp?id=1543014 >
28. **T. R. Hager C., F. Midkiff S., Park Jung-Min, L. Martin T.** // Performance and Energy Efficiency of Block Ciphers in Personal Digital Assistants. // Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061 USA, 2005.
29. .NET Compact Framework struktūra, moduliai, savybės. [Žiūrėta: 2011-05-09]. Prieiga per internetą: < http://msdn.microsoft.com/en-us/library/f44bbwa1%28v=VS.90%29.aspx >
30. C# programavimo kalbos savybės. [Žiūrėta: 2011-05-09]. Prieiga per internetą: < http://www.csharpfriends.com/Articles/getArticle.aspx?articleID=37 >
31. 2009-2010 metų mobiliųjų įrenginių rinkos statistika. [Žiūrėta: 2011-05-09]. Prieiga per internetą:

<http://mobithinking.com/stats-corner/global-mobile-statistics-2011-all-quality-mobile-marketing-research-mobile-web-stats-su >

32. Mobiliųjų įrenginių atminties struktūra, savybės, talpos paskirstymas. [Žiūrėta: 2011-05-09]. Prieiga per internetą: < http://www.pdagold.com/articles/detail.asp?a=243 >
33. **Toldinas J., Štuikys V., Damaševičius R., Ziberkas G., Banionis M.** // Energy Efficiency vs Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices. // Electronics and Electrical Engineering journal No. 2(108), Computer department and Software Engineering department, Kaunas University of Technology, Kaunas, 2011.

7. PAVEIKSLĖLIŲ SĄRAŠAS:

1. **Fei Y., Zhong L., Jha N. K.. //** An Energy-Aware Framework for Dynamic Software Management in Mobile Computing Systems. // Plaza, New York, NY, USA, 2008.
2. **T. R. Hager C., F. Midkiff S., Park Jung-Min, L. Martin T.** // Performance and Energy Efficiency of Block Ciphers in Personal Digital Assistants. // Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061 USA, 2005.
3. **Chi-Wu Huang, Che-Hao Chiang, Chien-Lun Yen, Yi-Cheng Chen, Kuo-Huang Chang, Chi-Jeng Chang** // The AES Application in Image Using Different Operation Modes // National Taiwan Normal University, Taipei, Taiwan, 2010.
4. **Daemen J., Rijmen V.** // The Rijndael block cipher. // Proton World Int. - Brussel, Katholieke Universiteit Leuven - Heverlee, Belgium, 2003.
5. **Toldinas J., Štuikys V., Damaševičius R., Ziberkas G., Banionis M. //** Energy Efficiency vs Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices // Computer Department, Software Engineering Department, Kaunas University of Technology, 2011.

8. SUMMARY

Nowadays technologies are being improved rapidly – extremely fast dedicated servers, high internet and network throughput, enables more and more applications to be moved to a virtual space, which is called 'Cloud computing'. Cloud computing is responsible that all shared computer resources, like Software, Hardware and Network would be allocated for services in optimal way. This feature enables to appear new services, like media streaming, documents editing online etc. This kind of services are based on 'Thin client' architecture, which is especially important for mobile devices. Main feature of this architecture is that applications which require many computations, now can be executed in dedicated servers and results can be displayed in mobile device screen. Fact, that aplication is not executed in mobile device environment decreases probability that important information will be infected by viruses, worms etc.

However, between many existing problems with mobile devices there are two major which should be stated in the first place: energy consumption and information security. The first issue is due to inadequate progress of computational resources and battery energy power (e.g., CPU speed, memory capacity is being increased several times within a few years, while battery power doubles only in a decade). The second issue is closely related with the first one. Today people want to work anywhere, so there is abillity that mobile devices including important information could be stolen or lost. That's why in order to protect data from publishing it in readable form we must encrypt it.

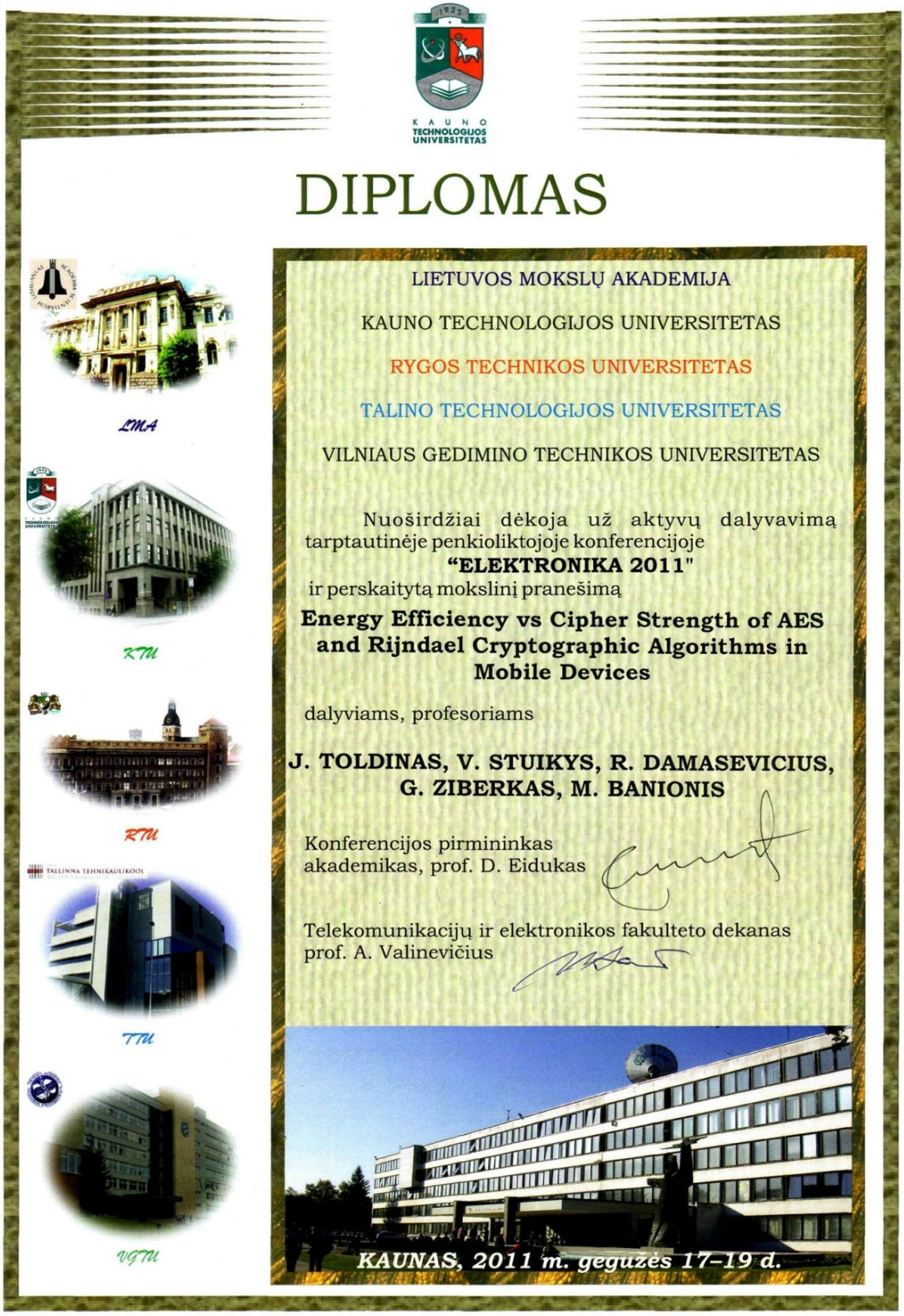
To better understand the relation between strength of the crypto algorithm and battery energy consumption we have chosen to investigate Rijndael symmetric encryption algorithm. This algorithm because of different key lengths (longer key means better strength) and block sizes is universal and used widely including mobile devices.

Solutions of .NET Compact Framework platform have motivated us to programme a tool which has calculated the energy consumption of encryption and decryption processes which were based on Rijndael algorithm.

The results of research have shown that „Pareto optimal" values, which show the best ratio between strength of the Rijndael crypto algorithm and battery energy consumption, are achieved when key lengths and block sizes are equal. Moreover, based on the experiment results, we can construct three security profiles for mobile device users as follows: 1) low energy / low security – so far considered secure, but theoretically crackable; 2) medium energy / medium security – suitable for very secret information, consumes ~10% more energy than low energy/security profile; 3) high energy / high security – suitable for top secret information, consumes ~12% more energy than medium energy/security profile.

9. PRIEDAI

9.1. Tarptautinės konferencijos „Elektronika 2011“ dalyvio diplomas



9.2. Publikacija „Energy Efficiency vs Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices“

Šiame priedo skyriuje pateiktas mokslinis straipsnis, išspausdintas žurnale ELEKTRONIKA IR ELEKTROTECHNIKA, 2011 m. Nr. 2 (108).

ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2011. No. 2(108)*
ELEKTRONIKA IR ELEKTROTECHNIKA

SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
T 120
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Energy Efficiency vs Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices

J. Toldinas
Computer Department, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, e-mail: eugenijus.toldinas@ktu.lt
V. Štuikys, R. Damaševičius, G. Ziberkas
Software Engineering Department, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, phone: +370 37 300399, e-mail: vytautas.stuikys@ktu.lt, ziber@soften.ktu.lt
M. Banionis
Computer Department, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, e-mail: mindaugas.banionis@stud.ktu.lt

Introduction

Currently, mobility is highly important. Everyone has mobile devices: not only laptops, but also smart phones, pocket PCs, GPS receivers, etc. Mobile/wireless devices are increasingly used not only for communication, but also for other critical applications such as data storage. However, due to small size they can be easily lost or stolen. Cryptography is used for securing information stored in mobile devices. Usage time of a mobile device is constrained by its most critical resource – battery. The user must be aware of the energy consumption characteristics of the applications and services he uses on his mobile device [1]. Encryption algorithms, which play a main role in information security systems, consume a significant amount of computing resources and battery energy, which are very limited. High energy consumption has a direct impact on the battery life, and, consequently, on the duration and extent of the user's mobility. Energy consumption reduction of portable system is of primary importance [2].

The design of crypto algorithms typically does not account for physical constraints such as limited battery energy. Therefore, the primary challenge in providing security in mobile devices is minimizing energy consumption and maximizing security [3]. Scalable features such as scalable key establishment protocols and scalable authentication schemes, in which different security, performance and energy trade-offs are enabled for different application scenarios are especially desirable [4].

Here we evaluate cipher strength (the number of bits in the key used to encrypt data) of AES and Rijndael crypto algorithms versus energy consumption in mobile devices aiming to find an energy efficient combination of crypto algorithm parameters for different user application scenarios and energy consumption strategies.

Context of Research

AES (Advanced Encryption Standard) is an encryption standard adopted by the U.S. government starting in 2001. It is widely used to protect network traffic, personal data, and corporate IT infrastructure. AES is a symmetric block cipher that encrypts/decrypts data in several rounds by taking a fixed block of 128 bits of data and producing the encrypted data. Each round for encryption uses a sub-key that is generated using a key schedule and performs a sequence of steps on the input state, which is then fed into the next round.

In 2003, the Government of USA announced that AES may be used to protect classified information: the cipher strength of all key lengths of AES are sufficient to protect classified information up to the SECRET level, however, TOP SECRET information requires use of either 192 or 256 bit keys [5]. However, the recent paper [6] claims that the 10-round AES is theoretically crackable by cryptanalysis [7].

The Rijndael algorithm is a symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in variable-length blocks. The block length and the key length can be set independently (AES cannot do this) to 128, 192 or 256 bits. Rijndael uses a variable number of rounds, depending on the key/block sizes, as follows:

- 9 rounds if both the key and block size is 128 bits;
- 11 rounds if either the key or block size is 192 bits;
- 13 rounds if either the key or block size is 256 bits.

Rijndael is expected to replace Data Encryption Standard (DES) and its later version Triple DES over the next few years in many cryptography applications.

Microsoft® provides a .NET framework technology that has a crypto service provider for information encryption/ decryption on a handheld PC with DES, 3DES, AES, RC2 algorithms [8].

As energy consumption awareness is highly important in mobile devices, we must ensure required functionality, data security level and reasonable use of energy at the same time. Empirically we can predict that cryptography with a longer key will produce higher levels of security at the cost of higher energy consumption. However, block size also has influence, because larger blocks will require more encryption rounds. Energy consumption also depends on the application scenario, e.g., if the user only encrypts data on a mobile device but decrypts its elsewhere, the energy/cipher strength trade-off will differ from the scenario, when a user encrypts/decrypts its data on a mobile device only. Therefore, in order to use crypto algorithms energy-efficiently one needs to understand the relationships between energy consumption and encryption parameters. Once these relationships are understood well then it is possible to optimize energy consumption vs. security requirement or vice-versa.

Methodology

The task of the experiments is to identify dependencies between cryptography key lengths and block sizes on one hand, and energy consumption strategy, energy / cipher strength trade-offs and cryptography application scenarios on the other hand. These dependencies are expressed using a feature diagram in Fig.1.

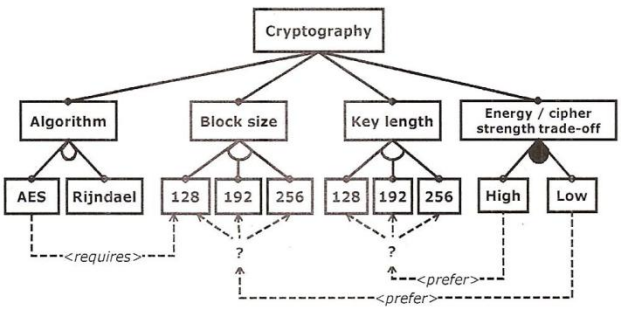


Fig. 1. Feature model of dependencies in cryptography domain

Fig. 2 outlines an algorithm that enables to perform measurements of energy consumption and obtain the desired relationships. We apply the OS-based measuring scheme [9], where the amount of the consumed energy over time is periodically written to the file during the data cryptography process. The remaining part of analytic framework and result interpretation was described in [10].

The energy consumption values for individual crypto algorithms are obtained by running their .NET Compact Framework Crypto Service Provider implementations, and measuring the current battery drain. For getting valuable results of battery drain when data is encrypted/decrypted, we iterate the cryptography process. Since encryption and decryption time may vary we perform encryption and decryption separately.

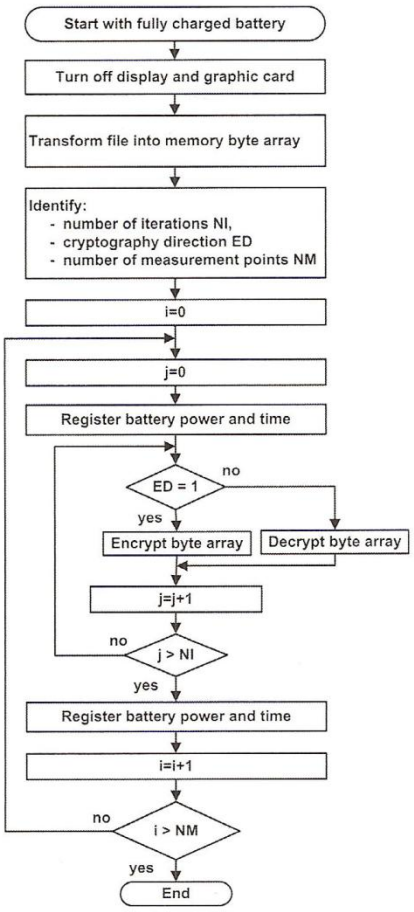


Fig. 2. Energy measurement algorithm for a crypto algorithm

Experiments

To realize the experiments we have developed the program that implements the algorithm (Fig. 2) in C# language for .NET Compact Framework. The experiments were performed on the PDA of the model ASUS P750 (Pocket PC platform, Intel PXA270 520 MHz CPU, 256 MB RAM, Windows Mobile © 6 Professional CE OS 5.2). We used .NET Compact Framework 3.5 version.

Motivated by the fact that the largest amount of information the users work on locally as well as on the internate is made by the media files (pictures, music, video), for encoding we used a benchmark 'Lena.bmp' (size = 786,486 B) image (see Fig. 3).

Fig. 3. Benchmark image used for encoding

The initial condition for all experiments is the same: battery is fully charged at 100% level. The image file is loaded from a storage memory to an array and an encryption algorithm is applied. To achieve a significant battery drain for more precise measurement, the encryption process is repeated 6000 times. After each experiment, the battery is agained charged to 100%. The same procedure is also applied for measuring energy consumption of a decryption algorithm.

We provide the summary of the experiment results in Tables 1 (for AES with fixed block size of 128 bytes) and Tables 2 (for Rijndael with variable block sizes of 128 bytes, 192 bytes and 256 bytes).

Table 1. Experimental results for AES/Rijdnael encryption

Block size, b	Key size, b	Elapsed time (all iterations), hh:mm	Battery energy consumed, %
128	**128**	**02:01:04**	**50**
128	192	02:19:10	57
128	256	02:38:13	64
192	128	02:27:01	55
192	**192**	**02:27:01**	**55**
192	256	02:12:51	62
256	128	02:12:15	60
256	192	02:27:09	59
256	**256**	**02:27:01**	**60**

Table 2. Experimental results for AES/Rijdnael decryption

Block size, b	Key size, b	Elapsed time (all iterations), hh:mm	Battery energy consumed, %
128	**128**	**02:29:59**	**57**
128	192	02:40:07	65
128	256	02:57:59	72
192	128	02:35:12	62
192	**192**	**02:35:03**	**63**
192	256	02:53:58	70
256	128	02:48:54	69
256	192	02:48:54	68
256	**256**	**02:48:09**	**68**

Analysis of experimental results

We treat the problem of finding best energy efficiency vs cipher strength as the *Pareto optimality* problem. Let E be a set of feasible design choices, where $e_{ij} = f(b_i, k_j)$, $e_{ij} \in E$ is a choice dependant upon two criteria: b_i (block size) and k_j (key size). Let Y be a subset of E, where $y_j = \min_{b_i} f(b_i, k_j)$, $y_j \in Y$. Then Y is a set of the Pareto-optimal solutions of E. The experimental results, which belong to a set of the Pareto-optimal solutions, are shown in Tables 1 and 2 in grey. We can note that for all Pareto optimal solutions the block size and the key size are equal.

Based on these results, we can construct three security profiles for mobile device users as follows:

1) *Low energy / low security:* so far considered secure, but theoretically crackable.

2) *Medium energy / medium security:* suitable for top secret information; consumes ~ 10% more energy than low energy/security profile.

3) *High energy / high security:* suitable for top secret information; consumes ~ 8% more energy than medium energy/security profile.

These profiles are summarized in Fig. 4.

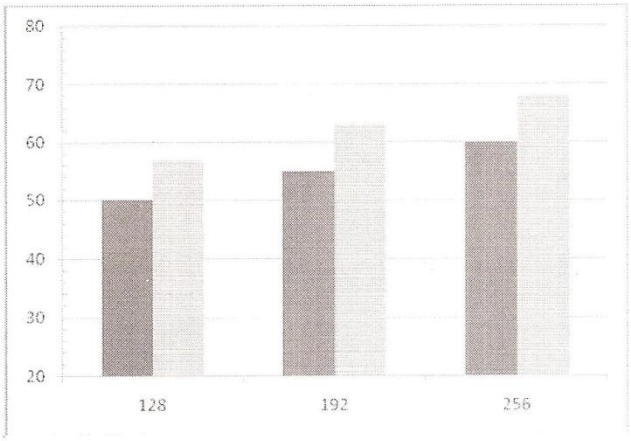


Fig. 4. Comparison of security profiles for encryption (left) and decryption (right): energy consumption (%) vs block/key size (b)

Another finding from Tables 1 and 2 is that decryption requires more battery energy than encryption. Furthermore, there is a linear relationship between enryption energy and decryption energy: decryption requires ~14% ($R^2 = 0.985$) more energy than encryption (see Fig. 5).

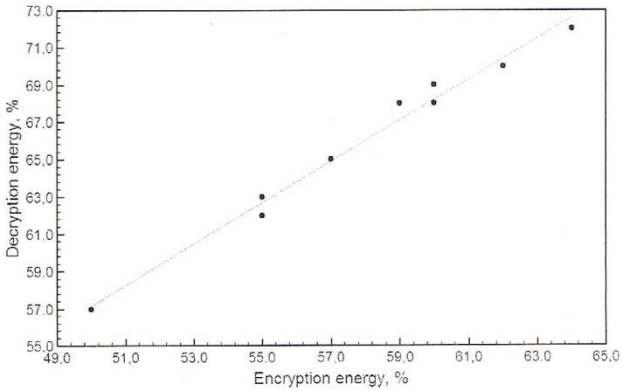


Fig. 5. Relationship between decryption vs. encryption energy

Evaluation and conclusions

For experiments we use the Microsoft .NET Compact Framework as a modern and popular platform for safe development mobile applications and secure information management. Though there are many encryption/ decryption algorithms we were restricted with the

algorithms provided by this Framework. The energy-efficiency of crypto algorithms with varying key and block sizes is highly different. Therefore, the user of a mobile system user should choose the most appropriate parameters of a crypto algorithm by taking into account the level of security required and the operational cost that the user is willing to accept depending on the security level they choose, and energy needed to perform encryption/ decryption operation with respect to the battery lifetime.

The main results of this paper are as follows:

1) The Pareto-optimal values for energy consumption of AES/Rijndael crypto algorithm are achieved when block sizes and key sizes are equal.

2) We proposed three energy/security profiles for users of mobile devices based on using 128, 192 and 256 b blocks/keys.

3) The results of energy consumption measurements when performing data encrypion can be used to reliably predict energy consumption of decryption operation: decryption requires 14% more energy than encryption.

References

1. **Tiliute D.E.** Battery management in wireless sensor networks // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 4(76). – P. 9–12.
2. **Baums A.** Energy consumption optimization in hard real-time system CMOS processors // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 4(68). – P. 19–22.
3. **Chandramouli R.** Battery power-aware encryption. // ACM Transactions on Information and System Security (TISSEC), 9(2). 2006. – P. 162-180.
4. **Dumčius A., Gužauskas N.** Mixed Data Encryption System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2002. – No. 6(41). – P. 12–15.
5. **Kelsey J., Lucks S., Schneier B., Stay M., Wagner D., Whiting D.** Improved Cryptanalysis of Rijndael // Fast Software Encryption. 2000. – P. 213–230.
6. **Biryukov A., Keller N., Khovratovich D., Shamir A.** Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds // Advances in Cryptology - Eurocrypt 2010, 29th Int. Conf. on the Theory and Applications of Cryptographic Techniques, Lecture Notes in Computer Science vol. 6110. – Springer, 2010. – P. 299–319.
7. **Toemeh R., Arumugam S.** Breaking Transposition Cipher with Genetic Algorithm // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 7(79). – P. 75–78.
8. **Toldinas J., Rudzika D., Štuikys V., Ziberkas, G.** Rootkit Detection Experiment within a Virtual Environment // Electronics and Electrical Engineering – Kaunas: Technologija, 2009. – No. 8(104). – P. 63–68.
9. **Toldinas E., Štuikys V., Damaševičius R., Ziberkas G.** Application–level energy consumption in communication models for handhelds // Electronics and Electrical Engineering – Kaunas: Technologija, 2009. – No. 6(94). – P. 73–76.
10. **Damaševičius R., Štuikys V., Toldinas E.** Embedded program specialization for multiple criteria trade–offs // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 8(88). – P. 9–14.

Received 2010 11 16

J. Toldinas, V. Štuikys, R. Damaševičius, G. Ziberkas, M. Banionis. Energy efficiency vs cipher strength of AES and Rijndael cryptographic algorithms in mobile devices. – Kaunas: Technologija, 2011. – No. 2(108). – P. xx–xx.

We analyse energy efficiency vs. cipher strenth of AES/Rijndael crypto algorithms in a mobile device with respect to block and key size. The experimental results show that Pareto-optimal solutions have equal block and key sizes. We also propose three energy/security profiles for the users of mobile devices. As decryption operation requires 14% more energy than encryption, the results of energy consumption measurements when performing data encrypion can be used to predict energy consumption of decryption operation. Ill. 5, bibl. 10, tabl. 2 (in English; abstracts in English, Russian and Lithuanian).

Е. Толдинас, В. Штуйкис, Р. Дамашявичюс, Г. Зиберкас, М. Банионис. Сравнительный анализ криптографических алгоритмов AES и Rijndael с точки зрения энергопотребления и безопасности в мобильных устройствах // Электроника и электротехника. – Каунас: Технология, 2011. – № 2(108). – С. xx–xx.

Анализируются криптографические алгоритмы AES и Rijndael, применяя их в мобильных устройствах с различными размерами блока шифруемых данных и различными размерами ключа. Полученные результаты эксперимента имеют Pareto-оптимальное решение при равенстве размера блока шифруемых данных и размера ключа. Предложены три профиля для пользователей мобильных устройств соотносящие энергию и безопасность с точки зрения энергосбережения. Предложено прогнозирование энергопотребления для дешифровки данных, на основе того, что дешифрование потребляет больше энергии, чем шифрование на 14%. Ил. 5, библ. 10, табл. 2 (на английском языке; рефераты на английском, русском и литовском яз.).

J. Toldinas, V. Štuikys, R. Damaševičius, G. Ziberkas, M. Banionis. AES ir Rijndael kriptoalgoritmų energijos suvartojimo ir saugumo palyginimas mobiliuose įrenginiuose // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 2(108). – P. xx–xx.

Nagrinėjamas AES ir Rijndael algoritmų energijos suvartojimas esant įvairiems šių algoritmų parametrams: bloko ir rakto dydžiams. Remiantis gautais eksperimentiniais rezultatais: 1) siūloma naudoti lygius bloko ir rakto dydžius, 2) pasiūlyti trys energijos/saugumo vartotojo profiliai, 3) pastebėta tiesinė priklausomybė tarp šifravimo ir dešifravimo metu sunaudojamos energijos kiekio, kuriuo remiantis galima prognozuoti dešifravimo metu sunaudojamos energijos kiekį. Il. 5, bibl. 10, lent. 2 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).