KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Darius Naunikas

Energijos suvartojimo naudojant kriptografinius servisus delniniuose kompiuteriuose tyrimas

Magistro darbas

Darbo vadovas
doc. dr. J. Toldinas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Darius Naunikas

Energijos suvartojimo naudojant kriptografinius servisus delniniuose kompiuteriuose tyrimas

Magistro darbas

Recenzentas
doc. dr. R. Damaševičius
2010-05-

Vadovas
doc.dr. J. Toldinas
2010-05-

Atliko
IFN-8/3 gr. stud.
Darius Naunikas
2010-05-15

Kaunas, 2010

Turinys

1. ĮVADAS

Šiomis dienomis, sparčiai augant informacinėms technologijoms bei atsirandant vis didesniems verslo bei mokslo poreikiams, siekiant, kad darbo vieta būtų ten kur yra darbuotojas ar mokslininkas, vis didesnė pasaulio virtualizacija yra tai, kas išreiškiama posakiu „debesų kompiuterija“ („Cloud Computing“). „Debesų kompiuterija“ - tai sekanti kompiuterių pasaulio evoliucijos dalis, apimanti tris pagrindinius komponentus: programinę įrangą (software), kompiuterinę įrangą (hardware) bei tinklus (network). Šių trijų pagrindinių bei kitų komponentų „sąjunga“ sudaro taip vadinamą „Debesį“ („Cloud“) [1]. Ši technologija turi neginčijamą naudą tiek mokslui tiek verslui, tiek ir kitoms gyvenimo sritims. „Debesys“ gali būti naudojami moksliniams skaičiavimams, tyrimų rezultatų apdorojimui, tokias sistemas tampa lengviau prižiūrėti, jos patikimesnės, lengviau prieinamos. Mobilumas ir mobilios sistemos yra neatsiejama šios sistemos dalis [2]. Mažėjant mobiliųjų įrenginių dydžiui bei gerėjant jų techninėms charakteristikoms, atsiranda vis didesnis poreikis naudotis išmaniaisiais mobiliais įrenginiais: nešiojamais kompiuteriais, delniniais kompiuteriais, išmaniais telefonais, kurių pagalba kiekvieną dieną galima atlikti vis daugiau ir daugiau funkcijų, operacijų, kad ir kur vartotojas bebūtų („on the go“): žiniatinklių peržiūra, dokumentų redagavimas, internetinė bankininkystė, grafinės pramogos ir t.t. [3]. Nors yra ir kitų probleminių šios technologijos sričių, kaip pagrindines norėčiau paminėti dvi: mobiliųjų įrenginių akumuliatorių energijos sąnaudos bei informacijos saugumas. Nors mobiliųjų įrenginių techninės charakteristikos (procesorius, atmintis) per paskutinius metus padidėjo tūkstančius kartų [4], tačiau mobiliųjų įrenginių akumuliatoriaus energijos ištekliai vis dar išlieka problematiški (trumpas veikimo laikas). Vienas iš svarbiausių kriterijų siekiant efektyviai naudoti mobiliuosius įrenginius - tai saugumas. O neatsiejama saugumo technologijų dalis yra padidėjusios energijos sąnaudos joms užtikrinti. Kadangi pagrindinė darbo „on the go“ koncepcijos mintis yra mobilumas, tai išlaikyti kuo ilgiau veikiantį mobilųjį įrenginį yra nemažas iššūkis, reikalaujantis racionalaus energijos vartojimo. Tinkamai pasirinkti, sukurti saugumo sprendimus galima tik gerai juos suprantant (žinant jų charakteristikas), žinant jų įtaką energijos sąnaudoms.

Magistrinio darbo tyrimo sritis – mobilios kompiuterinės sistemos. Tyrimo objektas – delniniai kompiuteriai su Windows mobile operacine sistema ir .NET compact framework platforma. Naudojant delninius įrenginius viena iš svarbiausių problemų yra akumuliatoriaus „gyvavimo“ laikas. Norint saugiai naudoti mobilius delninius kompiuterius – užtikrinti slaptos informacijos saugumą, reikia pasitelkti kriptografinius metodus, kurie yra imlūs energijos suvartojimui.

Šio magistrinio darbo tikslas – ištirti energijos suvartojimą delniniuose kompiuteriuose nustatant .NET compact framework platformos kriptografinėje bibliotekoje esamų metodų (DES, 3DES, AES ir RC2) [5] įtaką energijos suvartojimui.

Darbo uždaviniai:

- ✓ Atlikti kitų mokslininkų tyrimų rezultatus ir pateiktą medžiagą apie tyrimus magistrinio darbo tema.
- ✓ Suprojektuoti programinę įrangą tyrimui atlikti, naudojančią .NET compact framework platformos kriptografinius metodus.
- ✓ Sukurti tyrimo metodiką, surasti/suformuoti reikiamus duomenis tyrimui, empiriniu-praktiniu būdu nustatyti reikalingus tyrimui parametrus.
- ✓ Atlikti programinės įrangos realizaciją bei ją ištestuoti;
- ✓ Panaudojant tyrimui sukurtą programinę įrangą, atlikti tyrimą bei užfiksuoti tyrimo metu gautus rezultatus.
- ✓ Atlikti tyrimo rezultatų analizę ir pateikti rezultatus grafiniu būdu.
- ✓ Suformuoti ir pateikti išvadas bei pasiūlymus.

Naudojant delninius įrenginius viena iš svarbiausių problemų yra akumuliatoriaus „gyvavimo“ laikas. Norint saugiai naudoti delninius kompiuterius, reikia pasitelkti kriptografinius metodus, kurie yra labai imlūs energijos suvartojimui. Darbo tikslas – ištirti energijos suvartojimą delniniuose kompiuteriuose bei nustatyti .NET compact framework platformos kriptografinėje bibliotekoje esamų metodų įtaką energijos suvartojimui delniniuose kompiuteriuose.

Darbo struktūra:

- ✓ Darbo analizės dalyje pateikta informacijos saugumo klasifikacija, pagrindinės/svarbiausios mobiliųjų delninių kompiuterių saugumo problemos bei saugumo lygiai. Aprašomas mobiliųjų delninių kompiuterių saugus turinys. Detaliai aprašomi kriptografiniai algoritmai: DES, 3DES, AES, RC2, RSA, DSA, MD5, SHA, PRNG. Apžvelgiami kitų mokslininkų atlikti tyrimai bei medžiaga apie tyrimus bei jų rezultatai, susiję su akumuliatorių energijos sąnaudomis, taikant kriptografinius algoritmus. Šios dalies pabaigoje pateiktas analizės rezultatų apibendrinimas ir išvados.
- ✓ Projektavimo dalyje aprašyti tyrimui naudojamos programinės įrangos struktūra, tyrimui naudojamų .NET compact framework platformos kriptografinių algoritmų slaptųjų raktų dydžiai, tyrimo duomenų struktūra, apibendrinta programinės įrangos blokinė bei panaudojimo atvejų schemos. Taip pat šioje dalyje pateikiami kintamųjų

bei klasių diagramos, registruojamų tyrimo parametrų aprašymas, bei kiti programinės įrangos tyrimui reikalingi nustatymai.

- ✓ Eksperimentinėje dalyje aprašytas energijos suvartojimo, panaudojant .NET compact framework platformos kriptografinius algoritmus, tyrimas, jo metodika, pateikti skaitiniai ir grafiniai tyrimo rezultatai.
- ✓ Pabaigoje pateikiamos atlikto darbo pagrindinės išvados ir rezultatai.
- ✓ Prieduose pateikiami papildomi paveikslai su grafiniais tyrimo rezultatais ir kita su darbu susijusi naudinga medžiaga.

Magistrinio darbo tematika parašytas mokslinis straipsnis išspausdintas žurnale „ELEKTRONIKA IR ELEKTROTECHNIKA“ 2010 m. Nr. 5 (101) [26] ir perskaitytas pranešimas 14-oje tarptautinėje konferencijoje „ELEKTRONIKA 2010“ vykusioje 2010-05-18 Vilniuje.

2. DELNINIŲ MOBILIŲJŲ ĮRENGINIŲ SAUGA IR ENERGIJOS SĄNAUDOS

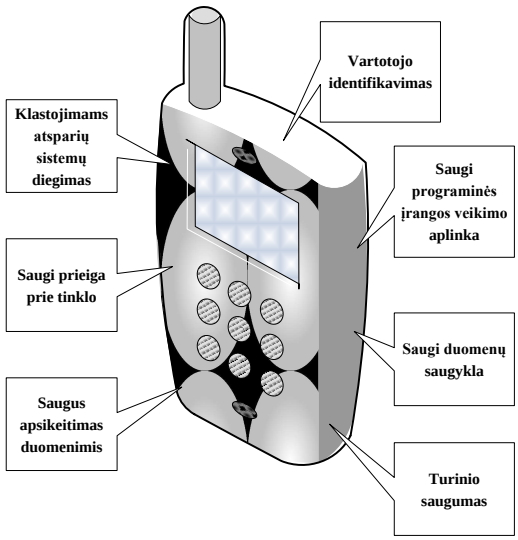
Šiandien vis daugiau akumuliatorių energija „varomų“, mobiliųjų sistemų – PDA, mobiliųjų telefonų, tinklinių jutiklių ir intelektualių kortelių ir kt. – naudojamos saugoti, prieiti naudoti, manipuliuoti „jautriais“ duomenimis, dėl to saugumas tampa pačiu svarbiausiu dalyku. Plačiai pripažįstama, kad netolimoje ateityje mobilieji įrenginiai taps „patikimais asmeniniais prietaisais“, leidžiančiais nustatyti naudotojo tapatybę, tapti pirkimo priemone ar būti naudingi įvairiose gyvenimo srityse. Saugumas tokiose sistemose apima vartotojo identifikavimą, informacijos saugojimo užtikrinimą, saugų programų vykdymą ir saugų bendravimą/komunikavimą.

Bell-LaPadula (BLP) modelis buvo sukurtas remiantis daugialypės saugos modeliu siekiant apsaugoti įslaptintą informaciją. Šis modelis naudojamas Jungtinių Amerikos Valstijų Gynybos ministerijoje (*DoD – Department of Defence*), kurioje informacija pagal slaptumą klasifikuojama į keturis lygius [6]:

- ✓ YPATINGAI SLAPTA - labai slaptą (LS), (TS);
- ✓ SLAPTA - slaptą (S), (S);
- ✓ KONFIDENCIALU - konfidencialų (K), (C);
- ✓ NEKLASIFIKUOTA - neįslaptintą (N), (UC).

2.1. Delninių mobiliųjų įrenginių sauga

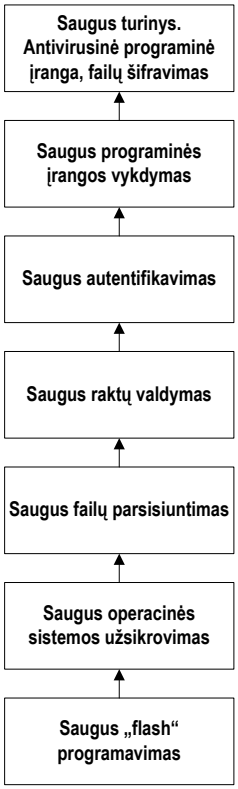
Saugumo mechanizmų vaidmuo yra užtikrinti duomenų konfidencialumą ir vientisumą bei dalyvaujančių šalių autentiškumą. Be paminėtų, taip pat labai svarbų funkcionalumą, kuris užtikrina nepaneigiamumą ir apsimetimo uždraudimą, apsaugą nuo kopijavimo, prevenciją nuo DoS išpuolių, filtravimą nuo virusų ir „kenkėjiškų“ programų.



1 pav. Svarbiausios saugumo problemos

Paveikslas Nr. 1 iliustruoja kai kurias iš svarbiausių saugumo problemų mobilaus įrenginio saugumo požiūriu [7]:

- ✓ Vartotojo identifikavimas. Siekia užtikrinti, kad tik autorizuoti subjektai gali naudoti prietaisą/programinę įrangą.
- ✓ Saugi duomenų saugykla. Apima saugumo jautrios informacijos, tokios kaip slaptažodžiai, PIN kodai, raktai, sertifikatai ir t.t., kuri gali būti saugojama vidinėje mobilaus įrenginio atmintyje (flash) .
- ✓ Saugi programinės įrangos veikimo aplinka. Siekiant užtikrinti saugią programų vykdymo aplinką yra būtina užtikrinti, kad būtų negalimi išpuoliai nuo „kenkėjiškų“ programų, tokių kaip virusai ar Trojos arkliai.
- ✓ Klastojimams atsparių sistemų diegimas. Reikalingas, kad užtikrinti kompiuterinės įrangos saugumą nuo įvairių fizinių ir elektrinių atakų.
- ✓ Saugi prieiga prie tinklo. Užtikrina, kad tik autorizuoti įrenginiai gali būti įjungti į tinklą ar naudotis paslaugą.
- ✓ Saugus apsikeitimas duomenimis. Apžvelgia saugų duomenų perdavimo privatumą ir vientisumą iš/į mobilųjį įrenginį.
- ✓ Turinio saugumas. Bet koks turinys, kuris parsiųstas į arba laikomas mobiliame įrenginyje, naudojamas pagal sąlygas, pateiktas turinio teikėjo (pvz., teisė tik skaityti, ne kopijuoti ir pan.)



2 pav. Hierarchiškai išdėstyti saugumo lygiai

Saugumo iššūkiai paprastai yra sudėtinga užduotis, netgi žiūrint vieno mobilaus įrenginio perspektyvoje. Norint aiškiau suprasti saugumo problematiką, reikia hierarchiškai išdėstyti saugumo lygius (pav. 2) [19].

2.2. Saugus delninio mobilaus įrenginio turinys

Delniniuose kompiuteriuose su operacine sistema Windows Mobile yra įdiegta standartinė failų užšifravimo/iššifravimo sistema „Encryption“ [25], kuri vykdo veiksmus išorinėje atmintyje (storage card). Pagal nutylėjimą failai užšifruojami/iššifruojami panaudojant AES algoritmą su 128 bitų raktu. Pakeitus nustatymus, kriptografinį algoritmą galima pakeisti į RC4 taip pat su 128 bitų raktu. Failai iš karto užšifruojami įrašant į išorinę atmintį (storage card) bei iššifruojami iš jos skaitant.

Pagrindiniai šios sistemos trūkumai:

- ✓ Failai įrašyti prieš įjungiant šią sistemą lieka neužšifruoti;
- ✓ Išėmus kortelę, perkaityti duomenų kitame įrenginyje neįmanoma;
- ✓ Perleidus OS (hard reset), šifravimo raktas prarandamas ir duomenys tampa nebeprieinami.

Mokslininkų grupė [8] išnagrinėjusi įvairius kriptografinius algoritmus, pasiūlė, kad algoritmai gali būti skirstomi į tris grupes, atsižvelgiant į jų stiprumą, didėjančia tvarka taip:

- ✓ Algoritmų, kurie buvo išnagrinėti mokslo visuomenės (arba garbingų kriptografijos specialistų) ir kuriems buvo nustatyta rimtų trūkumų naudojimui (pvz., kaip CMEA). Taip pat į šią grupę yra įtraukiami algoritmai, kurie nebuvo aptarti dėl to, kad yra pernelyg silpni. Algoritmai, priklausantys šiai grupei: Cellular Message Encryption Algorithm (CMEA), DES, RC2 ir RC4. Nors šios grupės algoritmų naudojimas yra skatinamas, tačiau reikia atidžiai įvertinti jų trūkumus.
- ✓ Algoritmų, kurie nebuvo nagrinėti mokslo visuomenės (arba garbingų kriptografijos specialistų) dėl to, kad jie nepopuliarūs visuomenėje arba jie nėra atviro kodo (open source). Algoritmai priklausantys šiai grupei, turėtų būti panaudojami tik jei reikia būtent šių konkrečių arba jei yra daroma prielaida, kad informacijos nebuvimas apie šiuos algoritmus yra laikinas.
- ✓ Algoritmų, kurie buvo išnagrinėti mokslo visuomenės (arba garbingų kriptografijos specialistų) ir kuriems nebuvo nustatyta trūkumų (nedažna situacija), arba buvo nustatyti trūkumui, bet jais negalima pasinaudoti. Šie trūkumai paprastai yra nepanaudojami, jei jie reikalauja neįmanomo darbo laiko, viršija reikalingų galimų blokų dydžius, milžiniškus paieškų kiekius, susijusius su raktų paieška ar pasirinktų

kodų ir t.t. Kita nepanaudojamų trūkumų kategorija yra kai tai yra aišku, kad vienos dalies silpnumas negali būti išplėstas iki pilnos versijos šifro. Aišku, šios grupės algoritmai yra rekomenduojami naudoti praktikoje.

2.3. Kriptografiniai metodai, skirti saugos užtikrinimui

Darbe bus nagrinėjamas delninio kompiuterio saugaus turinio užtikrinimas, naudojant failų užšifravimo/iššifravimo algoritmus taikant kriptografinius metodus. .NET sistemoje yra realizuotos kriptografijos klasės. Jos palaiko visus šiuo metu plačiai naudojamus kriptografinius algoritmus:

- ✓ DES (Digital Encryption Standard) – simetrinis blokų šifratorius;
- ✓ 3DES (Triple DES) – simetrinis blokų šifratorius, tvirtesnis už DES;
- ✓ Rijndael (Advanced Encryption Standard) – simetrinis blokų šifratorius;
- ✓ RC2 – simetrinis srauto šifratorius;
- ✓ RSA (Rivest, Shamir, and Adleman) – asimetrinis algoritmas šifravimui bei skaitmeniniam parašui;
- ✓ DSA (Digital Signature Algorithm) – asimetrinis algoritmas tik skaitmeniniam parašui;
- ✓ MD5 (Message digest) – saugus maišos algoritmas;
- ✓ SHA-1, SHA-256, SHA-384, SHA-512 – standartiniai saugūs maišos algoritmai;
- ✓ Pseudorandom Number Generator (PRNG) – pseudo atsitiktinis skaičių generatorius.

Kadangi delniniuose kompiuteriuose su Microsoft Windows CE operacine sistema yra naudojama .NET compact framework platforma, tai tyrimui bus naudojami tik šie užšifravimo/iššifravimo algoritmai:

- ✓ DES (Digital Encryption Standard);
- ✓ 3DES (Triple DES);
- ✓ Rijndael (Advanced Encryption Standard);
- ✓ RC2.

2.3.1. DES kriptografiniai algoritmai

Praėjusio amžiaus aštuntajame dešimtmetyje Horstas Feistelis (Horst Feistel) sukūrė blokų šifratoriaus architektūrą, kuri palaipsniui tapo duomenų šifravimo standartu (Digital Encryption Standard, DES) [9,10]. Šiame algoritme ta pati veiksmų seka ir tas pats 56 bitų raktas naudojami žinutės užšifravimui ir iššifravimui. Raktu yra parenkamas bet kuris iš 2^{56} peradresavimų. DES architektūra paremta 16 ciklų pakartojimu, kurių metu 64 bitų duomenų

blokas yra pastumiamas ir sukeičiamas, o po to paduodamas kitam ciklui. Tikslas yra įvesti painiavą ir išsisklaidymą užšifruojamame tekste kiekvieno ciklo metu, bet tokiu būdu, kad vėliau tą būtų galima iššifruoti. Sukeitimas įveda painiavą, kadangi pasidaro sunkiau rasti ryšį tarp užšifruojamo teksto (plaintext) ir jau užšifruoto teksto (chipertext). Postūmiai įveda išsisklaidymą, kadangi tai perskirsto informaciją, kas leidžia išsklaidyti statistinius pasikartojimus iš užšifruojamo teksto į užšifruotą tekstą.

Kadangi užšifruojamas tekstas dažniausia nebūna 64 bitų blokas, tai algoritmas visų pirma tekstą skaido 64 bitų blokais, paskutiniam priskirdamas trūkstamus bitus. Kiekvienas ciklas pasiima 64 bitų bloką, kuris po to skeliamas į 32 bitų blokelius. Dešinioji pusė užšifruojama specializuota šifravimo funkcija, naudojant „subraktą“, unikalų einamajam ciklui, o tada sudedama XOR operacija su kairiąja puse. Gauta dešinioji pusė persiunčiama kitam ciklui. Senoji dešinioji pusė pakeičiama kito ciklo kairiąja puse.

Tam, kad pagal užšifruojamą tekstą būtų dirbama su 64 bitų blokais, reikalingos gerai aprašytos taisyklės:

- ✓ Electronic Codebook (ECB) „Elektroninė kodų knyga" – leidžia dirbti su 64 bitų blokais nepriklausomai vienas nuo kito per visus 16 ciklų. Tačiau tai yra šio būdo trūkumas, kadangi mažėja saugumas. Šis būdas sudarinėja „kodų“ knygą iš užšifruojamo ir užšifruoto teksto porų. Jeigu įsilaužėliai sugebėtų sugeneruoti ką nors panašaus į šią knygą, saugumas pradingtų. Šiuo atveju slaptasis raktas net nepanaudojamas!
- ✓ Cipher Block Chaining (CBC) „Šifro blokų grandinės" – tai saugesnė technika, kuri nesudaro “kodų” knygos konstrukcijos. Algoritmas atlieka XOR operaciją kiekvienam 64 bitų užšifruojamo teksto blokui su praėjusio ciklo užšifruotu bloku. Pirmajai XOR operacijai yra sugeneruojamas pradinis vektorius, kadangi iki to užšifruojamas tekstas nebuvo užšifruotas.
- ✓ Cipher Feedback (CFB) „Išvesties atsakomoji reakcija" – šis būdas leidžia naudoti judančią aibę iš prieš tai sugeneruoto užšifruoto teksto ir per XOR operaciją su dabartiniu užšifruojamu tekstu pateikti kitam ciklui.
- ✓ Output Feedback (OFB) „Šifro atsakomoji reakcija" – šis būdas skiriasi nuo CFB tik tuo, kad jis naudoja judančią seką kitam ciklui, paimtą prieš atliekant XOR operaciją.

Triple DES naudojamas kaip stipresnė DES algoritmo alternatyva. Jis yra padarytas pagal DES algoritmo schemą ir atgalinė jo schema yra suderinama su tikruoju DES algoritmu. Pagerinimas yra tas, jog kiekvienas 64 bitų blokas yra užšifruojamas tris kartus, naudojantis DES algoritmu panaudojant tris skirtingus raktus. Triple DES užšifruoja kiekvieną bloką

pirmu raktu, tada iššifruoja rezultatą naudodamasis antru raktu ir galų gale užšifruoja vėl panaudojant trečią raktą. Tačiau, kaip galima pastebėti, algoritmo veikimo laikas praktiškai pailgėja tris kartus lyginant su DES algoritmu.

2.3.2. Rijndael kriptografinis algoritmas

Kitaip nei DES algoritmas, Rijndael (kitaip AES) [11] algoritmas yra kiek lankstesnis – gali naudoti kelis raktų dydžius (128, 192, 256 bitų). Jeigu DES algoritme duomenų blokas turėjo būti 64 bitų, tai čia blokai gali dirbti su 128, 192, 256 bitų blokais. Vykdomų ciklų skaičius Rijndael algoritme priklauso nuo rakto ir bloko dydžio.

Lentelė Nr. 1. Rijndael algoritmo raktų, blokų, ciklų sąrašas

	Rakto ilgis	Bloko dydis	Ciklų skaičius
AES – 128	128	128	10
AES – 192	192	192	12
AES – 256	256	256	14

Skaičiavimo metu baitas yra laikomas aštuonių bitų seka, sudarančią polinomą:

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

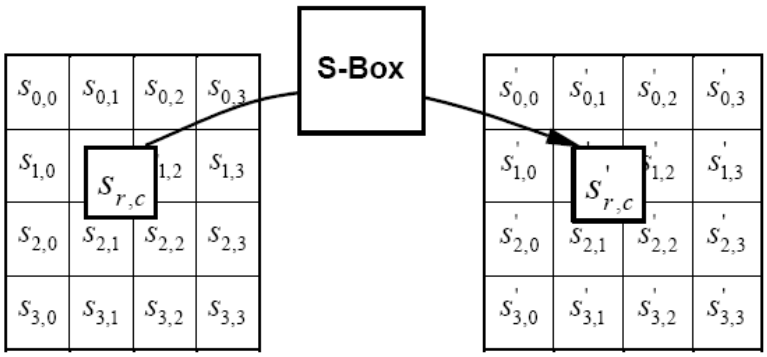
Algoritmo operacijos yra atliekamos su dvimačiu duomenų masyvu, vadinamu esama būsena. Šis masyvas susideda iš keturių eilučių baitų, kurių kiekvienoje yra bloko ilgio dalinto iš 32 stulpelių skaičius. Patogumo dėlei, duomenys yra imami po keturis kiekvieno baito bitukus ir verčiami į šešioliktainį kodą. Užšifravimui ir iššifravimui AES algoritmas naudoja ciklo funkciją, kuri susideda iš keturių skirtingų į baitus orientuotų transformacijų:

- ✓ Sukeisti baitus, naudojantis sukeitimų lentele;
- ✓ Sukeisti einamo masyvo eilutes pagal skirtingus atskaitos taškus;
- ✓ Maišyti duomenis kiekviename einamo masyvo stulpelyje;
- ✓ Pridėti ciklo raktą esamai būsenai.

Baitų sukeitimo transformacija nėra tiesinė sukeitimo operacija. Sukeitimų lentelė, kuri yra paversta (invertible), yra formuojama dviem žingsniais – (i) imama tam tikrų laukų daugyba (reikia nepamiršti, jog dauginami polinomai); (ii) pritaikoma afinioji transformacija [12]:

$$b_i^{'} = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i$$

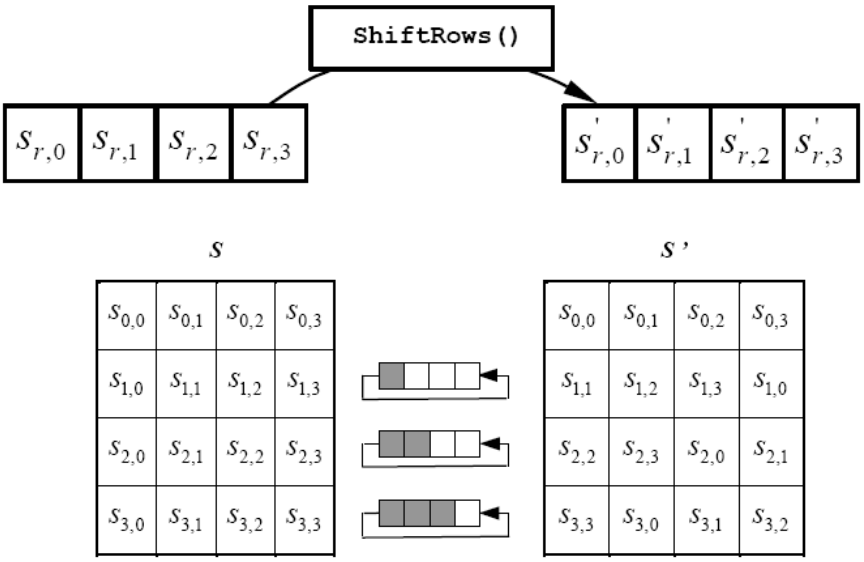
Taigi atliekama tokia transformacija:



3 pav. Sukeitimo operacijos bendroji schema

Čia S-Box yra sukeitimų matrica. Tarkime, jeigu apibrauktas elementas pradžioje yra {53}, tai iš sukeitimų lentelės reikia imti elementą, esantį sukeitimų lentelės 5 eilutės ir 3 stulpelio susikirtime, ir į naują vietą įrašyti ten esančią reikšmę.

Eilutės pastūmimo operacijoje atliekamas paskutinių eilučių ciklinis postūmis per skirtingą baitų skaičių. Pirmoji eilutė nestumiama.



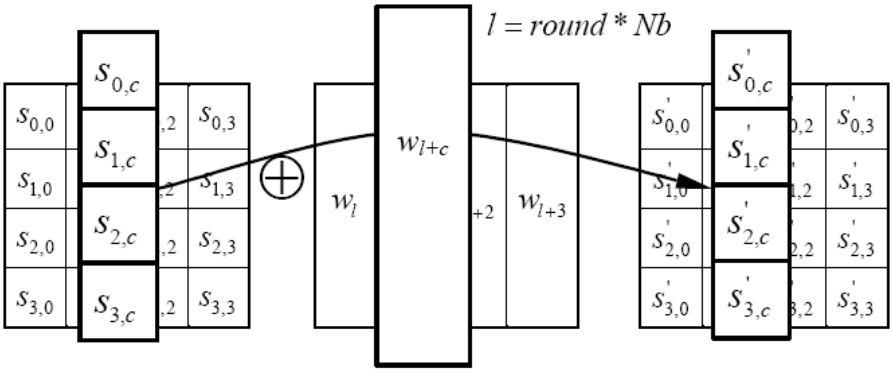
4 pav. Eilučių postūmio bendroji schema

Stulpelių maišos transformacija kiekvieną stulpelį ima kaip ketvirtos eilės polinomą. Tada yra atliekama specifinė daugyba su šiais polinomais (5 pav.). Taip gaunamos stulpelių reikšmės.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

5 pav. Polinomų daugyba

Ciklo rakto transformacijoje ciklo raktas yra pridedamas esamai būsenai pagal paprastą XOR operaciją (6 pav.). Prasisukus visiems ciklams gaunama užšifruota reikšmė.



6 pav. Ciklo rakto pridėjimo bendroji schema

2.3.3. RC2 kriptografinis algoritmas

RC2 kriptografinis algoritmas yra simetrinis blokinis šifras, naudojantis 64 bitų įėjimo duomenų bloko dydį [13]. Šis algoritmas kurtas tam, kad būtų pagerintas DES algoritmo veikimo laikas. Be šio pagerinimo buvo gautas kiek saugesnis užšifravimo algoritmas, kadangi jis naudoja kintamo dydžio raktą (nuo vieno iki 128 baitų).

Trumpai apie patį algoritmo veikimą:

1) inicializuoti žodžius R[0], ..., R[3] taip, kad jie turėtų 64 bitų įėjimo reikšmę;
2) išplėsti raktą K[0],..., K[63] – rakto išplėtimas atliekamas norint modifikuoti rakto buferį taip, kad kiekvienas išplėsto rakto bitas priklausytų nuo kiekvieno pateikto pradinio rakto bito sudėtingu būdu (dažnai tai būna skaičiaus π skaitmenų postūmio lentelė, pateikta šešioliktainiame kode);
3) parametrą *j* priskirti nuliui – šis parametras iš eilės eina per visus rakto K bitus;
4) atliekami penki miksavimo ciklai – bendrai tarus, miksavimo algoritmas yra ciklinis žodžio postūmis kairėn, pridedant einamojo rakto bitų reikšmes prie žodžio bitų reikšmių;
5) atliekamas vienas maišos ciklas – bendruoju atveju maišos funkcija nusakoma formule:

$$R[i] = R[i] + K[R[i-1]];$$

6) atliekami šeši miksavimo ciklai;
7) atliekamas vienas maišos ciklas;
8) atliekami penki miksavimo ciklai.

Kiekvieno miksavimo ciklo metu yra naudojami vis atskiri keturi raktai. Taip yra panaudojami visi išplėsti raktai.

2.3.4. RSA ir DSA kriptografiniai algoritmai

Algoritmo veikimas. Pirma yra sugeneruojama slaptojo ir viešojo raktų pora. Vienas svarbiausių dalykų generuojant raktus, yra kuo didesnis atsitiktinumas raktuose – tada bus gerokai sunkiau juos „nulaužti“. Naudojantis RSA [14] algoritmu reikia užšifruoti duomenis su viešuoju raktu. Iššifruoti duomenis tenka naudojantis slaptuoju raktu, vėliau sutikrinant duomenis su tikraisiais duomenimis. Kaip veikia raktų generavimas? Galime panagrinėti pavyzdį.

Atsitiktinai pasirinkime pirminius skaičius p ir q. Tam, kad algoritmo veikimas būtų stabilesnis, šie skaičiai turėtų skirtis vienas nuo kito ir būti pakankamai dideli (bent 1024 bitų ilgio). Turint šiuos skaičius, reikia rasti jų sandaugą. Toliau raskime Oilerio fi (φ) funkciją pagal formulę $\varphi = (p - 1)\cdot(q - 1)$. Turėdami šiuos du skaičius galime ištrinti p ir q skaičius, kadangi jie mums nebereikalingi. Juolab, kad tai užtikrina, jog mūsų pasirinktų skaičių niekas nesužinos ir taip bus sunkiau „nulaužti“ sugeneruotus raktus.

Toliau reikia vėl atsitiktinai pasirinkti skaičių e, kuris turi būti didesnis už 1 ir mažesnis už φ. Vėliau apskaičiuojame skaičių d, kuris randamas pagal formulę $d \cdot e = 1 \pmod{\varphi}$. Šį skaičių d reikia laikyti paslaptyje. Jeigu yra žinoma φ reikšmė, tai d reikšmę nesunkiai gausime pasinaudoję skaičiumi e. Jeigu žinomas n (kuris yra viešas), gauti φ reikšmę, nežinant p ir q reikšmių, yra gana sudėtinga. Taigi, slaptoji d reikšmė kartu su n atitinka slaptąjį raktą.

Bendra DSA algoritmo schema – pasinaudojus viešuoju raktu pasirašoma siunčiama žinutė, kuri iš pradžių yra paverčiama į duomenų seką, naudojantis maišos funkcija [15]. Siuntėjas užšifruoja gautą žinutę savuoju slaptuoju raktu, kad būtų sukuriamas siuntėjo asmeninis skaitmeninis parašas. Gaunant žinutę ir parašą, gavėjas iššifruoja parašą naudodamasis siuntėjo viešuoju raktu tam, kad būtų galima gauti žinutės seką ir vėl tuo pačiu maišos algoritmu užšifruoti tą žinutę. Jeigu žinutės seka sutampa su gauta iš siuntėjo, gavėjas gali būti tikras, jog žinutė nebuvo pakeista siuntimo metu.

2.3.5. Maišos algoritmai

Maišos funkcijos rezultatas visada būna vieno ilgio tam tikra duomenų seka. Geros maišos funkcijos savybė yra ta, kad du skirtingi įvedami duomenys niekada neturėtų duoti tokios pačios maišos funkcijos reikšmės. Ši savybė reiškia, jog nors maišos funkcijos rezultatas yra nedidelis, tačiau labai reprezentatyvus. Ir rasti dvi skirtingas duomenų sekas, duodančias tą patį maišos funkcijos rezultatą, būtų labai sudėtinga.

Pagrindinės klasės .NET framework platformoje yra MD5 ir SHA algoritmų klasės [16]. Šie algoritmai yra susiję vienas su kitu, kadangi abu yra kilę iš MD4 algoritmo.

Kiekvienas iš jų turi savų privalumų ir savų trūkumų. Tačiau viskas priklauso nuo to, kur ir kam jie yra naudojami.

2.4. Kriptografinių metodų energijos suvartojimo tyrimai

Mokslininkai Ramnath Venugopalan, Prasanth Ganesan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller, Mihail Sichitiu [17] tyrimui pasirinko bendrus kriptografinius algoritmus (simetrinio šifravimo bei maišos), kurie yra neatsiejama daugelio saugumo protokolų dalis. RC4 yra naudojamas IEEE 802.11 WEP, IDEA, ir MD5 yra PGP dalis, SHA-1, MD5, yra įtraukti į saugumo architektūros Interneto Protokolą (IPSEC). Šie algoritmai pateikia plačias pritaikymo ribas ir apima skirtingas matematinius ir duomenų apdorojimo operacijas. Jie dirba su įvairių dydžių duomenimis nuo 8 iki 32 bitų, vadinasi leidžia pasiekti efektyvumą įvairių architektūrų sistemose.

Lentelė Nr. 2. Eksperimente naudojami delniniai kompiuteriai

Platforma	Architektūra bitai	Dažnis MHz
Atmeaa 103	8	4
Atmesa 128	8	16
M16C/10	16	16
SA-1110	32	206
PXA250	32	400
UltraSparc2	64/32	440

Tyrimui buvo pasirinkti 5 skirtingų gamintojų ir skirtingų technologinių lygių delniniai kompiuteriai (lenetelė Nr. 2) nuo žemiausio lygio (4 MHz 8-bit Atmel AVR Atmega 103) iki aukščiausio lygio (400 MHz 32-bit Intel XScale).

Lentelė Nr. 3. Kriptografiniai metodai

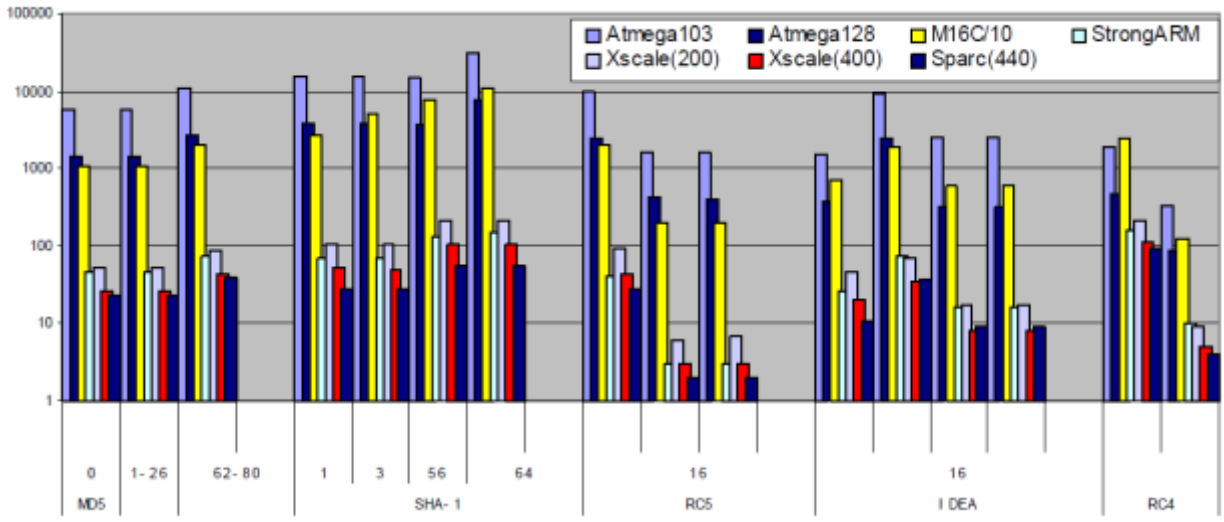
Algoritmas	Tipas	Raktas/Maiša bitai	Blokas bitai
RC4	srautinis	128	8
IDEA	blokinis	128	64
RC5	blokinis	64	64
MD5	maiša	128	512
SHA1	maiša	128	512

Palyginimui, tyrimui buvo įtraukta ir darbo stotis (440 MHz 64-bit SPARC CPU, veikianti 32-bit rėžimu) [17].

Eksperimentai buvo atlikti kiekvienam delninių kompiuterių architektūros tipui su visais saugos algoritmais (lentelė Nr. 3). Kiekvienai iš platformų buvo įdiegti/vykdomi tokie pat saugumo mechanizmai t.y. be jokių pakeitimų.

Įvesties ilgis buvo skirtingas užšifravimui, maišos pagrindu su fiksuoto dydžio paketais, siekiant pasiekti algoritminio užpildymo iki paketo ilgio efektą. Maišos algoritmai naudoja paprastąjį tekstą, kuris atitinka konkrečias baitų ribas.

Jei paprastasis tekstas nėra bloko dydžio sveikas daugiklis, jis yra užpildomas. RC5 ir IDEA algoritmai su duomenimis, kurių bloko dydis yra 64 bitai. MD5 ir SHA-1 algoritmai naudoja 512 bitų blokus. Tekstas, su kuriuo bandyti simetrinės kriptografijos algoritmai yra 128 bitų ilgio. Vienai architektūrai, XScale, eksperimentai buvo atliekami su dviem dažnio parametrais, 200 MHz ir 400 MHz, kai atminties pasiekimo laikas nepakito. Kiekviena iš algoritmo dalių, tokių kaip inicializacija, užšifravimas ir iššifravimas, buvo įvykdyta 1000 kartų su tais pačiais įvedimo parametrais, o iš rezultatų buvo išvesta vidutinė reikšmė (7 pav.).



7 pav. Tyrimo rezultatai Laikas ms.

Po tyrimo pateiktos tokios išvados [17]:

- ✓ Kaip ir buvo tikėtasi, silpniausių charakteristikų (architektūros) delninis kompiuteris ilgiausiai vykdė analizuojamus kriptografinius algoritmus, tuo daugiau išnaudodamas akumuliatoriaus energijos.
- ✓ Nors silpniausių charakteristikų (architektūros) delniniame kompiuteryje vykdant RC5 ir RC4 kriptografinius algoritmus sugaištas laikas beveik tas pats, galingesnių delninių kompiuterių rezultatai parodė, kad RC5 kriptografinis algoritmas yra 3 kartus greitesnis nei RC4, kas nulemia mažesnes energijos sąnaudas.

- ✓ Lyginant MD5 ir SHA-1 kriptografinius algoritmus pastebėta, kad MD5 yra žymiai greitesnis nei SHA-1.

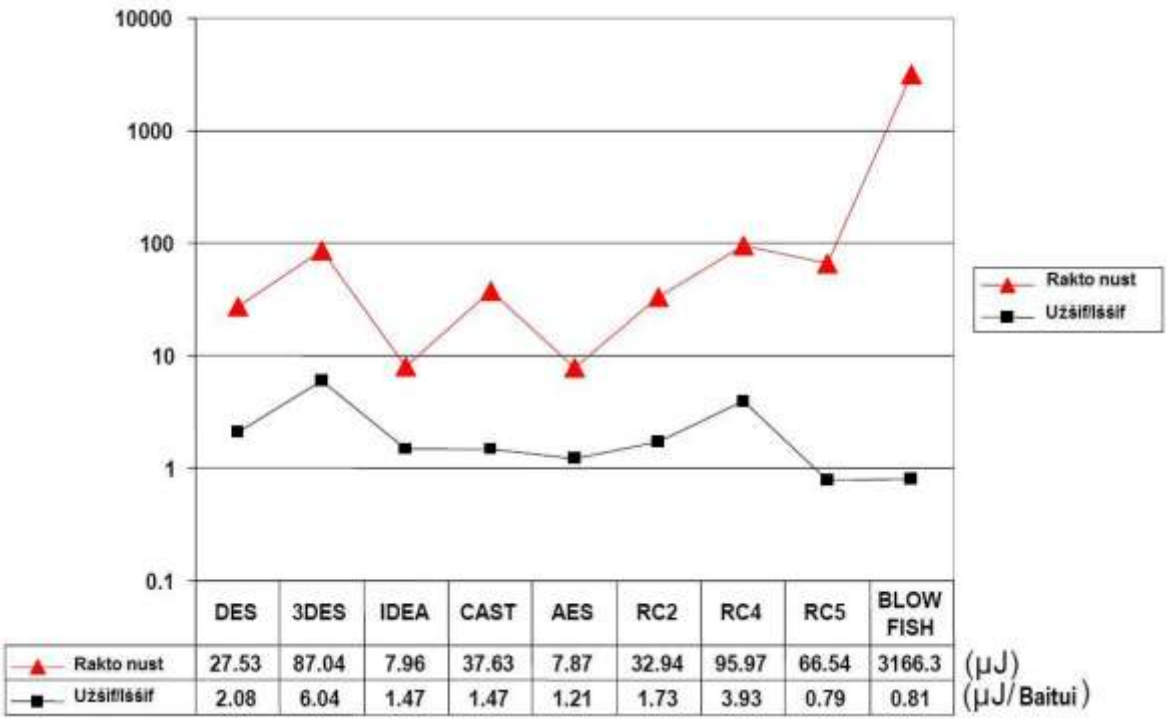
Detalesni rezultatai pateikti 4 lentelėje.

Lentelė Nr. 4. Tyrimo rezultatai

Algoritmas	**Dydis (baitai)**	**Veiksmas**	**Atmega 103**	**Atmega 128**	**M16 C/10**	**Strong ARM**	**Xscale (400)**	**Xscale (200)**	**Sparc (440)**
MD5	0	Maiša	5863	1466	1083	46	26	53	23
	1-26	Maiša	5890	1473	1075	46	26	53	23
	62-80	Maiša	10888	2722	2011	74	45	90	39
SHA-1	1	Maiša	15249	3812	2651	69	51	102	27
	3	Maiša	15781	3945	5303	69	50	103	27
	56	Maiša	14543	3636	7955	133	102	205	55
	64	Maiša	31107	7777	10907	145	103	207	56
RC5	16	Inicializacija	9641	2410	2074	41	45	91	28
		Užšifravimas	1651	413	197	3	3	6	2
		Iššifravimas	1636	409	202	3	3	7	2
IDEA	16	Inicializacija užšifravimui	1523	381	727	26	21	47	11
		Inicializacija iššifravimui	9417	2354	1927	76	35	69	36
		Užšifravimas	2555	325	596	16	8	17	9
		Iššifravimas	2614	325	597	16	8	17	9
RC4		Inicializacija	1886	472	2455	155	108	216	96
		Užšifravimas	344	86	123	10	5	9	4

Kita mokslininkų grupė Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan bei Niraj K. Jha [18] atliko tyrimą pasirinkdami ne keletą mobilių įrenginių, bet vieną. Tyrimas buvo vykdomas pritaikant kliento-serverio technologiją, kur klientas (delninis kompiuteris) prie LAN tinklo buvo prijungtas bevielio tinklo technologija (WLAN). Pasirinktas delninis kompiuteris: Compaq iPAQ H3670, su Intel SA-1110 StrongARM procesoriumi 206MHz. Atmintis 64MB RAM ir 16MB FlashROM. Prie bevielio tinklo jungiasi naudojant Cisco Aironet 350 series WLAN kortą. Maitinimo akumuliatorius - Ličio-jonų 950 mAh.

Atskirų kriptografinių algoritmų energijos suvartojimo vertės gaunamos vykdant kriptografinius metodus klientinėje dalyje (delniniame kompiuteryje). Paveiksle 8 pateikti skirtingų simetrinių kriptografinių algoritmų matavimo rezultatai.



8 pav. Simetrinių kriptografinių algoritmų matavimo rezultatai

AES užšifravimo algoritmo energijos sąnaudos pateiktos lentelėje Nr. 5.

Lentelė Nr. 5. AES užšifravimo algoritmo energijos sąnaudos naudojant skirtingo dydžio raktus bei taisykles

Rakto dydis	Rakto nustatymas µJ	ECB µJ/B	CBC µJ/B	CFB µJ/B	OFB µJ/B
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

Lentelėje Nr. 6 pateikti skirtingų maišos funkcijų matavimo rezultatai.

Lentelė Nr. 6. Maišos funkcijų matavimo rezultatai

Algoritmas	MD2	MD4	MD5	SHA	SHA1	HMAC
Energijos sąnaudos µ J / B	4.12	0.52	0.59	0.75	0.76	1.16

Lentelėje Nr. 7 pateikti skirtingų asimetrinių kriptografinių algoritmų matavimo rezultatai.

Lentelė Nr. 7. Skaitmeninio parašo algoritmų energijos sąnaudos

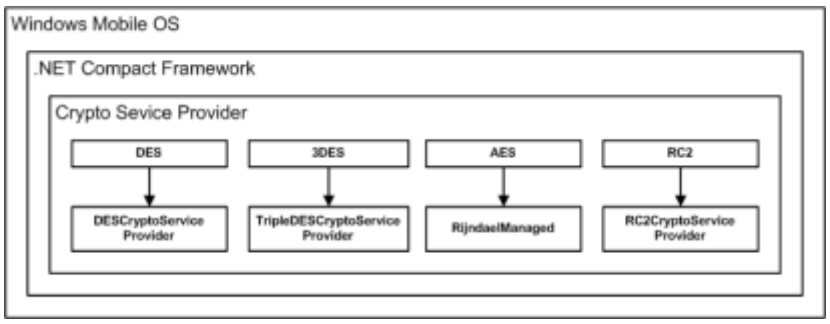
Algoritmas	Rakto dydis bitai	Rakto generavimas *(mJ)*	Pasirašymas *(mJ)*	Patvirtinimas *(mJ)*
RSA	1024	270.13	546.5	15.97
DSA	1024	293.20	313.6	338.02
ECDSA	163	226.65	134.2	196.23

Po tyrimo pateiktos tokios išvados [18]:

- ✓ Matavimai buvo atlikti pagal ECB (Electronic Codebook) taisykles;
- ✓ AES algoritmas naudoja mažiausiai energijos šifravimo funkcijoms vykdyti;
- ✓ 3 DES algoritmas naudoja daugiausia energijos šifravimo funkcijoms vykdyti;
- ✓ Rakto dydis turi nemažą įtaką energijos sąnaudoms;
- ✓ Maišos algoritmas MD2 yra labiausiai energiją naudojantis algoritmas;
- ✓ Maišos algoritmai MD4 ir MD5 yra mažiausiai energiją naudojantys algoritmai;
- ✓ Nors maišos algoritmai SHA ir SHA1 yra naujesni, nei MD4 IR MD5, tačiau savo algoritme turi daugiau žingsnių, kas nulemia jų didesnes energijos sąnaudas;
- ✓ Asimetrinių kriptografinių algoritmų vertinimas dviprasmiškas, nes skirtingi žingsniai naudoja skirtingus energijos kiekius, todėl vienareikšmiškai pasakyti, kuris yra mažiau energijos naudojantis algoritmas, sudėtinga.

2.5. Kriptografinių algoritmų įtakos energijos suvartojimui delniniuose kompiuteriuose tyrimo motyvacija

Microsoft ® pateikia modernią .NET framework platformą, kuri turi kriptografinių algoritmų servisą (8 pav.), kuris pateikia metodus informacijos užšifravimui ir iššifravimui delniniuose kompiuteriuose (PDA), naudojant DES, 3DES, AES, RC2 algoritmus [5].



8 pav. Kriptografijos servisai

.NET aplinka savyje apima didelę biblioteką suprogramuotų sprendimų išsprendžiančių įvairias programavimo problemas bei virtualią aplinką, kuri tvarko/valdo

programų vykdymą, parašytų būtent tai aplinkai. Microsoft ® rekomenduoja ir .NET platformą, kuri yra skirta naudojimui daugumoje naujų programinės įrangos realizacijų, skirtų Windows OS.

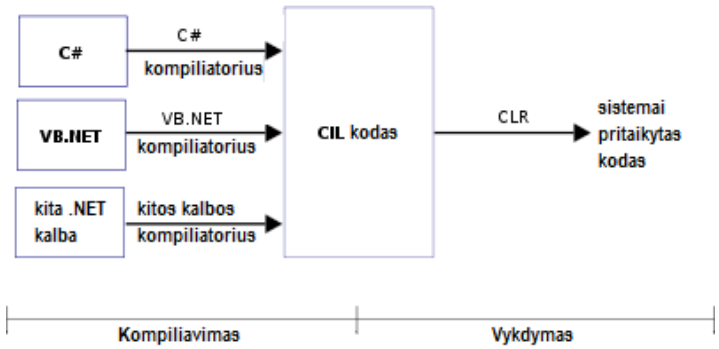
.NET framework Bazinės klasės biblioteka Base Class Library suteikia didelį kiekį savybių apimančių vartotojo sąsają (user interface), duomenų prieigą (data access), duomenų bazių sujungimus (database connectivity), kriptografiją (cryptography), žiniatinklių kūrimą (web application), skaitmeninius algoritmus (numeric algorithms) ir tinklinį duomenų apsikeitimą (network communications).

Programinė įranga sukurta .NET framework įrankiais, vykdoma programinėje aplinkoje, kuri tvarko/valdo įrangos vykdomosios aplinkos (runtime) reikalavimus. Ši aplinka yra .NET framework dalis ir yra vadinama Common Language Runtime (CLR). CLR suteikia programinės įrangos virtualią aplinką, todėl programų kūrėjui nereikia galvoti apie procesoriaus (CPU), kuris vykdys programą, galimybes,. CLR taip pat suteikia kitus ne mažiau svarbius servisus, tokius kaip saugumas (security), atminties valdymas (memory management) ir išimčių tvarkymas (exception handling). Klasių biblioteka bei CLR kartu sudaro .NET Framework platformą. Šios platformos sudedamosios dalys pavaizduotos 10 paveiksle [20].

10 pav. .NET framework sudedamosios dalys

Kaip ir parodyta 11 paveiksle su .NET suderinamos kalbos yra kompiliuojamos į antrinę nuo platformos nepriklausomą kalbą Common Intermediate Language (CIL). Toliau

nuo platformos priklausoma Common Lanuage Runtime (CLR), kompiliuoja CIL kodą į mašinai suprantamas komandas, kurios gali būti vykdomos toje specifinėje platformoje.



11 pav. .NET Framework kodo apdorojimas

Pagrindinės su .NET suderinamos kalbos yra šios:

C#, Visual Basic .NET, C++/CLI (Managed), F#, J#, JScript .NET, Windows PowerShell.

.NET Compact Framework paveldi visą bendrinės kalbos aplinkos (CLR) .NET Framework architektūra valdomo kodo vykdymui [21]. Platforma pateikia suderinamumą su Windows CE operacinę sistemą turintį prietaisą, kas leidžia naudoti esamas funkcijas ir integruoti savo nuosavus komponentus į aplikaciją. .NET Compact Framework platformos architektūra pateikta 12 paveiksle.

12 pav. .NET Compact Framework platformos architektūra

.NET Compact Framework platforma naudoja Windows CE operacinę sistemą pagrindinių funkcijų bei keletui specialių įrenginiui ypatybių vykdymui. Tokios dalys kaip Windows formos, grafinė aplinka, paveikslai, interneto paslaugos buvo pertvarkytos, kad efektyviai veiktų įrenginiuose, o ne tiesiogiai perkopijuoti iš .NET Compact Framework.

.NET Compact Framework platfoma suteikia šiuos suderinamumo kriterijus su Windows CE OS:

- ✓ Suderinamumas su vietiniais saugumo reikalavimais;
- ✓ Pilna integracija su įrenginio diegimo programomis;
- ✓ Sąveika su įrenginio kodu, naudojant COM, Interop ir platformos iškvietimą.

Pagrindinė .NET Compact Framework programavimo kalba yra C#. Pagrindinės savybės yra šios [21]:

- ✓ Moderini;
- ✓ Objektiškai orientuota;
- ✓ Unifikuota kintamųjų atžvilgiu;
- ✓ Leidžianti programuotojams lengvai kurti aplikacijas .NET Framework bei .NET Compact Framework platformose.

2.6. Analizės išvados

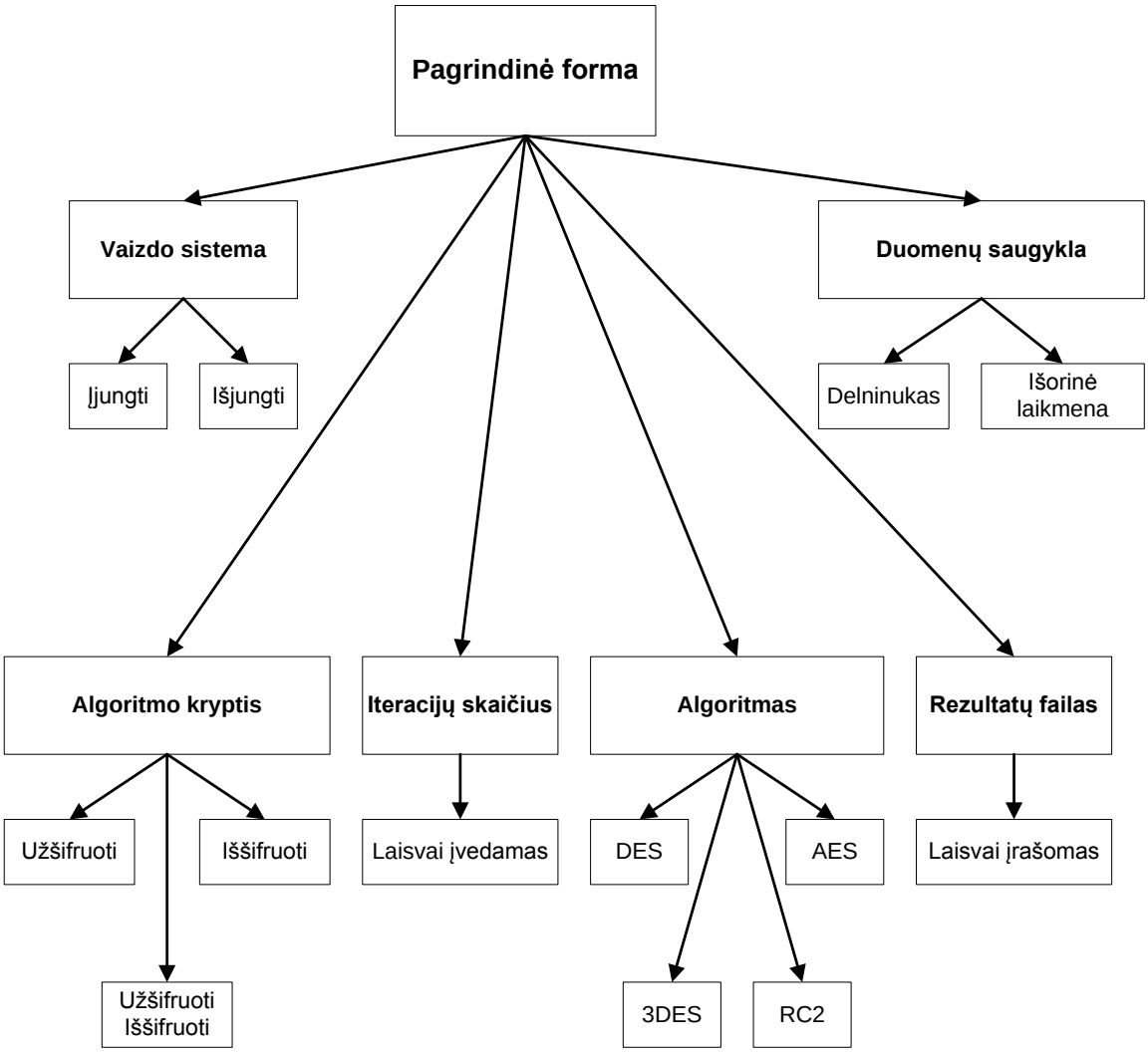
- ✓ Svarbi problema yra akumuliatoriaus energijos sunaudojimas delniniuose mobiliuose įrenginiuose.
- ✓ Tiksliai žinant energijos sąnaudas, naudojamas saugos algoritmų, galima užtikrinti reikiamo funkcionalumo panaudojimą, protingas/įvertintas energijos sąnaudas bei reikiamo saugumo lygio užtikrinimą tuo pačiu metu.
- ✓ Pasinaudojant šiais duomenimis galima kurti šiuolaikinius verslo bei saugos reikalavimus atitinkančią programinę įrangą.
- ✓ Moksliniuose straipsniuose daugiausia dėmesio skiriama interneto saugumo protokolų bei duomenų apsikeitimo tinkle saugumui ir tyrimams, o ne saugiam vidinių duomenų turiniui.
- ✓ Moksliniuose straipsniuose tyrimai atlikti įvairiausiais metodais ir priemonėmis. Vieni metodai yra teoriniai, o kiti praktiškai įgyvendinti.
- ✓ Magistrinio darbo tyrimui atlikti naudosiu savo sukurtą programinę įrangą (C#), kuri leis taikyti .NET compact framework‘e esančius kriptografinius metodus. Iš gautų rezultatų galėsime įvertinti kiek kuris kriptografinis metodas naudoja energijos resursų. Tyrimą atliksiu su delniniu kompiuteriu, kuris turi Windows Mobile 6.0 operacinę sistemą.
- ✓ C# moderni, objektiškai orientuota, programavimo kalba skirta .NET Framework bei .NET Compact Framework platformos klasių bibliotekos naudojimui.

3. PROGRAMINĖS ĮRANGOS DELNINIO KOMPIUTERIO ENERGIJOS SUVARTOJIMO TYRIMUI PROJEKTAVIMAS

Tyrimui atlikti yra sukurta programinė įranga. Programinė įranga suprogramuota Microsoft C# programavimo kalba. Programinė įranga užšifruoja ir iššifruoja tyrimui parinktus failus, įrašydama stebimus delninio kompiuterio parametrus į log failus, kurių pagrindu bus analizuojami rezultatai bei pateikiamos išvados.

3.1. Programos struktūra

Programinė įranga vykdo kriptografinius algoritmus t.y. užšifruoja ir iššifruoja failus naudojant skirtingus kriptografinius metodus. Vykdant tyrimą yra stebimi ir registruojami akumuliatoriaus parametrai: įtampa (V), bei energijos likutis (%). Kadangi ne visų delninių kompiuterių sisteminės bibliotekos išduoda įtampos (V) parametrus, bus stebimas tik akumuliatoriaus energijos likutis (%), tačiau sukurta programinė įranga leidžia stebėti bei registruoti ir įtampos (V) parametrą.



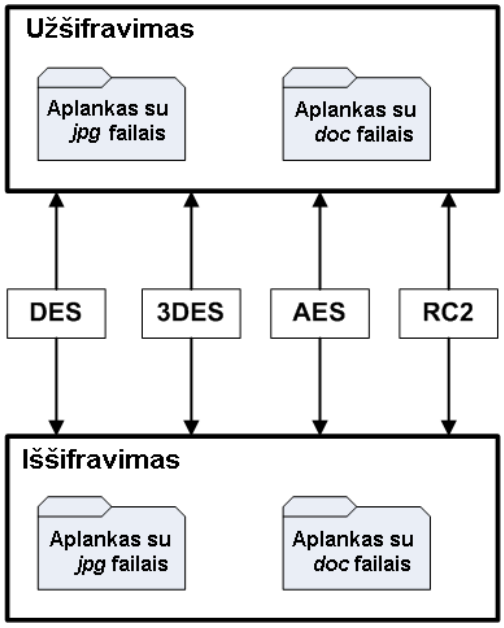
13 pav. Programos struktūra

Tyrimas bus atliekamas vykdant simetrinius kriptografinius algoritmus. Programoje realizuotas kriptografinių metodų taikymas - failų užšifravimas ir iššifravimas naudojant DES, 3DES, AES bei RC2 algoritmus. Sukurta programinė įranga leidžia parinkti algoritmo kryptį – visus tyrimo failus tik užšifruoti, tik iššifruoti, arba kiekvieną failą užšifruoti ir iššifruoti atskirai. Sukurta programinė įranga naudosis .NET Compact Framework platformoje esančiais kriptografiniais servisais. NET Compact Framework platformos klasių sąrašas pateiktas lentelėje Nr. 8. Skirtingiems kriptografiniams algoritmams naudojami skirtingo dydžio raktai. Raktų, kurie panaudoti programinėje įrangoje, skirti užšifruoti/iššifruoti failams, pateikti lentelėje Nr. 8.

Lentelė Nr. 8. Eksperimente naudojami kriptografiniai algoritmai, bei raktų dydžiai

Eil. Nr.	Algoritmas	Rakto dydis (bit)	.Net Framework klases pavadinimas
1.	DES	64	DESCryptoServiceProvider
2.	TripleDES	192	TripleDESCryptoServiceProvider
3.	RC2	128	RC2CryptoServiceProvider
4.	AES(RijnDael)	128	RijnDaelManaged

Metodai taikomi įvairaus tipo failams. Analizės metu nepavyko rasti statistikos, kokiems failų tipams svarbiausia atlikti kriptografinius algoritmus, todėl tyrimui parinkti dažniausiai naudojamų tipų failai. Tyrimui sukurtas katalogas „Tyrimas“. Siekiant atskirti skirtingų tipų failus, „Tyrimas“ kataloge sukurti katalogai (DOC, JPG), į kuriuos atitinkamai pagal tipą, yra įkeliami pradiniai failai (Dokumentai-tekstiniai (*.doc), Paveikslėliai-grafiniai (*.jpg)). Katalogų su failais struktūra pateikta 14 paveiksle.



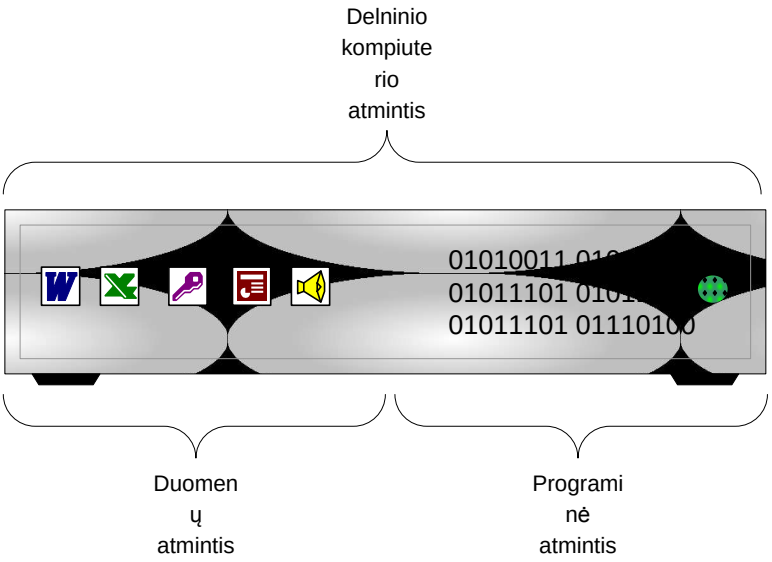
14 pav. Duomenų tyrimui katalogų ir failų struktūra

Sukurta programinė įranga neriboja failų pagal tipą, bet kurio tipo failus galima panaudoti tyrimui, t.y. galima užšifruoti/iššifruoti norimo tipo failus sukuriant norimo tipo aplanką ir patalpinant juos į jį.

Vykdant užšifravimo/iššifravimo algoritmus norint patikrinti, kad šifravimo funkcionalumas veikia teisingai yra išsaugomi užšifruoti ir vėl iššifruoti duomenys. Kad nesusimaišyti tarp duomenų apdoroti failai yra saugomi Tyrimas kataloge, bei yra išlaikomi tokie pat failų pavadinimai, kaip ir pradiniai, tik prie užšifruotų failo pavadinimo pradžioje pridedamas simbolis D, prie Iššifruotų simbolis E. Tai leidžia lengvai patikrinti ar sėkmingai įvyko algoritmas bei palyginti failų informaciją (failo dydis) .

Kaip žinome, vaizdo sistema yra viena iš pagrindinių energijos naudotojų delniniame kompiuteryje. Siekiant, kad tyrimo aplinka būtų kuo ergonomiškesnė bei tyrimo rezultatai būtų kuo tikslesni, buvo sukurta „Video Power Off“ funkcija, kurios pagalba vykdant eksperimentą bus galima išjungti delninio kompiuterio vaizdo sistemą, t.y. energija nebebus naudojama vaizdo rodymui, kol bus vykdomi kriptografiniai algoritmai. Tai užtikrins tikslesnius rezultatus – pagrindines energijos sąnaudas sudarys kriptografinių algoritmų energijos sąnaudos.

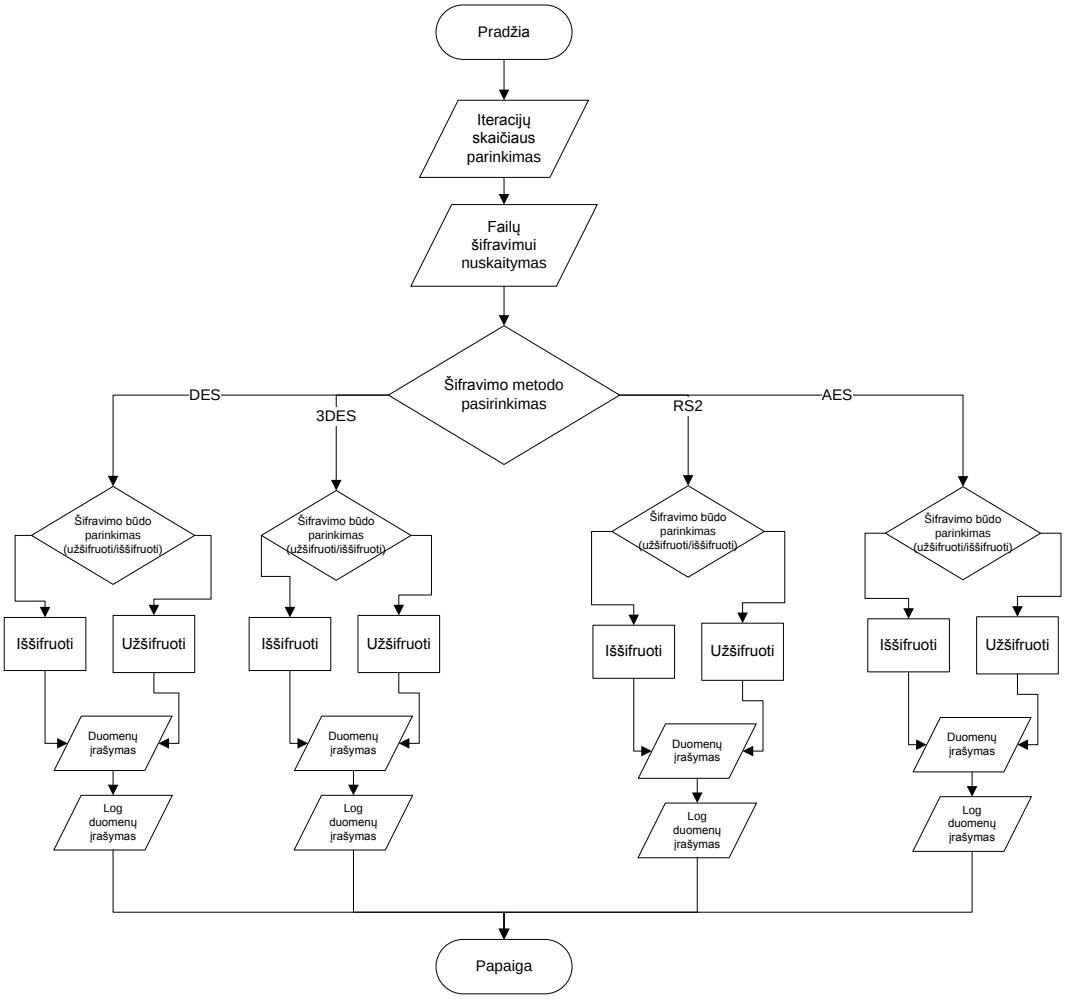
Kadangi delninio kompiuterio atmintis ribota (dalis esamos atminties naudojama programoms, o dalis failų saugojimui), kaip pavaizduota 15 paveiksle, o tyrimui bus naudojami įvairaus dydžio failai, programoje numatyta galimybė failus nuskaityti, bei įrašyti į išorinę atmintį (storage card).



15 pav. Atminties paskirstymas delniniame kompiuteryje

Siekiant tyrimą atlikti su kiek galima didesniais duomenų kiekiais bei tikintis tikslesnių ir kokybiškesnių rezultatų, kai netgi išorinės kortelės vietos duomenims gali būti

per mažai, yra naudojamos Iteracijos. Vietoj to, kad naudoti vieną didelį failą tyrimui, kas gali būti neįmanoma, yra naudojamas mažesnio dydžio failas, tačiau su juo yra atliekamas pasirinktas operacijų (iteracijų) skaičius. Taip pasiekiamas norimo dydžio duomenų apdorojimas. Apibendrintą programos veikimą galima pavaizduoti blokine schema (16 pav.).



16 pav. Apibendrinta programinės įrangos blokinė schema

Tyrimo rezultatai saugomi CSV tipo failuose (*.txt), kuriuos nesunku importuoti į grafinio apdorojimo sistemas, tokias kaip Microsoft Excel, SPSS. Siekiant tyrimo rezultatus apdoroti kuo lengviau ir efektyviau, programoje suprogramuota galimybė tyrimo rezultatus išsaugoti norimame faile (pavadinimas laisvai įrašomas) bei vietoje (lokaliame diske ar išorinėje kortelėje). Atliekant tyrimą failo pavadinime galima įrašyti algoritmo, kurį atliekame pavadinimą, kryptį, iteracijų skaičių, failų, su kuriais vykdome tyrimą, rūšį, bei/arba kitą svarbią informaciją, kas leis lengvai rūšiuoti bei apdoroti tyrimo rezultatus.

Kadangi tyrimo esmė yra akumuliatoriaus energijos suvartojimo tyrimas t.y. energija mažėja vykdant užšifravimą/iššifravimą, labai svarbu, kad tyrimo eigoje akumuliatorius visiškai neišsikrautų kol nebuvo užfiksuoti(įrašyti į failą) tyrimo rezultatai. Vizualiniu būdu

nustatyti energijos likutį, kai delninio kompiuterio vaizdo sistema yra išjungta, praktiškai neįmanoma.

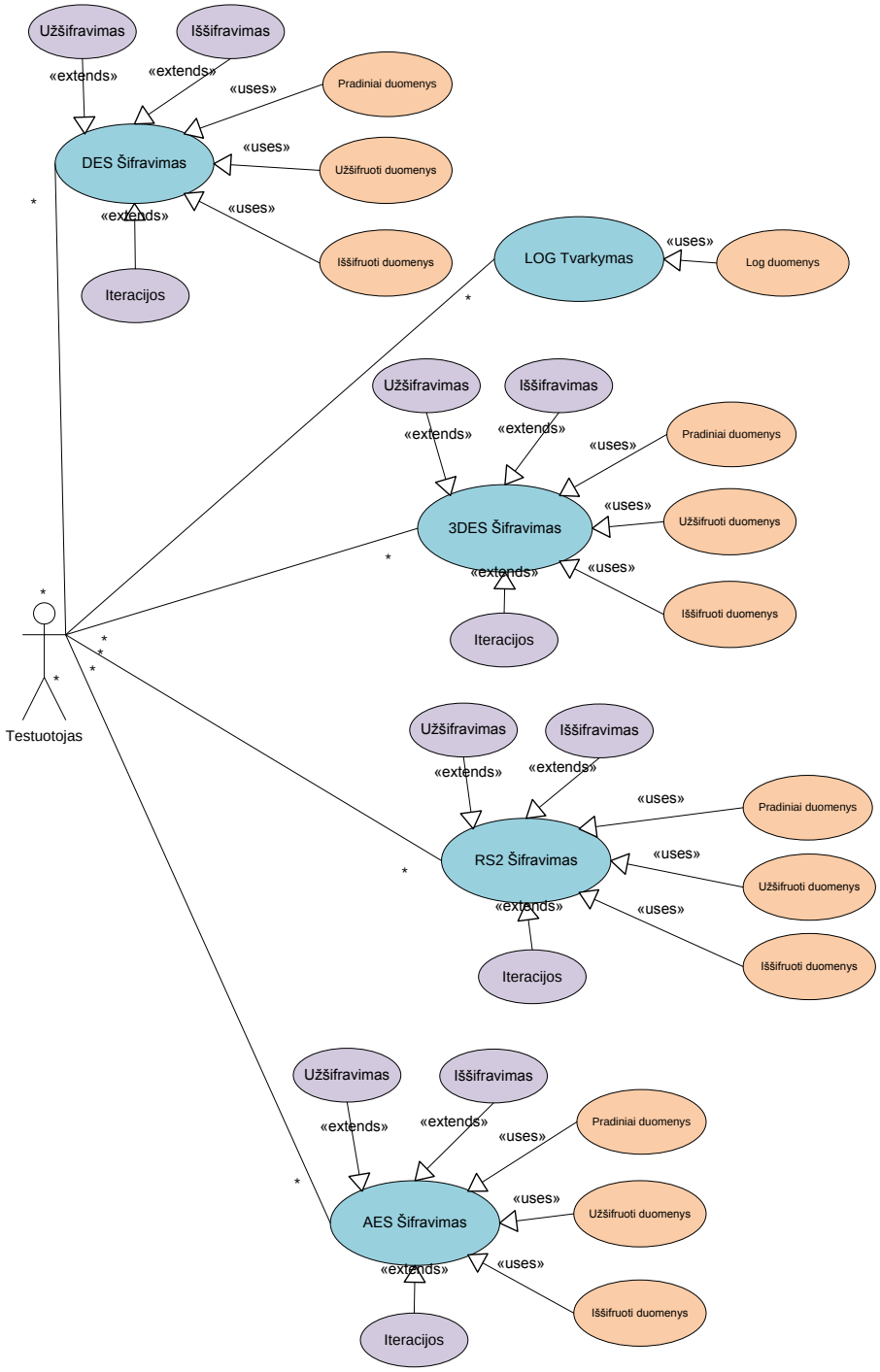
17 pav. Pranešimo tekstas

Tuo tikslu programoje yra realizuotas mechanizmas, kuris esant akumuliatoriaus energijos likučiui 15%, sustabdo užšifravimo/iššifravimo vykdymą, įrašo rezultatus, įjungia vaizdo sistemą, bei informuoja naudotoją informaciniu pranešimu. Pranešimo pavyzdys pateiktas 17 paveiksle. Šis funkcionalumas palengvina tyrimą: taip užtikrinama, kad tyrimo rezultatai bus užfiksuoti ir nebus prarasti.

Programos pagrindinės formos vaizdas pateiktas 18 paveiksle.

18 pav. Pagrindinės programos formos vaizdas

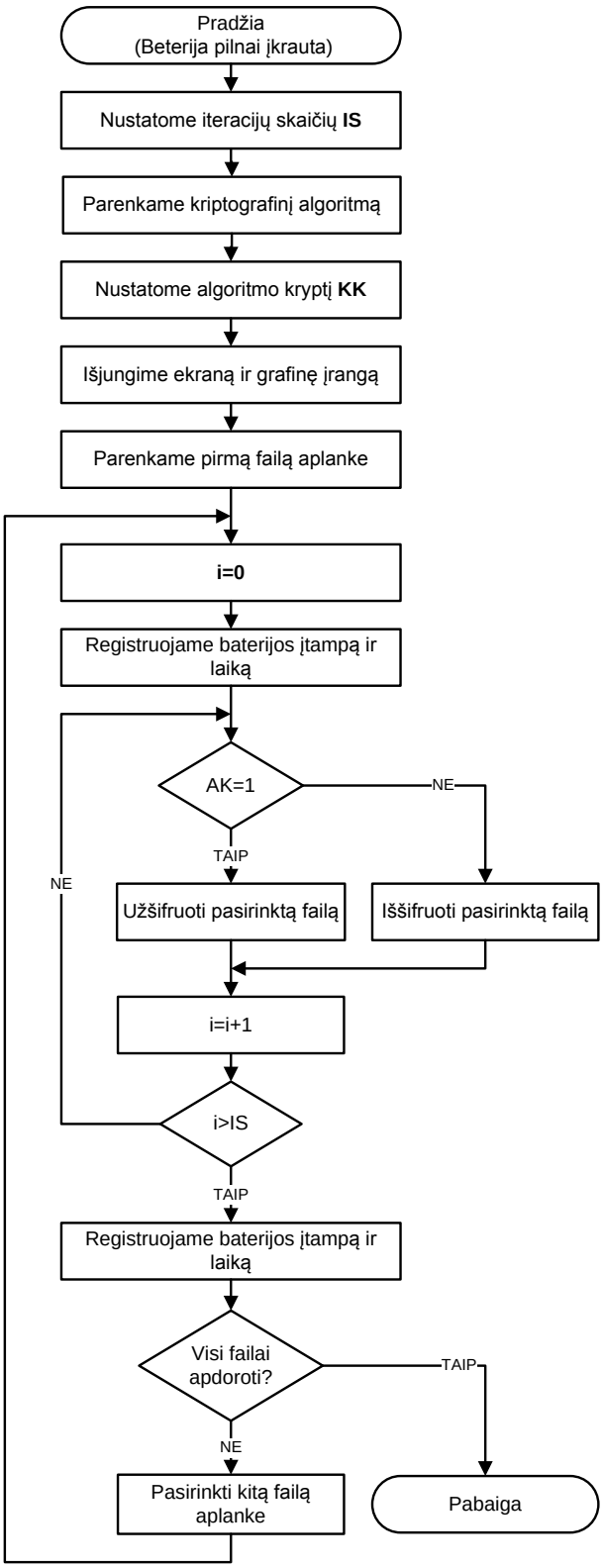
Panaudojimo atvejų schema pateikta 19 paveiksle.



19 pav. Panaudojimo atvejų schema

3.2. Programos aprašymas

Detali vieno kriptografinio algoritmo taikymo blokinė schema pateikta 20 paveiksle.



20 pav. Detali vieno kriptografinio algoritmo taikymo blokinė schema

Eksperimento rezultatai saugomi CSV tipo faile. Priklausomai nuo pasirinkimo prieš vykdant užšifravimą/iššifravimą rezultatų failą galima saugoti arba delninio kompiuterio atmintyje, arba išorinėje atmintyje (storage card). Log faile išsaugoma kriptografinė informacija (koks ir kokia kryptimi kriptografinis algoritmas buvo vykdomas), informacija apie failus, panaudotus eksperimente (failo pavadinimas, failo dydis), iteracijų skaičius (nusako kiek ciklų buvo atlikta), taip pat informacija apie akumuliatoriaus būseną (įtampa bei įkrova) prieš ir po failo apdorojimo, bei laikai prieš ir po failo apdorojimo. Tyrimui svarbiausias parametras yra energijos sąnaudos, tačiau ir kiti parametrai bus naudingi atliekant tyrimo rezultatų analizę bei leis pateikti įvairesnes/įdomesnes išvadas. Detalesnis log failo laukų aprašymas pateiktas Lentelėje Nr. 9.

Lentelė Nr. 9. Log failo laukų sąrašas su aprašymais

Eil. Nr.	Lauko pavadinimas	Lauko aprašymas
1.	ID	Eilutės identifikatorius
2.	Algoritmas	Vykdomas algoritmas (DES, 3DES, RS2, AES)
3.	Action	Veiksmas – algoritmo kryptis (1 – užšifravimas, 2 – iššifravimas, 3 – užšifravimas + iššifravimas vienam failui)
4.	File	Failo pavadinimas
5.	Size	Failo dydis
6.	IteracijuSk	Iteracijų skaičius
7.	BeValueP	Akumuliatoriaus energija (%) prieš vykdymą
8.	BeValueV	Akumuliatoriaus energija (V) prieš vykdymą
9.	AfValueP	Akumuliatoriaus energija (%) po vykdymą
10.	AfValueV	Akumuliatoriaus energija (V) po vykdymą
11.	BeTimeH	Laikas (valandos) prieš vykdymą
12.	BeTimeM	Laikas (minutės) prieš vykdymą
13.	BeTimeS	Laikas (sekundės) prieš vykdymą
14.	AfTimeH	Laikas (valandos) po vykdymo
15.	AfTimeM	Laikas (minutės) po vykdymo
16.	AfTimeS	Laikas (sekundės) po vykdymo
17.	SkirtP	Akumuliatoriaus sąnaudos (%)
18.	SkirtV	Akumuliatoriaus sąnaudos (V)
19.	SkirtLaikas	Laikas sunaudotas algoritmo vykdymui

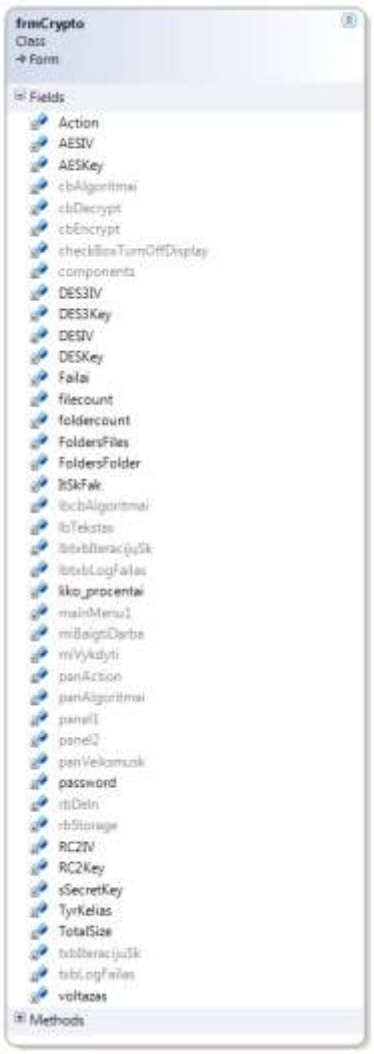
Kiekvieną kartą atlikus užšifravimo/iššifravimo algoritmą su norimais failais bei pasirinktu iteracijų skaičiumi į CSV failą yra įrašomi rezultatai. Rezultatų failo pavyzdys pateiktas 21 paveiksle.

```
ID,Algoritmas,Action,File,Size,IteracijuSk,Laikas,BeValueP,BeValueV,AfValueP,AfValueV,BeTimeH,
BeTimeM,BeTimeS,AfTimeH,AfTimeM,AfTimeS,SkirtP,SkirtV,SkirtLaikas,ItSkFak
1,,,JPG,7201086,,,,,,,,,,,,,,,,
2,3DES,2,\Storage Card\Tyrimas\JPG\7.JPG,1915552,70,2009-12-07
23:03:55,100,0.00,80,0.00,23,3,55,0,47,24,20,0.00,6209,70
2,3DES,2,\Storage Card\Tyrimas\JPG\6.JPG,1703597,70,2009-12-08
00:47:24,80,0.00,61,0.00,0,47,24,2,19,36,19,0.00,5532,70
2,3DES,2,\Storage Card\Tyrimas\JPG\5.JPG,1315357,70,2009-12-08
02:19:36,61,0.00,47,0.00,2,19,36,3,31,18,14,0.00,4302,70
2,3DES,2,\Storage Card\Tyrimas\JPG\4.JPG,939772,70,2009-12-08
03:31:18,47,0.00,38,0.00,3,31,18,4,22,8,9,0.00,3050,70
2,3DES,2,\Storage Card\Tyrimas\JPG\3.JPG,633160,70,2009-12-08
04:22:08,38,0.00,32,0.00,4,22,8,4,55,59,6,0.00,2031,70
2,3DES,2,\Storage Card\Tyrimas\JPG\1.JPG,233619,70,2009-12-08
04:55:59,32,0.00,29,0.00,4,55,59,5,8,37,3,0.00,758,70
2,3DES,2,\Storage Card\Tyrimas\JPG\2.jpg,460029,70,2009-12-08
05:08:37,29,0.00,25,0.00,5,8,37,5,33,15,4,0.00,1478,70
```

21 pav. Tekstinio failo pavyzdys su tyrimo rezultatų duomenimis.

Programoje yra sukurta standartinė forma `public partial class frmCrypto : Form.` Šioje formoje yra aprašyti naudojami kintamieji bei metodai, kurie vykdo kriptografinius algoritmus, išjungia/įjungia vaizdo sistemą, nuskaito pradinius duomenis, skaičiuoja skirtumus ir įrašo galutinius duomenis į rezultatų failus. Kintamųjų, naudojamų informacinėje sistemoje, sąrašas pateiktas 22 paveiksle. Ne visi kintamieji yra įrašomi į rezultatų failus. Dalis kintamųjų yra naudojami kaip tarpiniai: kiekvienam metodui sugeneruoti raktai, failų skaičius aplanke ir kt.

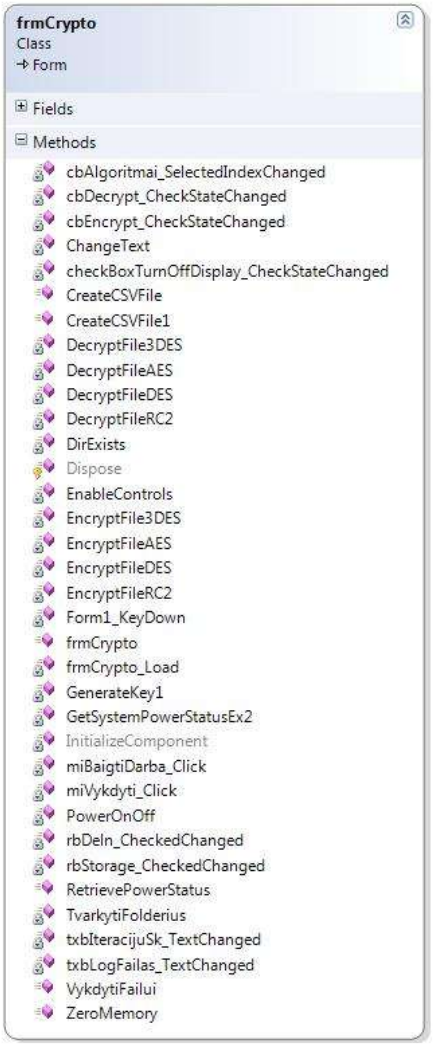
Paveiksle Nr. 23 yra pavaizduoti metodai, naudojami programoje. Kaip ir kintamieji, ne visi metodai yra naudojami pagrindiniam funkcionalumui atlikti. Dalis jų yra pagalbiniai. Pagrindiniai metodai naudojami užšifravimui „EncryptFile3DES“, „EncryptFileDES“, „EncryptFileAES“, „EncryptFileRC2“. Kaip parametrai metodams yra paduodami failų pavadinimai prieš ir po užšifravimo, bei sugeneruoti slapti raktai. Pagrindiniai metodai naudojami iššifravimui „DecryptFile3DES“, „DecryptFileDES“, „DecryptFileAES“, „DecryptFileRC2“. Ir šiems metodams kaip parametrai yra paduodami failų pavadinimai prieš ir po iššifravimo, bei sugeneruoti slapti raktai.



22 pav. Naudojamų kintamųjų sąrašas

Vaizdo sistemos išjungimui/įjungimui yra naudojamas metodas „PowerOnOff“. Dar vienas iš pagrindinių naudojamų metodų yra RetrievePowerStatus. Jo pagalba iš sistemos yra gaunami tyrimui reikalingi delninio kompiuterio energijos parametrai - akumuliatoriaus įtampa (V) bei akumuliatoriaus energijos likutis (%). Dar vienas labai svarbus metodas - „CreateCSVFile“, kuris tyrimo duomenis įrašo į tekstinį CSV tipo failą.

Kviečiant metodą „GenerateKey1“ yra sugeneruojami slapti raktai. Raktai reikalingi failų užšifravimo/iššifravimo procesui vykdyti. Metodas „TvarkytiFolderius“ nuskaito informacija apie Tyrimas kataloge esančius katalogus, bei juose esančius failus, su kuriais bus atliekamas tyrimas. Priklausomai nuo pasirinktų nustatymų t.y. nuo to kur, yra įkelti failai tyrimui surenka informaciją arba apie delninio kompiuterio atmintyje, arba išorinėje atmintyje (storage card) esančius „Tyrimo“ katalogus ir failus. Kaip ir metodas „TvarkytiFolderius“, metodas „VykdytiFailui“ pagal pasirinktus nustatymus (algoritmą, kryptį) atitinkamai iškviečia reikiamus užšifravimo/iššifravimo algoritmus.



23 pav. Naudojamų metodų sąrašas

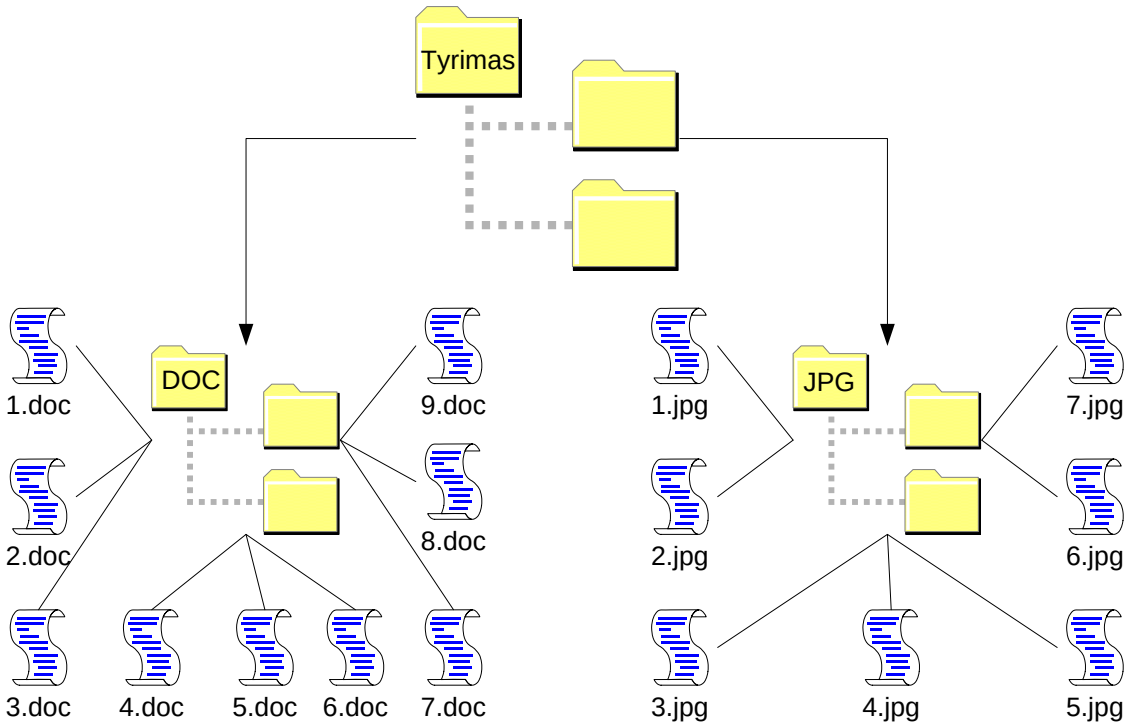
Šiam metodui kaip parametras yra perduodamas failo pavadinimas, kurį jis apdoroja. O visą užšifravimo/iššifravimo funkcionalumą apimantis bei iškviečiantis metodas yra „miVykdyti_Click“. Jame realizuotas visas užšifravimo/iššifravimo funkcionalumas: įrašomi pradiniai ir galutiniai tyrimo rezultatai, patikrinami akumuliatoriaus parametrai, priklausomai nuo situacijos iškviečiami pranešimai, nuskaitomi vartotojo pasirinkti nustatymai, apdorojamas iteracijų pasirinkimas, iškviečiami pagrindiniai metodai šifravimui/iššifravimui bei kiti metodai, reikalingi funkcionalumui vykdyti.

4. EKSPERIMENTINIS DELNINIO KOMPIUTERIO ENERGIJOS SUVARTOJIMO TYRIMAS

Eksperimentas vykdomas su delniniu kompiuteriu (PDA), kurio modelis ASUS P750 turintis tokius parametrus (Pocket PC platform, Intel PXA270 520 MHz CPU, 256 MB RAM, Windows Mobile © 6 Professional OS).

4.1. Tyrimo metodika

Eksperimente naudojami failai įkelti į aplanką Tyrimas. Tyrime naudojami dviejų rūšių failai (tekstiniai-doc ir grafiniai-jpg) atitinkamai įkelti į katalogus DOC ir JPG. Katalogų bei failų struktūra pavaizduota 24 paveiksle. Failų atskyrimas pagal tipą į aplankus žymiai palengvins rezultatų apdorojimą.



24 Pav. Katalogų ir failų struktūra

Siekiant ištirti ar duomenų failo dydis turi įtakos energijos suvartojimui, failai eksperimentui parinkti skirtingų dydžių. Taip pat, siekiant išsiaiškinti ar duomenų tipas (tekstiniai-doc, grafiniai-jpg) turi įtakos energijos suvartojimui, tyrimui parinkti panašaus dydžio failai: tiek tekstiniai, tiek grafiniai. Parinkti failų dydžiai pateikti lentelėje Nr. 10.

Delninio kompiuterio akumuliatoriaus užtenka apie 7 valandas (420 min.). Norint tinkamai atlikti tyrimą, būtina pradžioje teisingai parinkti iteracijų skaičių tyrimui (jei bus parinktas neteisingas iteracijų skaičius, delninio kompiuterio akumuliatoriaus energijos gali

neužtekti įvykdyti vieno ar kito algoritmo bei užfiksuoti rezultatų). Tuo tikslu buvo įvykdyti pasiruošiamieji šifravimo/iššifravimo algoritmo bandymai, pasirinkus 1-10 iteracijų.

Lentelė Nr. 10. Tyrimui naudotų failų dydžių sąrašas

Eil. Nr.	Failo pavadinimas	Failo dydis (KB)	Failo pavadinimas	Failo dydis (KB)
1.	9.doc	1.825	7.jpg	1.871
2.	8.doc	1.593	6.jpg	1.664
3.	7.doc	1.055	5.jpg	1.285
4.	6.doc	803	4.jpg	918
5.	5.doc	593	3.jpg	619
6.	4.doc	421	2.jpg	450
7.	3.doc	340	1.jpg	229
8.	2.doc	286		
9.	1.doc	124		

Empiriniu - praktiniu būdu, pagal gautus pirminius duomenis buvo parinktas iteracijų kiekis kiekvienam algoritmui bei algoritmo krypčiai. Parinktas iteracijų skaičius pateiktas lentelėje Nr. 11.

Lentelė Nr. 11. Eksperimentui parinktų iteracijų sąrašas

	Tekstiniai-DOC		Grafiniai-JPG	
	Užšifravimas	Iššifravimas	Užšifravimas	Iššifravimas
DES	1500	80	1500	80
3DES	1000	70	1000	70
AES	200	140	200	140
RC2	100	80	100	80

Siekiant, kad tyrimo rezultatus būtų paprasta ir greita apdoroti, sudaryta tyrimo rezultatų failų vardų sistema. Tyrimo rezultatų failų vardas sudaromas pagal algoritmą pateiktą lentelėje Nr. 12.

Tyrimas pradedamas su pilnai įkrautu akumuliatoriumi. Prieš atliekant kiekvieną tyrimo dalį užšifravimo/iššifravimo veiksmą kiekvienam kriptografiniam algoritmui, akumuliatorius yra pilnai įkraunamas. Pirmame etape atliksime eksperimentą su tekstiniais-doc tipo failais. Į aplanką //Tyrimas/DOC yra įkeliami tyrimui paruošti tekstiniai-doc tipo failai. Nustatomas iteracijų skaičius, nurodoma vieta, kur saugomi pradiniai ir tarpiniai tyrimo duomenys, bei parenkamas rezultatų failo pavadinimas. Parenkamas kriptografinis metodas.

Lentelė Nr. 12. Tyrimo failo pavadinimų sistema

AAABBBCCCCDDDDE.txt		
AAA	JPG DOC	Failo, kuriam vykdom algoritmą tipas
BBB	Off On	Algoritmas vykdomas išjungus/įjungus vaizdo sistemą
CCCC	DES 3DES AES RC2	Vykdomas algoritmas
DDDD	1000	Iteracijų skaičius
E	D E	Algoritmo kryptis (užšifravimas/iššifravimas)

Pirmiausia parenkama algoritmo kryptis užšifravimas. Prieš paleidžiant kriptografinio algoritmo vykdymą yra parenkama savybė, kuri išjungia vaizdo sistemą, bei paleidžiamas algoritmo vykdymas. Prasisukus procesui programa įrašo duomenis į rezultatų failą, įjungia vaizdo sistemą, bei pateikia tekstinį pranešimą, kad algoritmas įvykdytas. Kadangi tyrime naudojami algoritmai yra simetriniai, tai įvykdžius užšifravimo algoritmą iš karto reikia vykdyti ir iššifravimo algoritmą, kad užšifravimui ir iššifravimui būtų naudojamas ta pats raktas. Nustatomas reikiamas iteracijų skaičius, bei paleidžiamas iššifravimo algoritmas.

Procesas įvykdomas su visais kriptografiniais algoritmais (DES, 3DES, AES, RC2), nustatant reikiamą iteracijų skaičių bei įrašant tyrimo duomenis į atitinkamus tyrimo rezultatų failus. Kiekvieno proceso pabaigoje yra palyginami pradiniai nešifruoti ir iššifruoti duomenys (tarpiniai užšifravimo ir iššifravimo rezultatai). Failų turinys sutampa, tai reiškia, kad užšifravimo ir iššifravimo algoritmas įvykdytas sėkmingai.

Antrame etape eksperimentas atliekamas su grafiniais-jpg tipo failais. Tyrimui paruošti grafiniai-jpg failai įkeliami į //Tyrimas/JPG. Kaip ir tyrimo etape su tekstiniais-doc tipo failais yra parenkami pradiniai nustatymai, įvykdomi kriptografiniai algoritmai, eksperimento rezultatai surašomi į atitinkamus rezultatų failus, bei patikrinama ar pradiniai ir failai po apdorojimo sutampa. Tyrimo rezultatų failai yra perkeliami į stacionarų kompiuterį apdorojimui.

4.2. Tyrimo rezultatai

4.2.1. Skaitiniai tyrimo rezultatai

Realūs tyrimo duomenys pateikti lentelėse Nr. 13-28.

Lentelė Nr. 13. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
AES DOC	99	0	0	0	0
20	79	5476	356.45	5476	356.45
19	60	4810	311.04	10286	667.48
11	49	3171	206.05	13457	873.54
9	40	2499	156.84	15956	1030.37
6	34	1796	115.72	17752	1146.09
5	29	1287	82.13	19039	1228.22
3	26	1048	66.41	20087	1294.63
4	22	988	55.76	21075	1350.39
1	21	447	24.12	21522	1374.51

Lentelė Nr. 14. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
AES DOC	99	0	0	0	0
16	84	4401	249.51	4401	249.51
14	70	3825	217.72	8226	467.24
10	60	2740	144.24	10966	611.47
8	52	2052	109.79	13018	721.26
5	47	1503	81.01	14521	802.27
4	43	1154	57.49	15675	859.76
3	40	989	46.48	16664	906.24
2	38	817	39.03	17481	945.27
2	36	369	16.88	17850	962.16

Lentelė Nr. 15. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
DES DOC	99	0	0	0	0
18	81	5134	2673.34	5134	2673.34
17	62	4585	2332.76	9719	5006.10
13	51	3312	1545.41	13031	6551.51
9	42	2702	1176.27	15733	7727.78
7	35	2189	867.92	17922	8595.70
6	29	1567	615.97	19489	9211.67
4	25	1334	498.05	20823	9709.72
4	21	1224	418.21	22047	10127.93
2	19	670	180.91	22717	10308.84

Lentelė Nr. 16. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
DES DOC	99	0	0	0	0
21	77	6553	142.58	6553	142.58
20	64	5869	124.41	12422	266.99
13	44	3933	82.42	16355	349.41
9	35	2906	62.73	19261	412.15
6	29	2148	46.29	21409	458.44
5	24	1503	32.85	22912	491.29
4	20	1271	26.56	24183	517.85
4	16	1081	22.30	25264	540.16
1	15	281	6.15	25545	546.31

Lentelė Nr. 17. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
3DES DOC	97	0	0	0	0
18	79	4708	1782.23	4708	1782.23
17	62	4422	1555.18	9130	3337.40
10	52	2750	1030.27	11880	4367.68
9	43	2412	784.18	14292	5151.86
6	37	1708	578.61	16000	5730.47
4	33	1319	410.64	17319	6141.11
4	29	1132	332.03	18451	6473.14
4	25	1243	278.81	19694	6751.95
2	23	646	120.61	20340	6872.56

Lentelė Nr. 18. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
3DES DOC	97	0	0	0	0
19	81	5673	124.76	5673	124.76
17	64	4965	108.86	10638	233.62
11	53	3263	72.12	13901	305.74
8	45	2547	54.89	16448	360.63
5	40	1836	40.50	18284	401.13
4	36	1257	28.75	19541	429.88
4	32	1087	23.24	20628	453.12
3	29	921	19.52	21549	472.64
1	28	394	8.44	21943	481.08

Lentelė Nr. 19. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
RC2 DOC	100	0	0	0	0
22	78	6794	178.22	6794	178.22
19	59	6155	155.52	12949	333.74
13	46	4077	103.03	17026	436.77
9	37	3041	78.42	20067	515.19
7	30	2293	57.86	22360	573.05
5	25	1599	41.06	23959	614.11
4	21	1332	33.20	25291	647.31
3	18	972	27.88	26263	675.20
2	17	432	12.06	26695	687.26

Lentelė Nr. 20. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
RC2 DOC	100	0	0	0	0
22	78	6701	142.58	6701	142.58
20	58	5902	124.41	12603	266.99
13	45	3968	82.42	16571	349.41
9	36	2928	62.73	19499	412.15
7	29	2170	46.29	21669	458.44
4	25	1465	32.85	23134	491.29
4	21	1180	26.56	24314	517.85
3	18	1054	22.30	25368	540.16
2	16	439	9.65	25807	549.80

Lentelė Nr. 21. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
AES JPG	99	0	0	0	0
21	78	5732	365.36	5732	365.36
18	60	5046	324.94	10778	690.30
15	45	4020	250.88	14798	941.18
10	35	2966	179.25	17764	1120.43
7	28	2039	120.77	19803	1241.20
3	25	821	44.56	20624	1285.75
5	20	8037	87.74	28661	1373.50

Lentelė Nr. 22. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
AES JPG	99	0	0	0	0
17	79	4524	255.75	4524	255.75
15	64	4044	227.45	8568	483.21
12	52	3133	175.62	11701	658.83
8	44	2303	125.47	14004	784.30
5	39	1585	84.54	15589	868.84
2	37	706	31.19	16295	900.03
4	33	1227	61.42	17522	961.45

Lentelė Nr. 23. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
DES JPG	98	0	0	0	0
20	78	5267	2740.22	5267	2740.22
18	61	4668	2437.02	9935	5177.23
14	46	3865	1881.63	13800	7058.87
10	36	2975	1344.35	16775	8403.22
8	28	2263	905.74	19038	9308.96
3	25	972	334.19	20010	9643.16
6	19	1668	658.08	21678	10301.24

Lentelė Nr. 24. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
DES JPG	98	0	0	0	0
20	80	6434	146.15	6434	146.15
19	61	5516	129.97	11950	276.12
14	47	4345	100.35	16295	376.47
10	37	3106	71.70	19401	448.17
6	31	2087	48.31	21488	496.48
3	28	799	17.82	22287	514.30
5	23	1572	35.10	23859	549.40

Lentelė Nr. 25. Tyrimo duomenys užšifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
3DES JPG	98	0	0	0	0
19	79	5290	1826.81	5290	1826.81
18	61	4588	1624.68	9878	3451.49
13	48	3503	1254.42	13381	4705.91
9	39	2673	896.24	16054	5602.15
6	33	1770	603.83	17824	6205.98
3	30	887	222.80	18711	6428.77
5	25	1388	438.72	20099	6867.49

Lentelė Nr. 26. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
3DES JPG	98	0	0	0	0
20	80	6209	127.88	6209	127.88
19	61	5532	113.73	11741	241.60
14	47	4302	87.81	16043	329.41
9	38	3050	62.74	19093	392.15
6	32	2031	42.27	21124	434.42
3	29	758	15.60	21882	450.01
4	25	1478	30.71	23360	480.72

Lentelė Nr. 27. Tyrimo duomenys užšifravimas

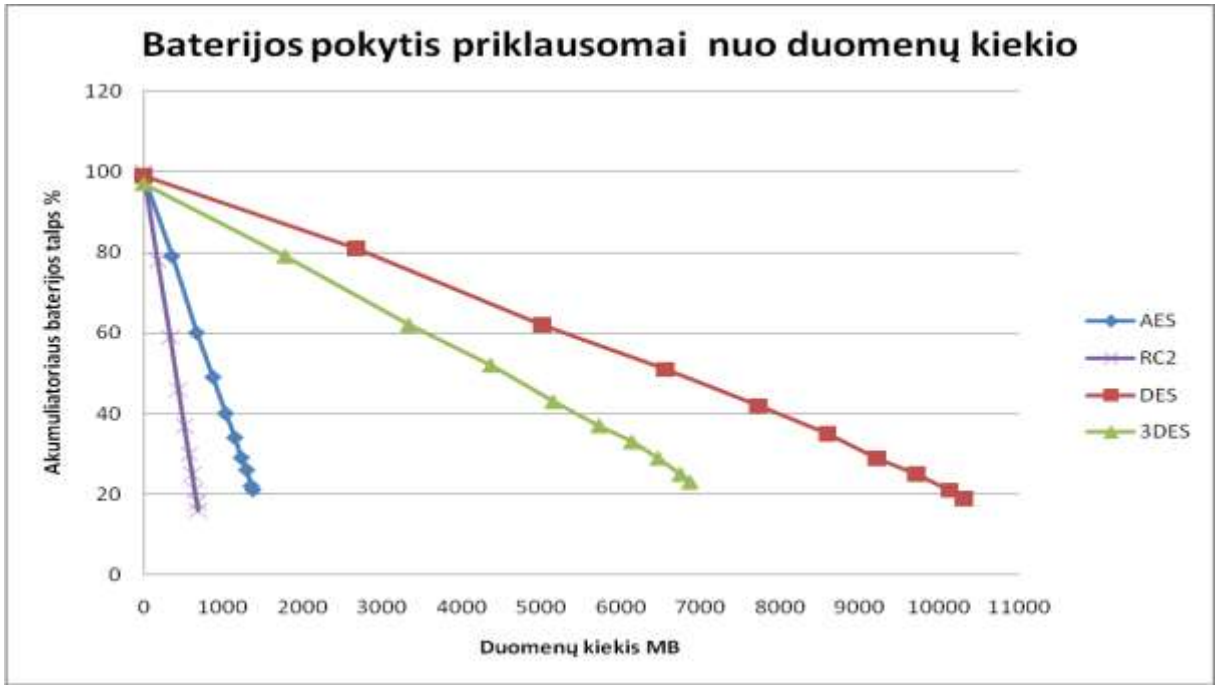
Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
RC2 JPG	99	0	0	0	0
21	78	6806	182.68	6806	182.68
20	58	6213	162.47	13019	345.15
15	43	4732	125.44	17751	470.59
10	33	3358	89.62	21109	560.21
8	25	2346	60.38	23455	620.60
2	23	869	22.28	24324	642.88
5	18	1756	43.87	26080	686.75

Lentelė Nr. 28. Tyrimo duomenys iššifravimas

Failo akum. pokytis, %	Akum. likutis viso, %	Failo apdorojimo laikas, s	Apdorotas duomenų kiekis failui, MB	Apdorojimo laikas viso, s	Duomenų kiekis viso, MB
RC2 JPG	99	0	0	0	0
21	78	7018	146.15	7018	146.15
20	58	6275	129.97	13293	276.12
16	42	4818	100.35	18111	376.47
11	31	3486	71.70	21597	448.17
7	24	2309	48.31	23906	496.48
2	22	871	17.82	24777	514.30
6	16	1685	35.10	26462	549.40

4.2.2. Grafiniai tyrimo rezultatai

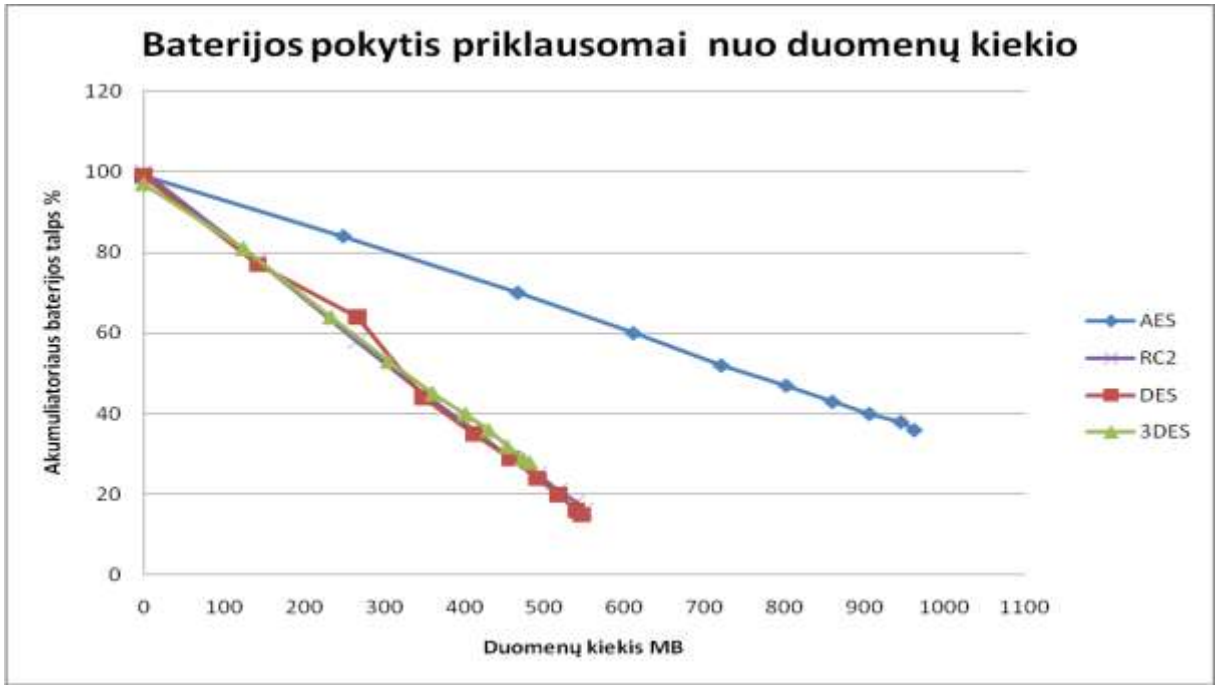
Iš gautų rezultatų matosi, kad užšifruojant tekstinius-doc tipo failus daugiausia duomenų apdorojama naudojant DES ir 3DES algoritmus. Mažiausiai duomenų apdorojama panaudojant RC2 algoritmą. t.y. užšifruojant duomenis panaudojant algoritmus RC2 bei AES bus sunaudojama daugiau energijos nei užšifruojant DES bei 3DES. Grafinis vaizdas pateiktas 25 paveiksle.



25 pav. Duomenys gauti užšifruojant tekstinius-doc tipo duomenis

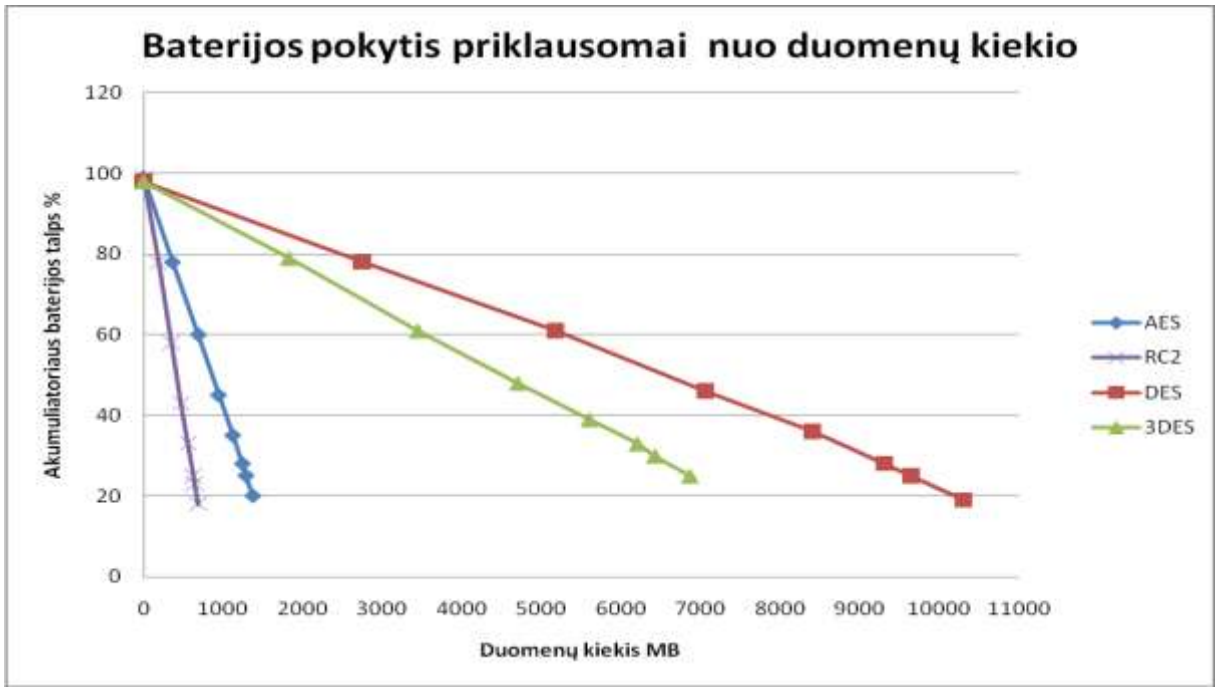
Pagal iššifravimo rezultatus tekstiniais-doc tipo duomenims daugiausia duomenų apdorojama naudojant AES algoritmą. Naudojant kitus algoritmus tokius kaip RC2, DES bei 3DES duomenų apdorojimas nors ir labai panašus, bet yra žymiai mažesnis nei AES. Vadinasi, iššifruojant visais kitais algoritmais nei AES, energijos bus sunaudota daugiau, nei AES algoritmui.

Lyginant su užšifravimo rezultatais panaudojant AES ir RC2 algoritmus, kur apdoroti duomenų kiekiai tuo pačiu ir energijos sąnaudos yra labai panašios, tai apdorotas duomenų kiekis taikant DES ir 3DES kriptografinius metodus skiriasi kelis kartus, tuo pačiu didinat energijos sąnaudas iššifravimui. Grafinis vaizdas pateiktas 26 paveiksle.



26 pav. Duomenys gauti iššifruojant tekstinius-doc tipo duomenis

Analogiškai kaip ir tekstiniams-doc tipo duomenims, užšifruojant grafinius-jpg tipo failus daugiausia duomenų apdorojama naudojant DES ir 3DES algoritmus.

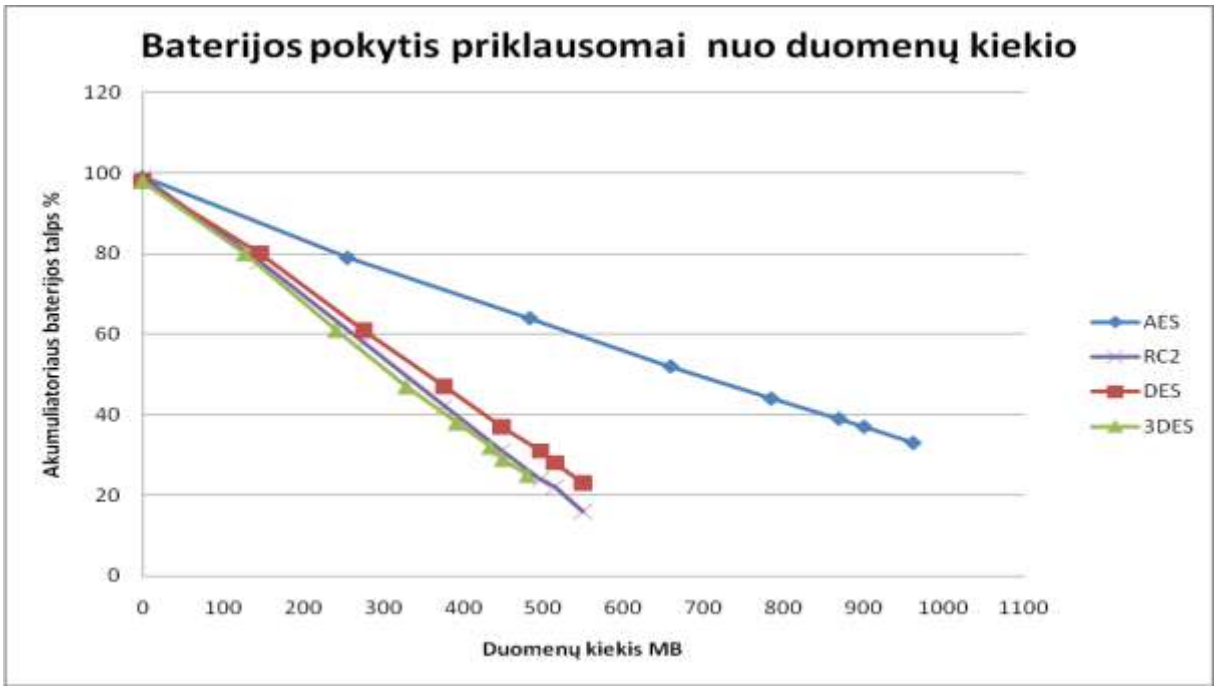


27 pav. Duomenys gauti užšifruojant grafinius-jpg tipo duomenis

Mažiausiai duomenų apdorojama panaudojant RC2 algoritmą. t.y. užšifruojant duomenis panaudojant algoritmus RC2 bei AES bus sunaudojama daugiau energijos nei užšifruojant DES bei 3DES. Grafinis vaizdas pateiktas 27 paveiksle.

Analogiška situacija ir su grafiniais-jpg tipo duomenimis. Pagal iššifravimo rezultatus daugiausiai tekstinių-doc duomenų apdorojama naudojant AES algoritmą. Naudojant kitus algoritmus, tokius kaip RC2, DES bei 3DES, duomenų apdorojimas nors ir labai panašus, bet yra žymiai mažesnis nei AES. Vadinasi, iššifruojant visais kitais algoritmais nei AES, energijos bus sunaudota daugiau, nei AES algoritmui.

Lyginant su šifravimo rezultatais panaudojant AES ir RC2 algoritmus, kur apdoroti duomenų kiekiai tuo pačiu ir energijos sąnaudos yra labai panašios, tai apdorotas duomenų kiekis taikant DES ir 3DES kriptografinius metodus skiriasi kelis kartus, tuo pačiu didinat energijos sąnaudas iššifravimui. Grafinis vaizdas pateiktas 28 paveiksle.



28 pav. Duomenys gauti iššifruojant grafinius-jpg tipo duomenis

Vykdant duomenų užšifravimą taikant AES kriptografinį algoritmą, užšifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafinis vaizdas pateiktas 36 paveiksle.

Vykdant duomenų iššifravimą taikant AES kriptografinį algoritmą, iššifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafiškai rezultatai pateikti 37 paveiksle.

Kaip ir AES kriptografinio algoritmo atveju, vykdant duomenų užšifravimą taikant DES kriptografinį algoritmą, užšifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafiškai rezultatai pateikti 38 paveiksle.

Vykdant duomenų iššifravimą taikant DES kriptografinį algoritmą, iššifruotų duomenų kiekis nuo failo tipo, palyginus tarp visų algoritmų, skiriasi daugiausia (pradedant nuo 50% akumuliatoriaus energijos likučio, nors tiesinė priklausomybė vis tiek išlieka t.y. abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio). Grafiškai rezultatai pateikti 39 paveiksle.

Kaip ir AES kriptografinio algoritmo atveju, vykdant duomenų užšifravimą taikant 3DES kriptografinį algoritmą, užšifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafiškai rezultatai pateikti 40 paveiksle.

Taip pat kaip ir AES kriptografinio algoritmo atveju, vykdant duomenų iššifravimą taikant 3DES kriptografinį algoritmą, iššifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafiškai rezultatai pateikti 41 paveiksle.

Kaip ir AES kriptografinio algoritmo atveju, vykdant duomenų užšifravimą taikant RC2 kriptografinį algoritmą, užšifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafiškai rezultatai pateikti 42 paveiksle.

Taip pat kaip ir AES kriptografinio algoritmo atveju, vykdant duomenų iššifravimą taikant RC2 kriptografinį algoritmą, iššifruotų duomenų kiekis nuo failo tipo praktiškai nesiskiria. Abiejų tipų failų akumuliatoriaus energijos pokytis yra tiesiškai priklausomas nuo duomenų kiekio. Grafiškai rezultatai pateikti 43 paveiksle.

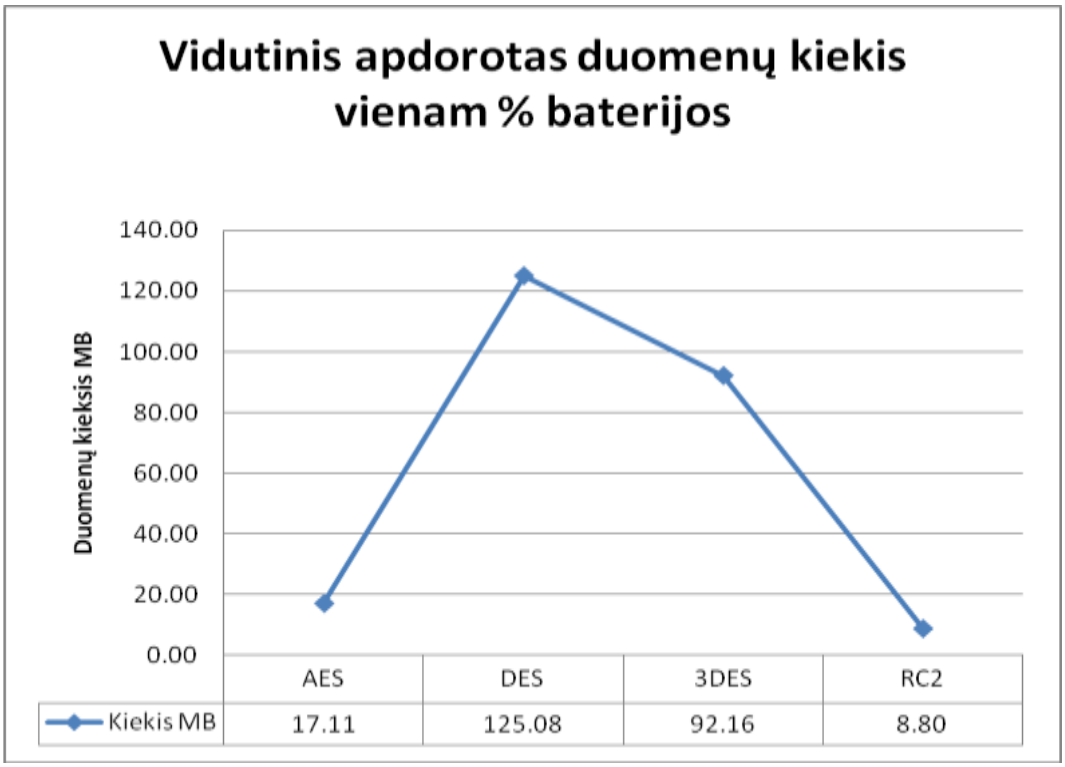
Nepriklausomai nuo duomenų grafinio-jpg failo dydžio, apdorotas duomenų kiekis, užšifruojant pagal algoritmus, išlieka pastovus. Grafiškai rezultatai pavaizduoti 44 paveiksle.

Nepriklausomai nuo duomenų grafinio-jpg failo dydžio, apdorotas duomenų kiekis, iššifruojant pagal algoritmus, išlieka pastovus. Grafiškai rezultatai pavaizduoti 45 paveiksle.

Nepriklausomai nuo duomenų tekstinio-doc failo dydžio, apdorotas duomenų kiekis, užšifruojant pagal algoritmus, išlieka pastovus. Grafiškai rezultatai pavaizduoti 46 paveiksle.

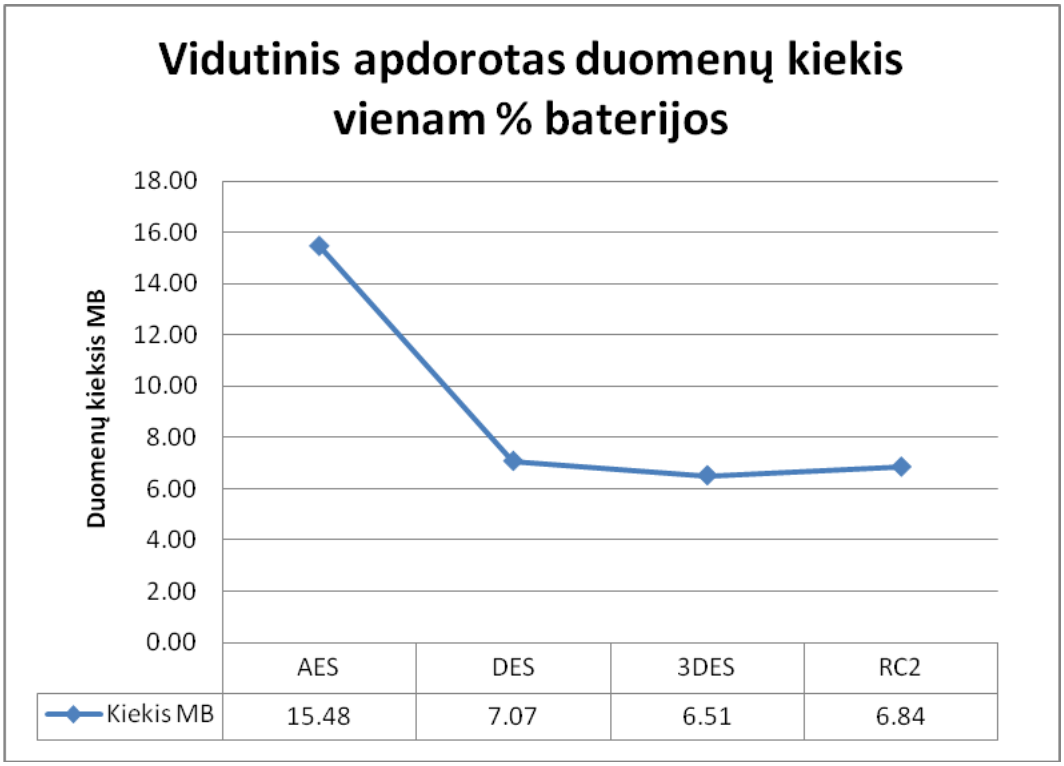
Nepriklausomai nuo duomenų tekstinio-doc failo dydžio, apdorotas duomenų kiekis, iššifruojant pagal algoritmus, išlieka pastovus. Grafiškai rezultatai pavaizduoti 47 paveiksle.

Vidutinis apdorotas duomenų kiekis, užšifruojant grafinius-jpg failus, pavaizduotas 29 paveiksle.



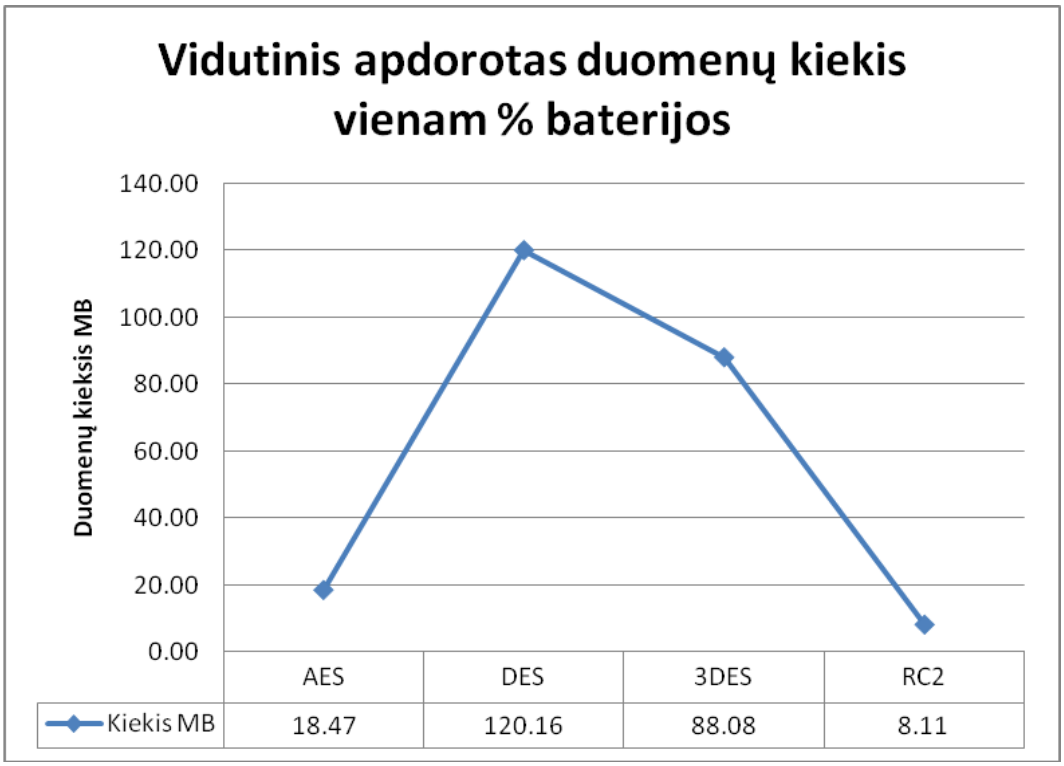
29 pav.. Duomenys gauti užšifruojant grafinius-jpg tipo duomenis

Vidutinis apdorotas duomenų kiekis, iššifruojant grafinius-jpg failus, pavaizduotas 30 paveiksle.



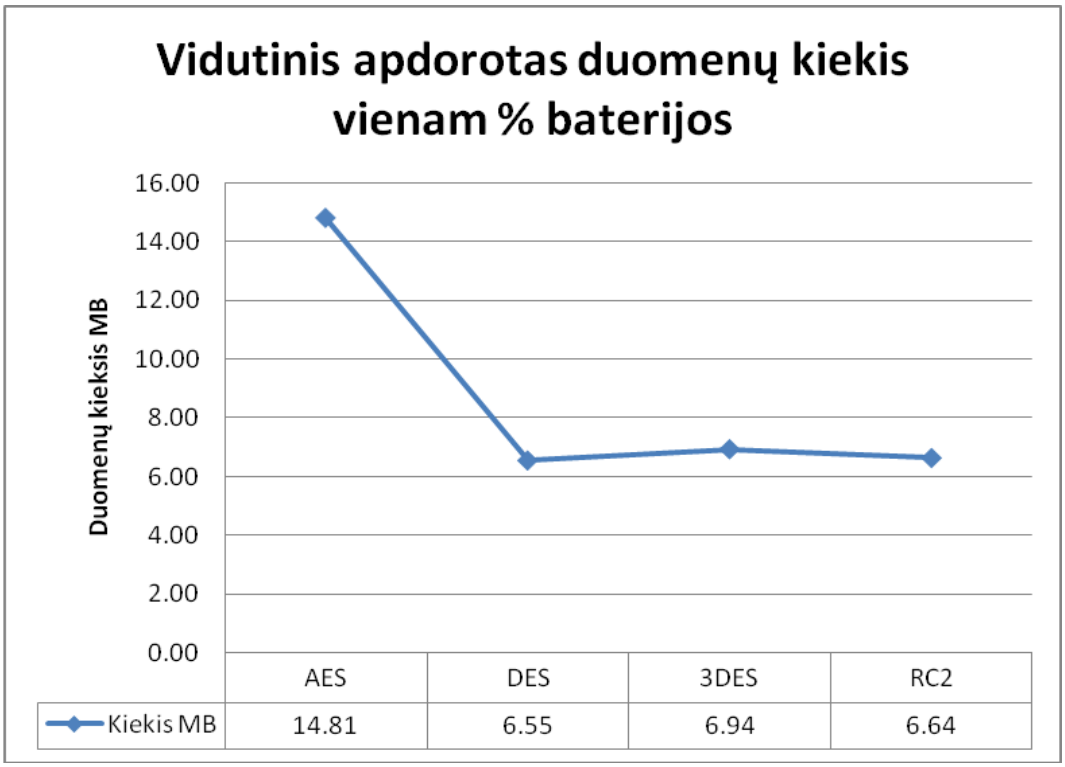
30 pav. Duomenys gauti iššifruojant grafinius-jpg tipo duomenis

Vidutinis apdorotas duomenų kiekis, užšifruojant tekstinius-doc failus, pavaizduotas 31 paveiksle.



31 pav.. Duomenys gauti užšifruojant tekstinius-doc tipo duomenis

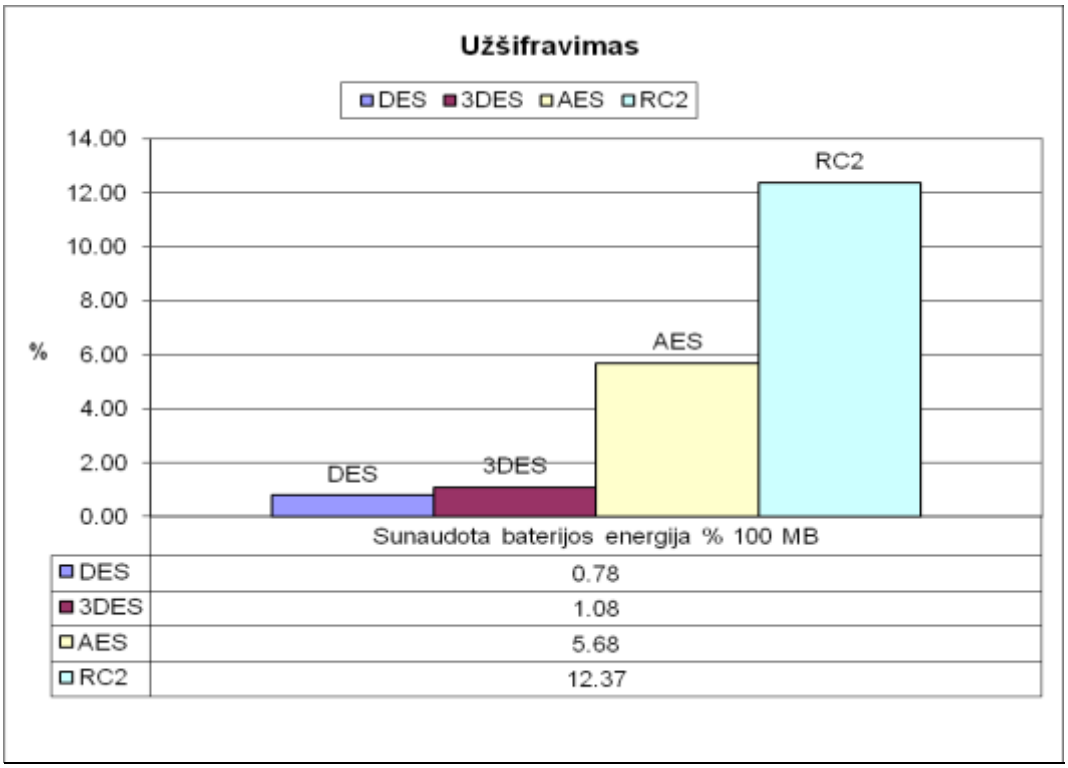
Vidutinis apdorotas duomenų kiekis, iššifruojant tekstinius-doc failus, pavaizduotas 32 paveiksle.



32 pav. Duomenys gauti iššifruojant tekstinius-doc tipo failus

Iš anksčiau pateiktų paveikslų Nr. 29-32, pateikiančių vidutinius apdorotus duomenų kiekius skirtingiems duomenų tipams bei algoritmo kryptims, matosi, kad nepriklausomai nuo duomenų tipo (tekstiniai-doc arba grafiniai-jpg) tiek užšifruojant, tiek iššifruojant energijos sunaudojama labai panašiai.

Siekiant apibendrinti tyrimą bei pateikti dar detalesnius ir aiškiau grafiškai matomus tyrimo rezultatus, buvo suskaičiuotos energijos sąnaudos 100 MB duomenų, pavaizduojant užšifravimo ir iššifravimo rezultatus skirtinguose paveiksluose. Kaip jau minėjau anksčiau, skirtumai tarp duomenų tipo (tekstiniai-doc arba grafiniai-jpg) nežymūs, todėl pateikiami apibendrinti duomenys. Užšifravimo duomenys 100 MB duomenų pateikti 33 paveiksle.

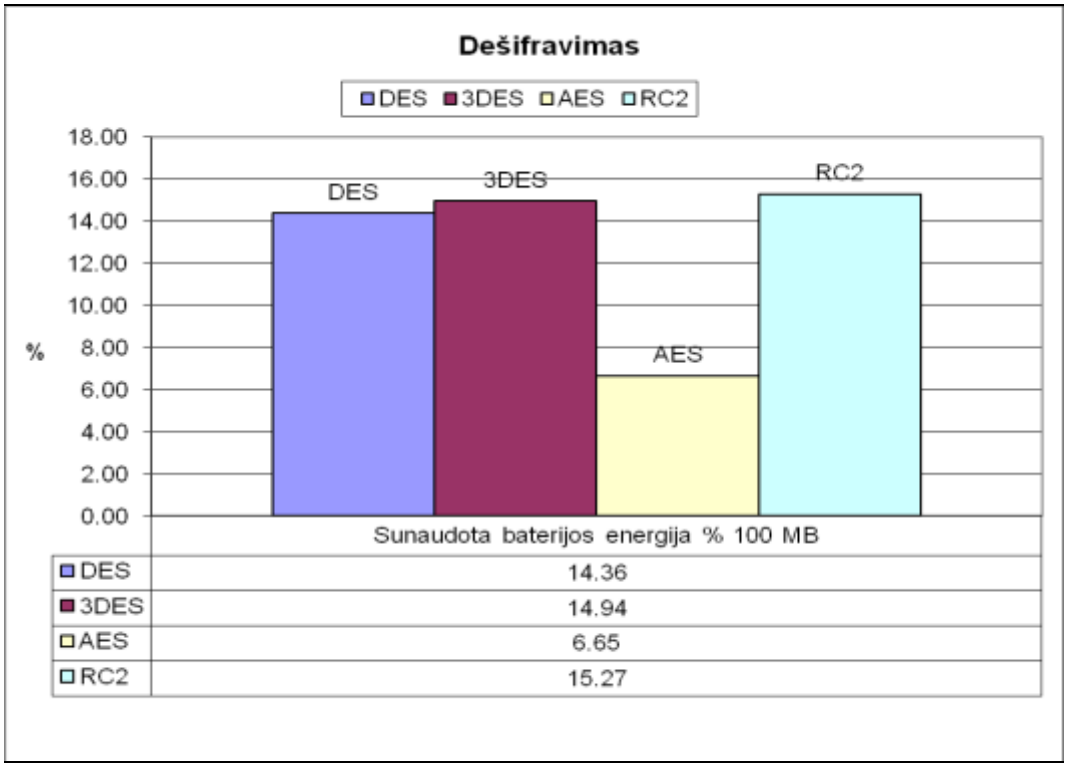


33 pav. Akumuliatoriaus energijos sąnaudos % 100MB duomenų

Lyginat atskirai kiekvieno algoritmo (DES, 3DES, AES, RC2) vykdymo kryptį (užšifravimas/iššifravimas) pagal tyrimo duomenis galima teigti, kad užšifruojant ir iššifruojant duomenis skirtingais algoritmais, energijos sąnaudos vykdant vieną algoritmą, ne visiems algoritmams naudoja panašų energijos kiekį.

Algoritmui AES energijos sąnaudos yra panašios (užšifravimas - 5,68% / iššifravimas - 6,65%) taip pat ir algoritmui RC2 energijos sąnaudos yra panašios (užšifravimas - 12,37% / iššifravimas -15,27%). Tačiau algoritmams DES bei 3DES energijos sąnaudos užšifruojant lyginant su energijos sąnaudomis iššifruojant skiriasi keturiolika kartų ir yra atitinkamai tokios: algoritmas DES (užšifravimas - 0,78% / iššifravimas -14,36%), algoritmas 3DES

(užšifravimas - 1,08% / iššifravimas -15.27%). Užšifravimo duomenys 100 MB duomenų pateikti 34 paveiksle.



34 pav. Akumuliatoriaus energijos sąnaudos % 100MB duomenų

Lyginat visus algoritmus (DES, 3DES, AES, RC2) pagal vykdymo kryptį (užšifravimas/iššifravimas) iš rezultatų matosi, kad tam pačiam duomenų kiekiui (100 MB) apdoroti energijos sąnaudos skirtingiems algoritmams yra skirtingos. Užšifruojant algoritmas RC2 (10,92%) naudoja dvigubai daugiau energijos nei algoritmas AES (5,68%). Tuo tarpu algoritmai DES (0,78%) ir 3DES (1,08%) naudoja 10-12 kartų mažiau energijos nei RC2. Kitaip nei užšifruojant, iššifruojant duomenis, taikant DES, 3DES, RC2 algoritmus energijos sąnaudos yra panašios atitinkamai (14,36%, 14,94%, 15,27%). Algoritmas AES (6,65%) naudoja dvigubai mažiau energijos nei algoritmai DES, 3DES, RC2.

5. IŠVADOS

- ✓ Tyrimas patvirtino hipotezę, kad užšifravimo/iššifravimo metu energijos sąnaudos yra panašios skirtingo tipo duomenims (tekstiniai-doc, grafiniai-jpg).
- ✓ Nepriklausomai nuo failo dydžio ir atitinkamo tipo (tekstiniai-doc, grafiniai-jpg), energijos sąnaudos vienam duomenų baitui skiriasi mažiau nei 10% , naudojant tą patį algoritmą (DES, 3DES, AES, RC2).
- ✓ Energijos sąnaudos skiriasi priklausomai nuo algoritmo tipo (DES, 3DES, AES, RC2). Tokiam pat duomenų kiekiui apsaugoti (100 MB) algoritmas RC2 naudoja dvigubai daugiau energijos nei algoritmas AES. Algoritmai DES ir 3DES naudoja 10-12 kartų mažiau energijos nei RC2.
- ✓ Energijos sąnaudos pagal algoritmus skiriasi nuo algoritmo krypties (užšifravimas, iššifravimas). Failų užšifravimas naudojant algoritmus (DES, 3 DES) .NET compact framework platformos bibliotekas yra keturiolika kartų greitesnis nei iššifravimas. Failų užšifravimas ir iššifravimas naudojant (AES, RC2) atitinkamai skiriasi mažiau nei 20%. Rekomenduočiau .NET compact framework platformos kriptografinės bibliotekos algoritmus DES ir 3DES naudoti tik duomenų užšifravimui, juos iššifruojant kituose įrenginiuose, turinčiuose pastovų energijos šaltinį. Tokiu atveju reikia išspręst rakto apsikeitimo bei jo saugumo užtikrinimo uždavinį.
- ✓ Algoritmų DES ir 3DES raktų dydžiai skiriasi 3 kartus, tačiau energijos sąnaudos tiek užšifruojant, tiek iššifruojant skiriasi ne daugiau 30%.
- ✓ Turint tyrimo rezultatus, žinant koks yra informacijos saugumo lygio poreikis, informacijos, kurią reikia apsaugoti kiekį bei esamą delninio kompiuterio akumuliatoriaus energijos būseną (energijos likutį), vartotojas gali protingai/pagrįstai įvertinti kaip tvarkyti duomenis energijos sąnaudų atžvilgiu.
- ✓ Dar geresnis sprendimas yra sukurti programinę įrangą, kuri naudodama tyrimo duomenis pateiks įvertinimą ir patarimus bei pasiūlys geriausią saugos algoritmą, įvertindama esamą energijos būseną, galimas energijos sąnaudas, užtikrinant reikiamą saugumo lygį, priklausomai nuo vartotojo profilio.
- ✓ Magistrinio darbo tematika parašytas mokslinis straipsnis išspausdintas žurnale „ELEKTRONIKA IR ELEKTROTECHNIKA“ 2010 m. Nr. 5 (101) [26] ir perskaitytas pranešimas 14-oje tarptautinėje konferencijoje „ELEKTRONIKA 2010“ vykusioje 2010-05-18 Vilniuje.

6. LITERATŪRA

1. Pokharel M., Park J. Cloud computing: future solution for e-governance // Proc. of the 3rd International Conference on Theory and Practice of Electronic Governance, Bogota, Columbia, November 10-13, 2009, pp. 409-410.
2. Gartner. Gartner News Room, January 18, 2010. http://www.gartner.com/it/page.jsp?id=1282413.
3. Kim H., Smith J., Shin K. Detecting Energy-Greedy Anomalies and Mobile Malware Variants // Proc. of MobiSys'08, Breckenridge, Colorado, USA, June 17-20, 2008, pp. 239-252.
4. Garret M. Powering down // Communication of the ACM. ACM media, New York, September 2009, vol.51, NO 9, pp. 43-46.
5. Microsoft Corporation, .NET Framework Developer Center. http://msdn.microsoft.com/en-us/netframework/default.aspx
6. Venčkauskas, A.; Toldinas, J. Kompiuterių ir operacinių sistemų sauga. Kaunas, 2007. 39p.
7. Anand Raghunathan, Srivaths Ravi, Sunil Hattangady, Jean-Jacques Quisquater // Securing Mobile Appliances: New Challenges for the System Designer ,NEC Laboratories America, Princeton, NJ, USA, Texas Instruments Inc., Dallas, TX, USA, Universite catholique de Louvain, Louvain-la-Neuve, Belgium.
8. Limor Elbaz, Hagai Bar-El. // Strength Assessment Of Encryption Algorithms White Paper, Date 2000, pp 12.
9. Tilborg, Henk C. A van. // Fundamentals of Cryptology : A Professional Reference and Interactive Tutorial, Kluwer Academic Publishers 1999 pp 63-69.
10. Zimmermann, Ph. An Introduction to Cryptography, 1999, Network Associates.
11. Advanced Enryption Standard, 2001, Fedreal Information.
12. Ryabko, B.; Fionov, A. // Basics of Contemporary Cryptography for IT Practitioners, World Scientific Publishing Company, Incorporated, 2005, pp. 148-158.
13. Burnett, Mark //Hacking the Code : ASP. NET Web Application Security, Syngress Publishing, 2004, pp. 165-170.
14. Davis, Carlton // IPSec : Securing VPNs RSA Press, McGraw-Hill Professional Book Group , 2001, pp. 82-83, 100-101.
15. Serpanos, D.N. Giladi, R. // Security and Embedded Systems, IOS Press, 2006, pp. 85-86
16. MSDN library, Microsoft, March 20th, 2010 http://msdn.microsoft.com/en-us/library/ms978415.aspx.

17. Ramnath Venugopalan, Prasanth Ganesan, Mihail Sichitiu, Alexander Dean, Frank Mueller, Pushkin Peddabachagari // Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis, Center for Embedded Systems Research, Depts. of ECE and CS North Carolina State University, Raleigh.
18. Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan, Niraj K. Jha // Analyzing the Energy Consumption of Security Protocols, Dept. of Electrical Engineering, Princeton University, Princeton.
19. Raghunathan A., Ravi S., Hattangady S., Quisquater J-J. Securing Mobile Appliances: New Challenges for the System Designer // Proc. of the 3d IEEE Int. Conf. on Design, Automation and Test in Europe, DATE'03, IEEE 2003.
20. Wikipedia, Net Framework, March 20th, 2010 http://en.wikipedia.org/wiki/.NET_Framework.
21. MSDN library, Microsoft, March 20th, 2010 http://msdn.microsoft.com/lt-lt/library/9s7k7ce5.aspx.
22. Venugopalan R., Ganesan P., Peddabachagari P. Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis // Proc. of CASES'03, San Jose, California, USA, Nov. 1, 2003, pp. 188-197.
23. Toldinas E., Štuikys V., Damaševičius R., Ziberkas G. Application-Level Energy Consumption In Communication Models For Handhelds // Electronics and Electrical Engineering 6(94), 2009, pp. 73-76.
24. Damaševičius R, Štuikys V., Toldinas E. Embedded program specialization for multiple criteria trade-offs // Electronics and Electrical Engineering 8(88), 2008, pp. 9-14.
25. MSDN library, Microsoft, March 20th, 2010 http://msdn.microsoft.com/en-us/library/bb416357.aspx
26. J. Toldinas, V. Štuikys, D. Naunikas. Power awareness experiment for crypto service-based algorithms // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 5(101). – P. 57-62.

7. SUMMARY

As a result of further expansion of Information Communication Technologies (ICT), one can observe the emergency of a new trend in using the technologies now: a global virtualization of the world that is express through the term ‘cloud computing’. Cloud computing is the future generation of computing which is characterized by main entities - Software, Hardware and Network. The collective nature of all these entities, as combined into a coherent system with modern computational features such as mobility, is known as the Cloud. Mobility and mobile computing play a significant role in this context. Because of continued miniaturization, ubiquitous communication, and increasing computation power, mobile handheld (aka Personal Digital Assistant – PDA) users can now perform many online tasks on the go, including web browsing, document editing, multimedia streaming, and Internet banking, to name a few.

Though there are many problems yet to be solved within this new computing paradigm, two major concerns should be mentioned in the first place: energy consumption and information security. The first issue is due to an adequate progress of computational power and energy (battery) power (e.g., processor and computer speed have increased thousands times, while the battery power is a scarce resource for mobile devices). The second issue is associated with the first one. Indeed, now people want to work on the go, where battery is the main energy resource. Because of mobile devices (phones, PDAs, tablet computers, etc.) are as little as possible they simply could be lost or stolen. The information stored in them might be available for the public use. To reduce such a risk we must encrypt information of the devices.

Security mechanisms address computing services, such as authentication for user admission, intrusion detection and prevention as well as counter-measures for other forms of attacks (e.g., denial of service) and data protection in storage, in e-mails or to provide secure transactions.

The aim of this paper is to consider the matching between a family (variants) of the given cryptography algorithms and a given set of prescribed constraints (e.g., performance, energy consumption awareness, safety levels, user profile and various trade-offs amongst the constraints).

The task we consider in the paper is as follows: to identify the energy consumption and performance trade-offs for crypto service that implements four algorithms (DES, 3DES, AES, RC2) within the wide spread technology .NET Compact Framework.

8. PRIEDAI

Visi priedai yra sudėti į kompaktinį diską. Kompaktinis diskas prisegtas 8.4 priede.

8.1. Tarptautinės konferencijos „Elektronika 2010“ dalyvio diplomas

Diplomas pažymintis dalyvavimą 14-oje tarptautinėje konferencijoje „ELEKTRONIKA 2010“ vykusioje 2010-05-18 Vilniuje.

35 pav. Diplomas

8.2. Publikacija „Power Awareness Experiment for Crypto Service-Based Algorithms“

Šiame priede yra pateikiamas mokslinis straipsnis išspausdintas žurnale ELEKTRONIKA IR ELEKTROTECHNIKA. 2010 m. Nr. 5 (101).

ISSN 1392 – 1215 ELECTRONICS AND ELECTRICAL ENGINEERING / ELEKTRONIKA IR ELEKTROTECHNIKA 2010. No. 5(101)

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

Power Awareness Experiment for Crypto Service-Based Algorithms

J. Toldinas
Computer Department, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, e-mail: eugenijus.toldinas@ktu.lt
V. Štuikys, 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
D. Naunikas
Computer Department, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, e-mail: darius.naunikas@stud.ktu.lt

Introduction

As a result of further expansion of Information Communication Technologies (ICT), one can observe the emergence of a new trend in using the technologies now: a global virtualization of the world that is expressed through the term 'cloud computing'. Cloud computing is the future generation of computing which is characterized by main entities - Software, Hardware, Network and ability to use the remote portable resources such as a results of some computations. The collective nature of all these entities, as combined into a coherent system with modern computational features such as mobility, is known as the Cloud [1]. Mobility and mobile computing play a significant role in this context [2]. Because of continued miniaturization, ubiquitous communication, and increasing computation power, mobile handheld (aka Personal Digital Assistant – PDA) users can now perform many online tasks on the go, including web browsing, document editing, multimedia streaming, and Internet banking, to name a few [3].

Though there are many problems yet to be solved within this new computing paradigm, two major concerns should be mentioned in the first place: energy consumption and information security. The first issue is due to an adequate progress of computational power and energy (battery) power (e.g., processor and computer speed have increased thousands times [4], while the battery power is a scarce resource for mobile devices). The second issue is associated with the first one. Indeed, now people want to work on the go, where battery is the main energy resource. Because mobile devices (phones, PDAs, tablet computers, etc.) are as little as possible they simply could be lost or stolen. The information stored in them becomes accessible for the non-authorized use. To reduce such a risk we must encrypt information within the devices.

Security mechanisms address computing services, such as authentication for user admission, intrusion detection and prevention as well as counter-measures for other forms of attacks (e.g., denial of service) and data protection in storage, in e-mails or to provide secure transactions [5].

The aim of this paper is to consider the matching between a family (variants) of the given cryptography algorithms and a given set of prescribed constraints (e.g., performance, energy consumption awareness, safety levels, user profile and various trade-offs amongst the constraints).

The task we consider in the paper is as follows: to identify the energy consumption and performance trade-offs for crypto service that implements four algorithms (DES, 3DES, AES, RC2) within the wide spread technology .NET Compact Framework [6].

Context and general framework to analyze the task

The context of the task is a modern organization (Fig. 1), where a set of battery-dependent mobile devices are connected to the stationary computing resources through wireless communication links. The devices may operate under different operating systems exploited on the go.

Today the technology enables the use of different mobile operating systems as it is depicted in Fig 2. Energy management within operating environments is being performed at multiple layers of the systems: the physical layer, the operating system (OS) layer, and the application layer [7]. Since faster Central Processing Units (CPUs) and larger memories tend to require more power to operate at the same time enabling better

functionality of applications, techniques to reduce and manage energy consumption at the application level are necessary. On the other hand, the application layer should be protected from the malicious interventions of hackers into the systems.

Fig. 1. Modern organization structure based on cloud computing

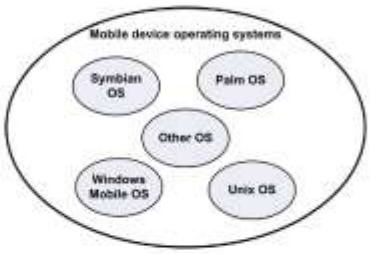


Fig. 2. Mobile device operating systems

Providers of mobile devices and their software try to solve the security issues proposing different approaches such as anti-virus or file encrypt-based. One way for file encryption is the operating system tool, such as bit locker, or the encryption with well-known cryptographic algorithms. The other solution to security proposes a Security Content layer as Anti-virus and File Encrypt facilities.

Microsoft ® proposes a Modern .NET framework technology that has the crypto service provider (Fig.3) with service providers for information encryption/decryption on handheld PC with DES, 3DES, AES, RC2 algorithms [6].

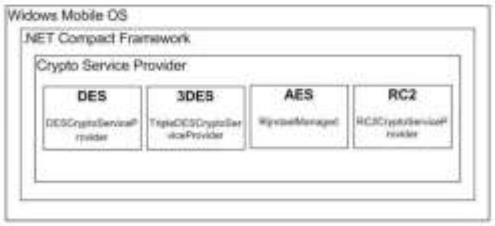


Fig. 3. .NET Compact Framework Crypto Service Provider based on [6]

When we have the crypto service provider with well known algorithms, we can use it for information encryption/decryption. However, the information hiding comes not for free: that requires energy resources, too. As it has been already mentioned, awareness of the energy consumption is highly important in mobile devices. But how we can ensure the needed functionality, the reasonable use of energy and the different information security levels at the same time? What is trade-offs of those contrasting requirements? Empirically we can predict that stronger cryptography algorithms consume more energy. In such a way 3DES must consume more energy than DES, because it repeats the DES cryptography three times. Thus there are many unclear questions we try to give an answer through some experiments we describe in this paper.

In Fig. 4, we present a general scheme that is to be connected with the .NET Compact framework (Fig. 3) to provide our experiments. We have selected two types of information (documents and pictures) to be encrypted and decrypted as the most relevant of applications.

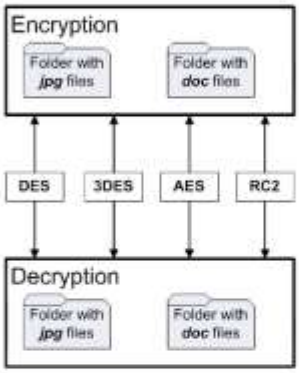


Fig. 4. Cryptography algorithms and file types

Note that here we try to protect information files (e.g., documents, secure personal information, etc.), but not the program files which constitute a separate security problem.

Methodology

The task of the experiments is to identify various dependencies among the energy consumption features, encryption/decryption modes, different cryptography algorithms and different information types. Fig. 5 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 energy consumption values for individual cryptographic algorithms are obtained by running their .NET Compact Framework Crypto Service Provider implementations, and measuring the current battery drawn. For getting valuable results of battery drain when data is encrypted/decrypted, we iterate cryptography process. Because encryption and decryption time may vary we perform encryption and decryption separately.

Before starting cryptography process we identify: the number of iterations (**NI**), the cryptography algorithm (**DES, 3DES, AES, RC2**), and the cryptography direction

(**ED** - encryption or decryption). To ensure a more precise measurement results we eliminate some energy consumption features such as display backlight and graphic card by turning it off before starting cryptography process (Fig. 5). Selected files for encryption/decryption were used separately: documents (**doc** format) and pictures (**jpg** format).

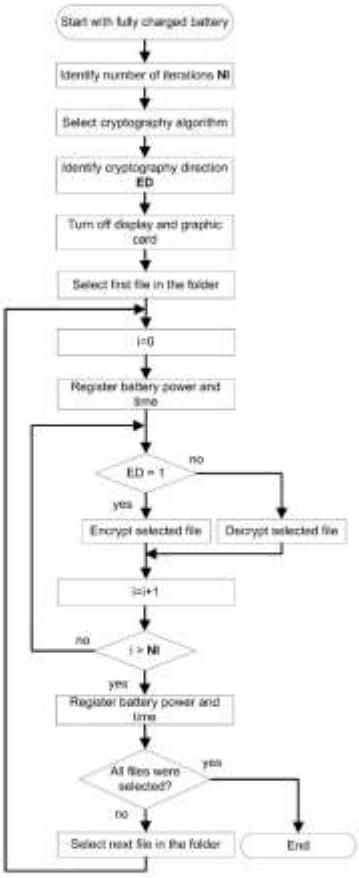


Fig. 5. Energy measurement algorithm for a given crypto algorithm

The algorithm exploits the only one crypto algorithm at a time starting with the fully charged battery (100%). Next what is important to note is that we need to allow discharging the battery until approximately 20%. To achieve this level when operating we need to manage the process as flexibly as possible because there are severe restrictions on memory size and availability to obtain the measured values. For this purpose, we have applied the cryptography for two types of files (document and picture) within created folders DOC and JPG, and placed there the appropriate files of the different length (the type of folder is not specified in Fig. 5). We use own files because benchmark files were not found for our context. The size of the files in folders is limited by the size of random-access memory (RAM). Files placed in the folders were selected to achieve approximately the same size of the DOC folder (6,87MB) and the JPG folder (6,86MB). Files were sorted in descending order by the file size because we did iterations of cryptography algorithm to reach suitable measurement results, and files with bigger size were first encrypted/decrypted through the iterations. The iteration number NI we have identified experimentally (e.g., for DES NI=1500, for 3DES NI=1000, for AES NI=200 and for RC2 NI=100).

Experiments

To realize the experiments we have developed the program that implements the algorithm (Fig. 5) 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. The DOC folder contains nine files sorted in descending order by file size (9.doc-1.825KB, 8.doc-1.593KB, 7.doc-1.055KB, 6.doc-803KB, 5.doc-593KB, 4.doc-421KB, 3.doc-340KB, 2.doc-286KB, 1.doc-124KB). The JPG folder contains seven file sorted in descending order by the file size (7.jpg-1.871KB, 6.jpg-1.664KB, 5.jpg-1.285KB, 4.jpg-918KB, 3.jpg-619KB, 2.jpg-450KB, 1.jpg-229KB).

We provide the summary of the experiment results in Tables 1 (for .doc files) and Table 2 (for .jpg files). Each Table contains the crypto algorithms, the amount of encrypted/decrypted information in MB, elapsed time and total consumed energy in % for that amount of information. For example, in order to encrypt about 10GB using the DES algorithm computer needs about 6 hours of processor's time and consumes about 80% of energy.

Table 1. Summary of the experimental results (document files encryption-decryption)

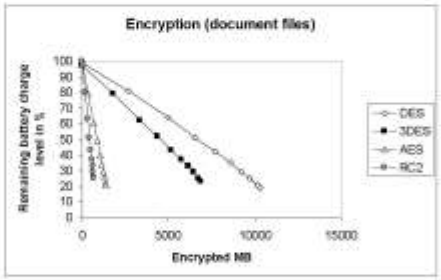
Encryption			
Crypto algorithm	Amount of information MB	Elapsed time hh:mm	Battery power consumed %
DES	10308	06:19	80
3DES	6873	05:39	74
AES	1374	05:58	78
RC2	687	06:32	75
Decryption			
Crypto algorithm	Amount of information MB	Elapsed time hh:mm	Battery power consumed %
DES	550	06:51	79
3DES	482	06:06	72
AES	962	04:58	64
RC2	550	07:10	84

Note. In the second column of Table 1, the amount of information is calculated according to the number of iterations NI (see values at the end of the previous section).

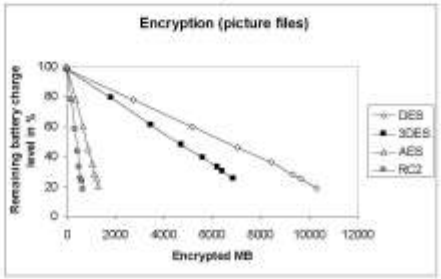
Table 2. Summary of the experimental results (picture files encryption-decryption)

Encryption			
Crypto algorithm	Amount of information MB	Elapsed time hh:mm	Battery power consumed %
DES	10301	06:01	79
3DES	6868	05:35	73
AES	1294	06:08	79
RC2	647	07:15	81
Decryption			
Crypto algorithm	Amount of information MB	Elapsed time hh:mm	Battery power consumed %
DES	549	06:38	77
3DES	482	06:30	75
AES	961	04:52	63
RC2	518	07:21	83

As Tables 1 and 2 provide us with the final measurement points only, we deliver the details of the process in charts. In Fig. 6, we present all measured points for each algorithm when document files (a) and picture files (b) are encrypted. In Fig. 7, we present all measured points for each algorithm when document files (a) and picture files (b) are decrypted. We can see that either encryption or decryption for document files and picture files consume approximately the same amount of energy (if they are about of the same size). Next, the energy consumption either in encryption or decryption mode linearly depends on the file size.

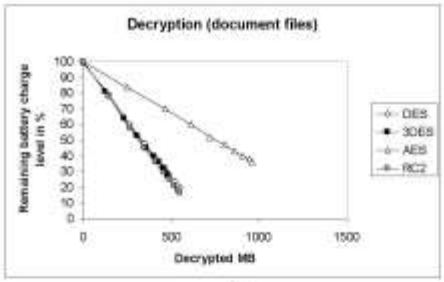


a)

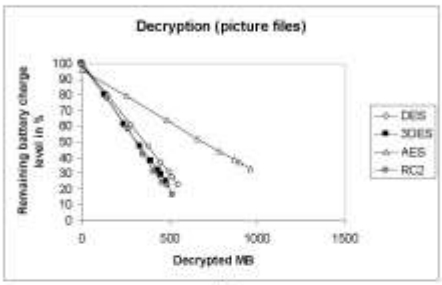


b)

Fig. 6. Energy consumption when encrypted document files (a) and picture files (b)



a)



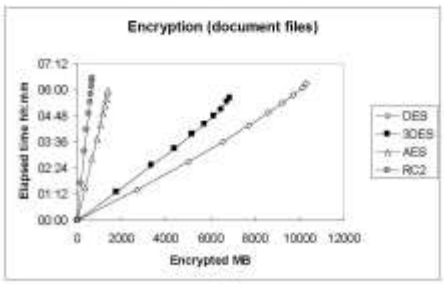
b)

Fig. 7. Energy consumption when decrypted document files (a) and picture files (b)

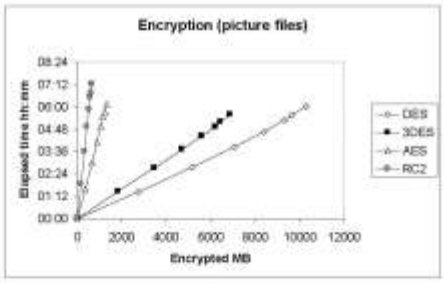
In Fig. 8 and 9, we deliver time-information amount dependencies for the same encryption/decryption algorithms and files used. Again, there is a linear dependency among those factors.

Now let us compare the behavior of encryption and decryption algorithms with respect to energy consumption. As we can see (cp. Fig. 6 and Fig. 7) that decryption requires much more battery energy and time resources than encryption for algorithms DES and 3DES. However, the remaining algorithms behave in other manner.

We explain those discrepancies in detail in Fig. 10. For getting more visible results, we calculate battery energy consumption for the 100MB of encrypted/decrypted information by each type of crypto algorithms (Fig. 10).

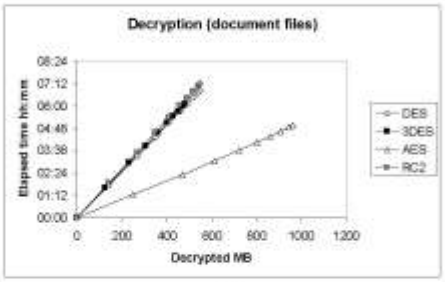


a)

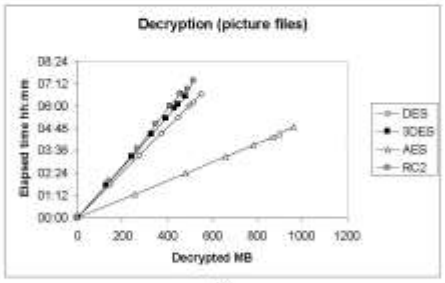


b)

Fig. 8. Time consumed when encrypted document files (a) and picture files (b)

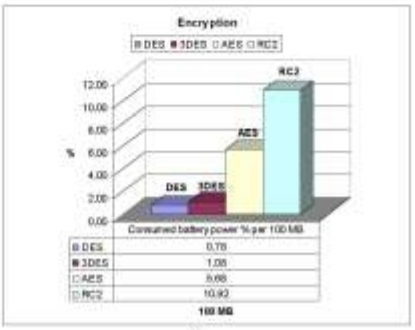


a)



b)

Fig. 9. Time consumed when decrypted document files (a) and picture files (b)



a)

b)

Fig. 10. Battery energy consumption by crypto algorithms for encryption (a) and decryption (b) of 100 MB of information

Note. also that prior of using an algorithm a length of the encryption key is to be selected first and then generated its value. In all experiments we depicted in Charts and Tables the key length was as follows: 4 blocks x 16 bytes for DES and 128 bytes for AES and RC2.

Discussion, evaluation and conclusions

We have selected for our energy-based experiments the Microsoft .NET Compact Framework as a modern and widely used platform for the safe mobile program development and secure information management. Though there are a wide variety of encryption/decryption algorithms we were restricted with the four algorithms used within the Framework. The behaviour of different encryption algorithms with respect to energy consumption is highly different: for the same amount of information (100 MB), e.g., RC2 requires approximately twice more energy than AES and about twelve times more than 3DES. Encryption and decryption modes of AES and RC2 require approximately the same amount of energy for the same information. However the modes of the first algorithms (DES and 3DES) behave quite differently: the decryption mode requires about fifteen times more energy.

In a wider context, from a user perspective, one could interpret the presented results in the following way. There is a great deal of variability of using the results. The basic variable features, as related to the energy consumption, are:

- algorithm class (symmetric, asymmetric, encryption, decryption);
- algorithm type (for our case DES, 3DES, AES, RC2);
- block size (for DES, 3DES), and length of key (128, 192**, 256**, for AES and RC2; ** - are not implemented in the Framework we have used);
- information type a user needs to manage (secret, unsecured);
- the amount of the information for each type;
- mode of the information is to be managed (no use of the algorithms, decryption and the use of a particular part of information, decryption-use-encryption, use-encryption).

Having the results, such as ones in Fig. 10, knowing the needs for the information safety levels, the amount of information to be protected and current state of the battery, a user can reasonably decide of how the task is to be managed with the energy savings in mind.

However, the better strategy is to develop a program that using the collected data would give the prediction and advice on energy/safety trade-offs for a user depending on his/her profile.

References

1. **Pokharel M., Park J.** Cloud computing: future solution for e-governance // Proc. of the 3rd International Conference on Theory and Practice of Electronic Governance. – Bogota, Columbia, November 10–13, 2009. – P. 409–410.
2. Gartner. Gartner News Room, January 18, 2010. http://www.gartner.com/it/page.jsp?id=1282413.
3. **Kim H., Smith J., Shin K.** Detecting Energy-Greedy Anomalies and Mobile Malware Variants // Proc. of MobiSys'08. – Breckenridge, Colorado, USA, June 17–20, 2008. – P. 239–252.
4. **Garret M.** Powering down // Communication of the ACM. – New York: ACM media, 2009. – Vol. 51. – No. 9. – P. 43–46.
5. **Venugopalan R., Ganesan P., Peddabachagari P.** Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis // Proc. of CASES'03. – San Jose, California, USA, 2003. – P. 188–197.
6. Microsoft Corporation, .NET Framework Developer Center. http://msdn.microsoft.com/en-us/netframework/default.aspx
7. **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.
8. **Raghunathan A., Ravi S., Hattangady S., Quisquater J–J.** Securing Mobile Appliances: New Challenges for the System Designer // Proc. of the 3d IEEE Int. Conf. on Design, Automation and Test in Europe (DATE'03), IEEE 2003.
9. **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 03 22

J. Toldinas, V. Štuikys, G. Ziberkas, D. Naunikas. Power Awareness Experiment for Crypto Service-Based Algorithms // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 5(101). – P. 57–62.

In the context of cloud computing two interrelated problems (awareness of power consumption and information safety of mobile devices) are highly important issues. The paper presents some results of experiments we have carried with 4 standard cryptography-based algorithms (AES, RC2, DES and 3DES) aiming to identify their greediness for energy, as well as to collect data for identification of relationships among various characteristics (folder size, information type, security level, algorithm type, encryption/decryption mode, performance/energy, etc.). The experiment is based on our previously developed methodology. The basic results are: 1) performance/energy characteristics are linearly dependent on the folder size and practically independent upon the information type (document or picture), 2) decryption requires much more energy resources then encryption for some class of algorithms. Ill. 10, bibl. 9, tabl. 2 (in English; abstracts in English, Russian and Lithuanian).

Е. Толдинас, В. Штуйкис, Г. Зиберкас, Д. Наунинас. Эксперимент определения потребляемой энергии алгоритмами на базе крипто-сервиса // Электроника и электротехника. – Каунас: Технология, 2010. – № 5(101). – С. 57–62.

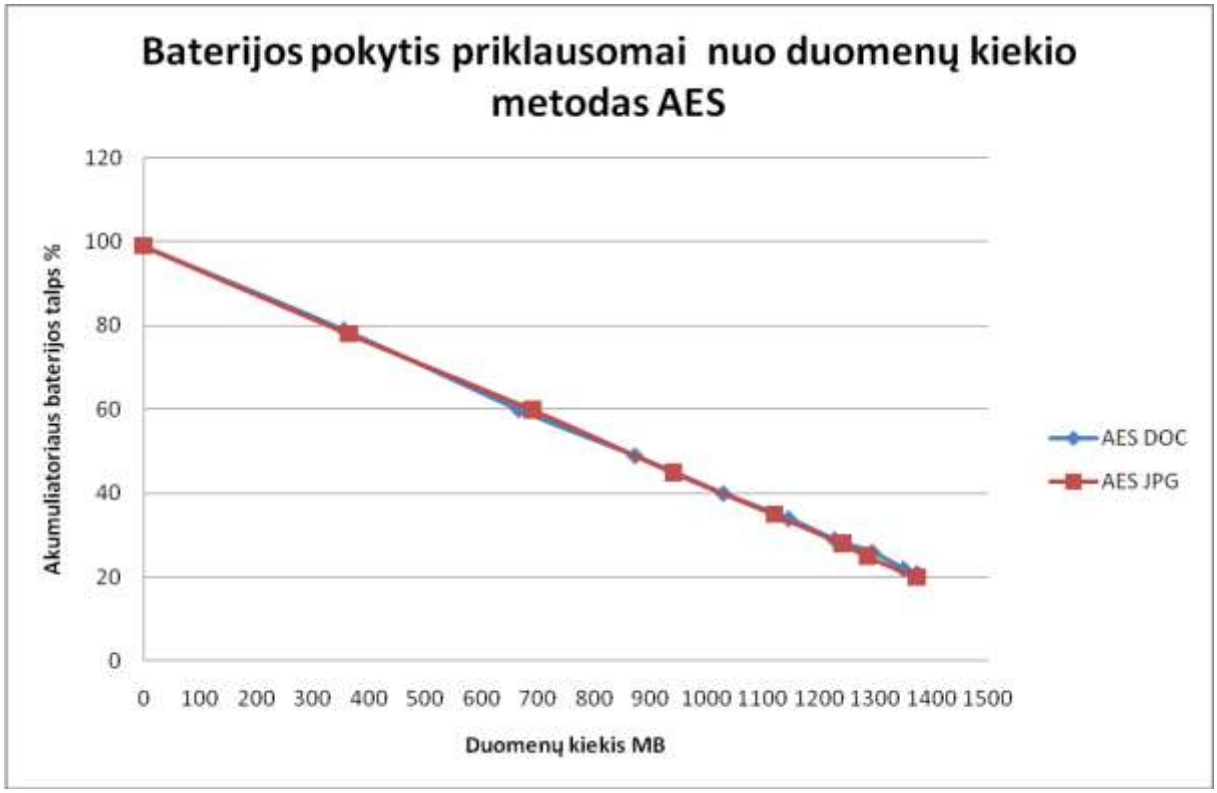
В контексте распределенных вычислительных ресурсов две взаимозависимые проблемы (понимание процесса потребления энергии и безопасность хранения информации в мобильных устройствах) имеют особое значение. В статье представлены результаты эксперимента, проведенного нами с 4-мя алгоритмами шифрования данных (AES, RC2, DES and 3DES) с целью определения их энергоемкости, а также сбора данных для идентификации взаимозависимости между различными характеристиками (размер каталога, тип информации, уровень безопасности, тип алгоритма, режим шифрования/дешифрования, быстродействие/энергопотребление и т.д.). Эксперимент основан на ранее разработанной нами методологии. Основные результаты: 1) характеристики быстродействия/энергопотребления линейно зависимы от размера каталога и практически независимы от типа информации (текст или графика), 2) в отличие от шифрования, дешифрование требует больших ресурсов энергии для некоторых алгоритмов. Ил. 10, библ. 9, табл. 2 (на английском языке; рефераты на английском, русском и литовском яз.).

J. Toldinas, V. Štuikys, G. Ziberkas, D. Naunikas. Energijos suvartojimo eksperimentas kriptografijos paslaugos algoritmams // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 5(101). – P. 57–62.

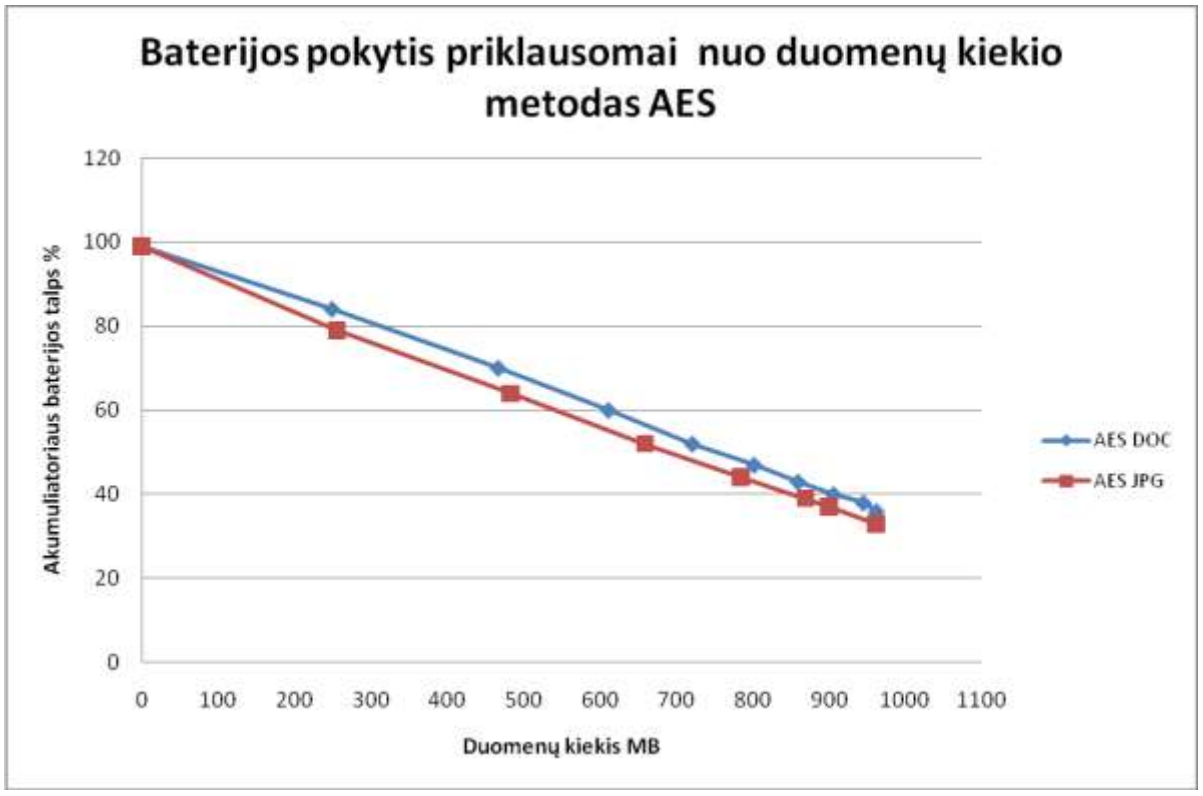
Virtualizacijos ir mobilumo kontekste dvi tarpusavyje susietos problemos, t.y. energijos suvartojimo žinojimas ir informacijos apsauga mobiliuose įtaisuose, yra labai svarbios. Šiame straipsnyje pateikiami kai kurie eksperimento rezultatai su 4 standartiniais kriptografijos algoritmais (DES, 3DES, AES, RC2) siekiant nustatyti jų 'godumą' energijai, o taip pat surinkti duomenis, kad būtų galima identifikuoti priklausomybes tarp įvairių charakteristikų (katalogo dydžio, informacijos tipo, saugumo laipsnio, algoritmo tipo, šifravimo/dešifravimo elgsenos, našumo/laiko ir energijos). Eksperimentas atliktas pagal anksčiau pasiūlytą metodiką. Pagrindiniai rezultatai tokie: 1) našumo/energijos charakteristikos yra tiesiškai priklausomos nuo katalogo, kurį reikia šifruoti/dešifruoti, dydžio ir praktiškai nepriklauso nuo informacijos tipo (tekstas ar paveikslas); 2) iššifravimas reikalauja žymiai (net 15 kartų) daugiau energijos negu šifravimas kai kuriems algoritmams (DES ir 3DES). Il. 10, bibl. 9, lent. 2 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).

8.3. Papildomi tyrimo rezultatai

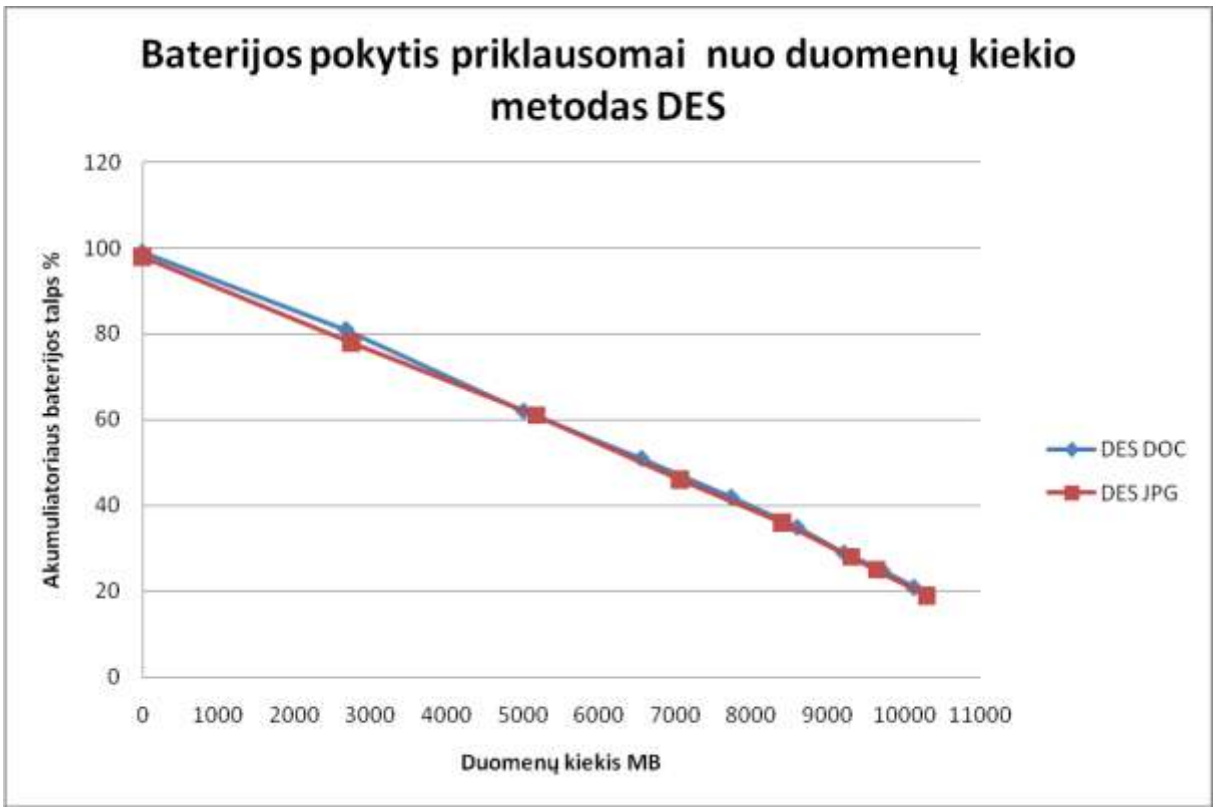
Visi eksperimento rezultatai yra surašyti į Microsoft Excel failus. Juose taip pat yra ir rezultatai atvaizduoti grafiškai.



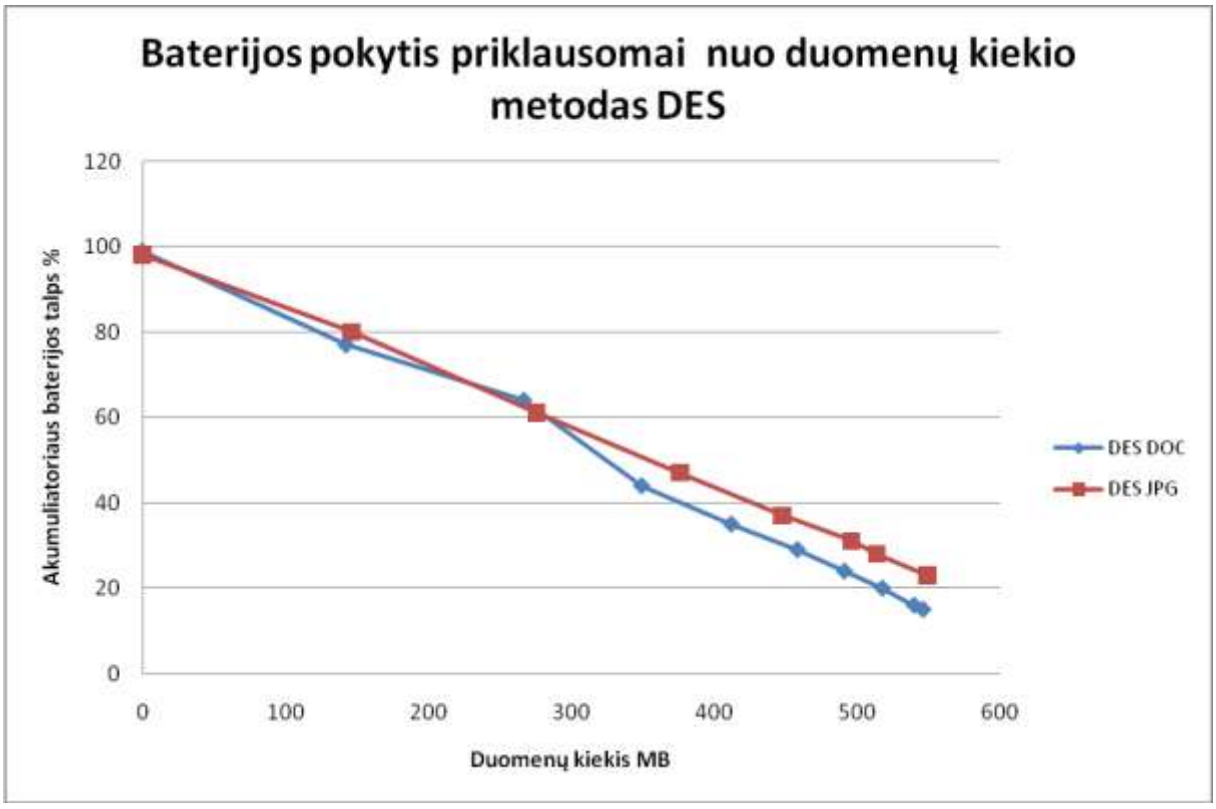
36 pav. Duomenys gauti užšifruojant



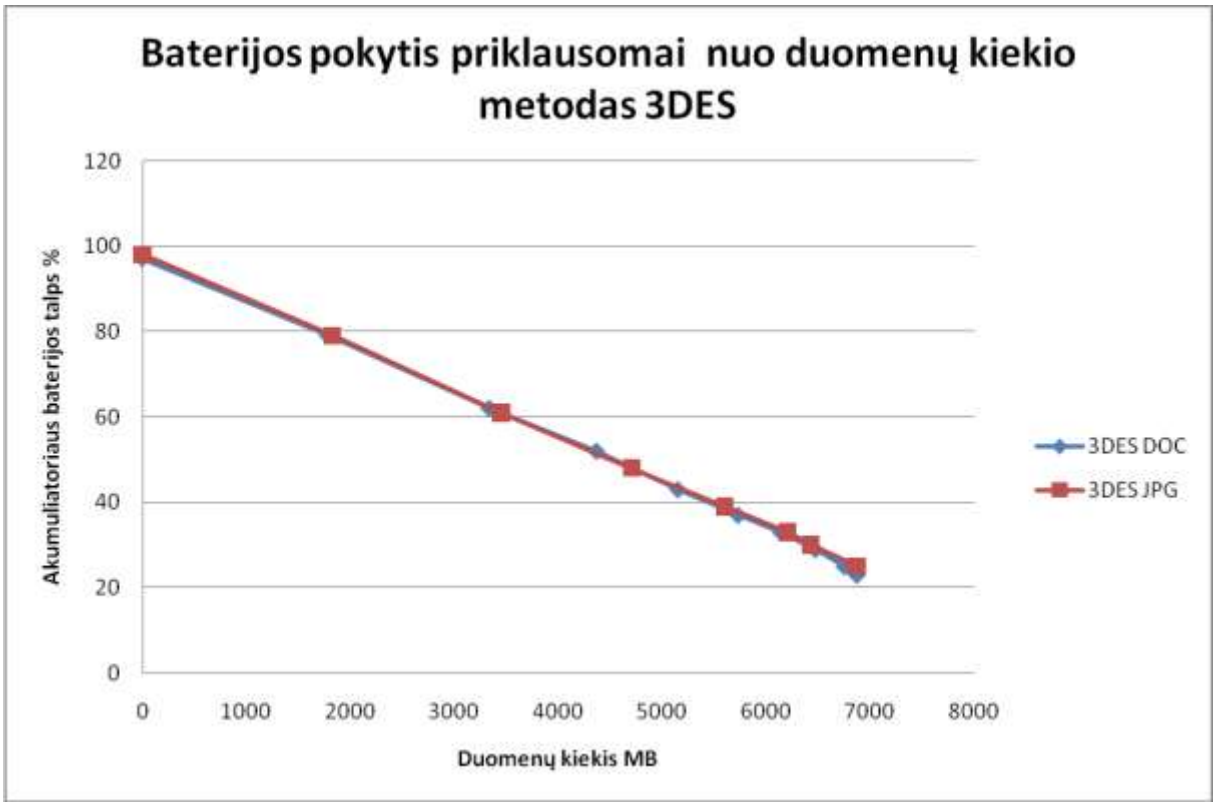
37 pav. Duomenys gauti iššifruojant



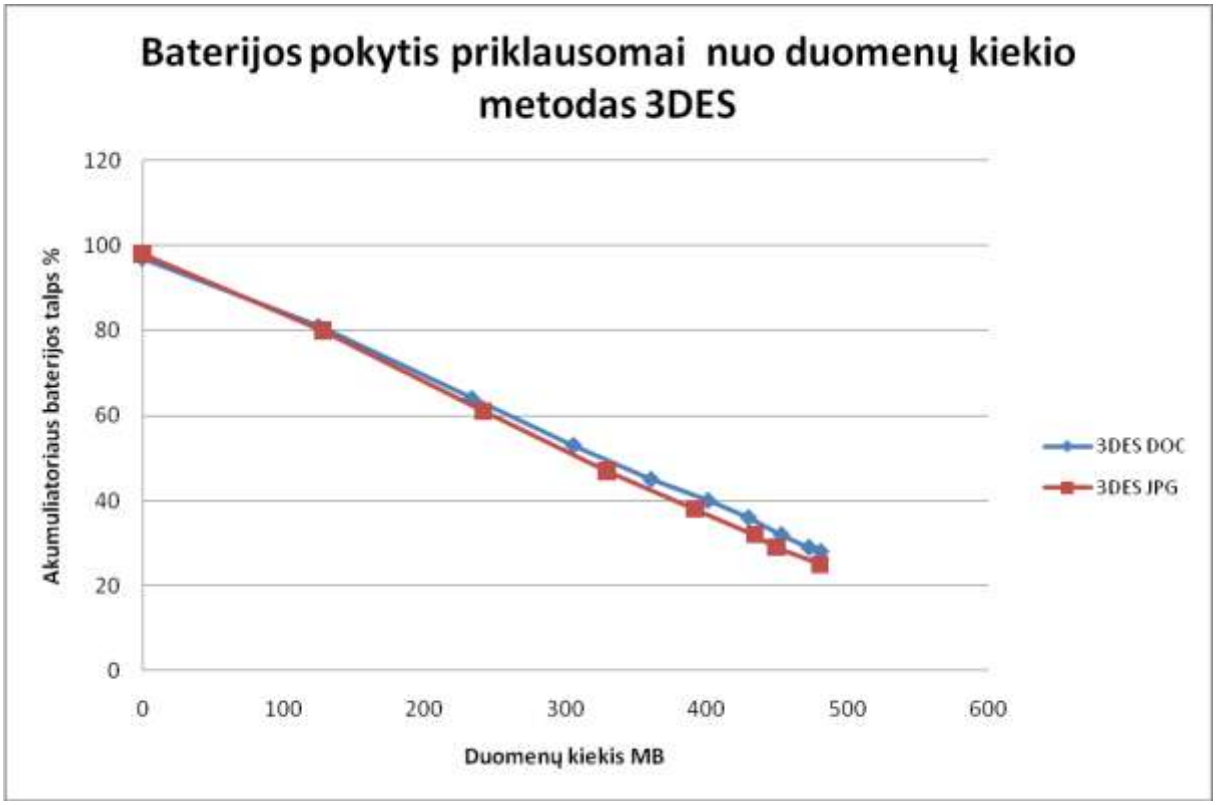
38 pav. Duomenys gauti užšifruojant



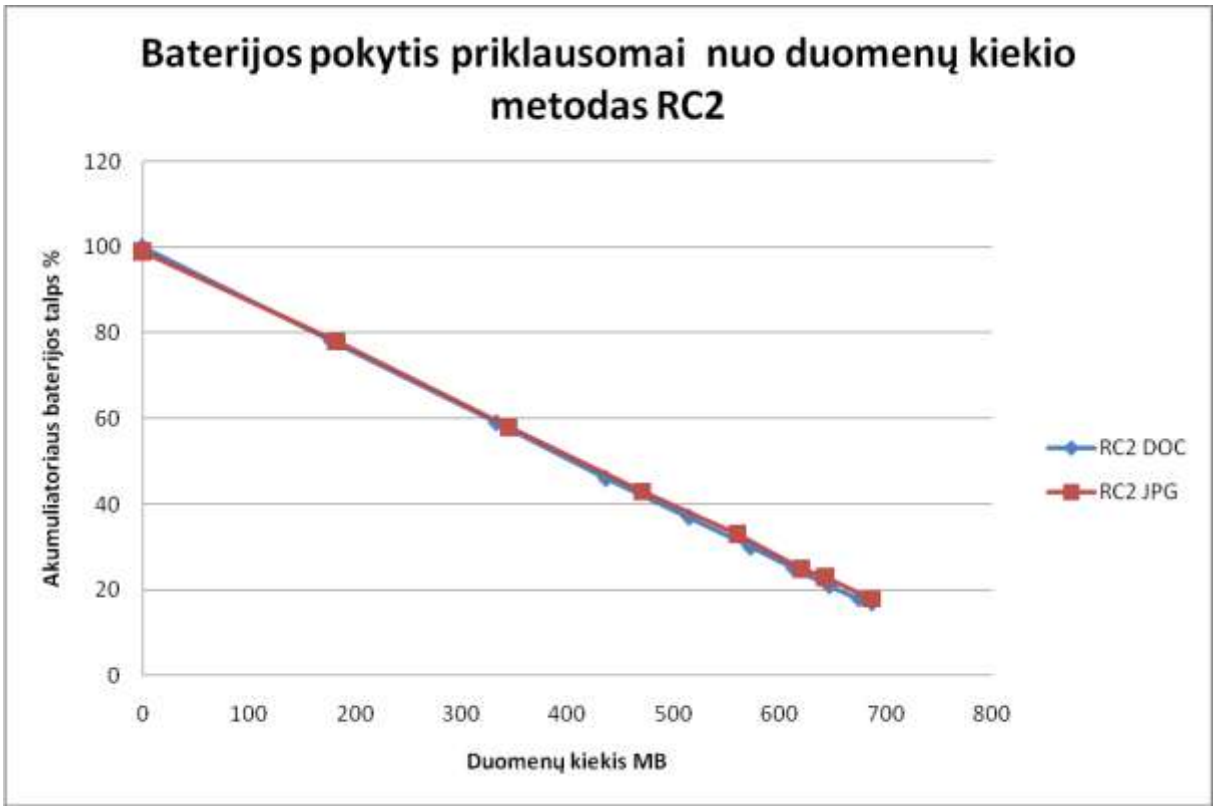
39 pav. Duomenys gauti iššifruojant



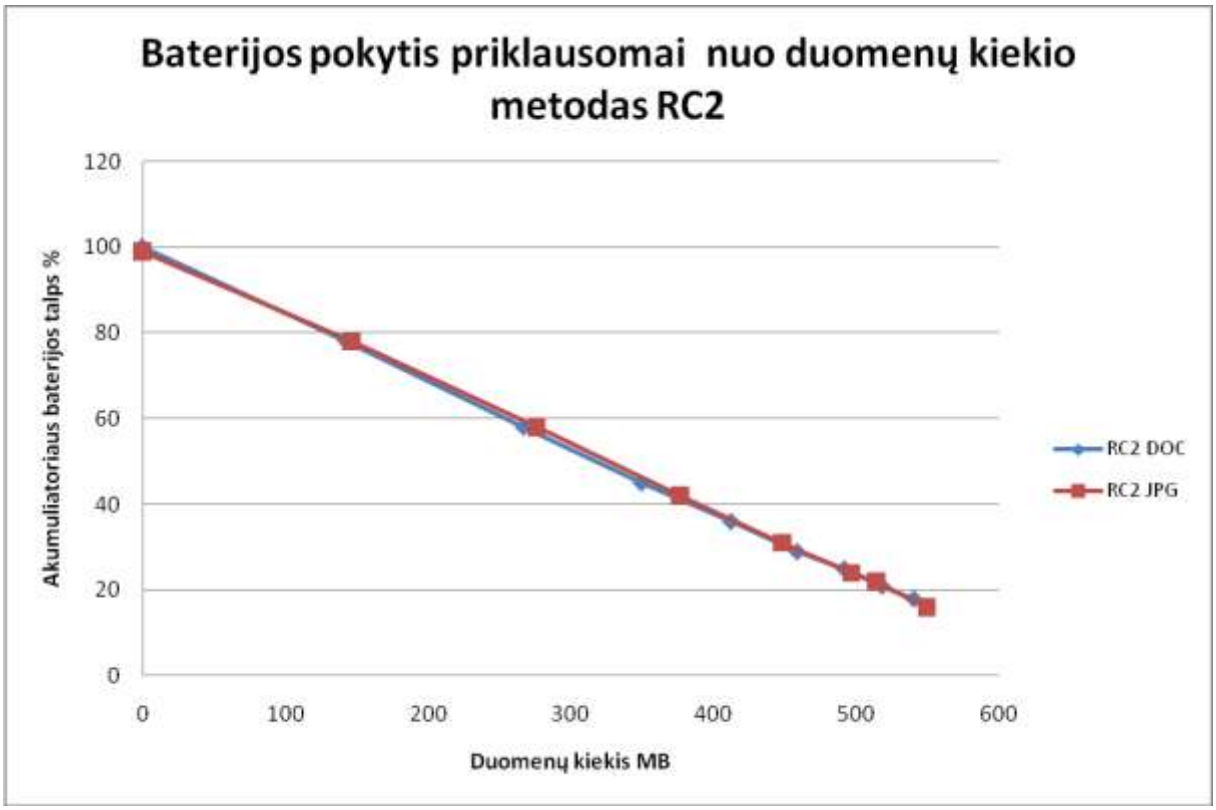
40 pav. Duomenys gauti užšifruojant



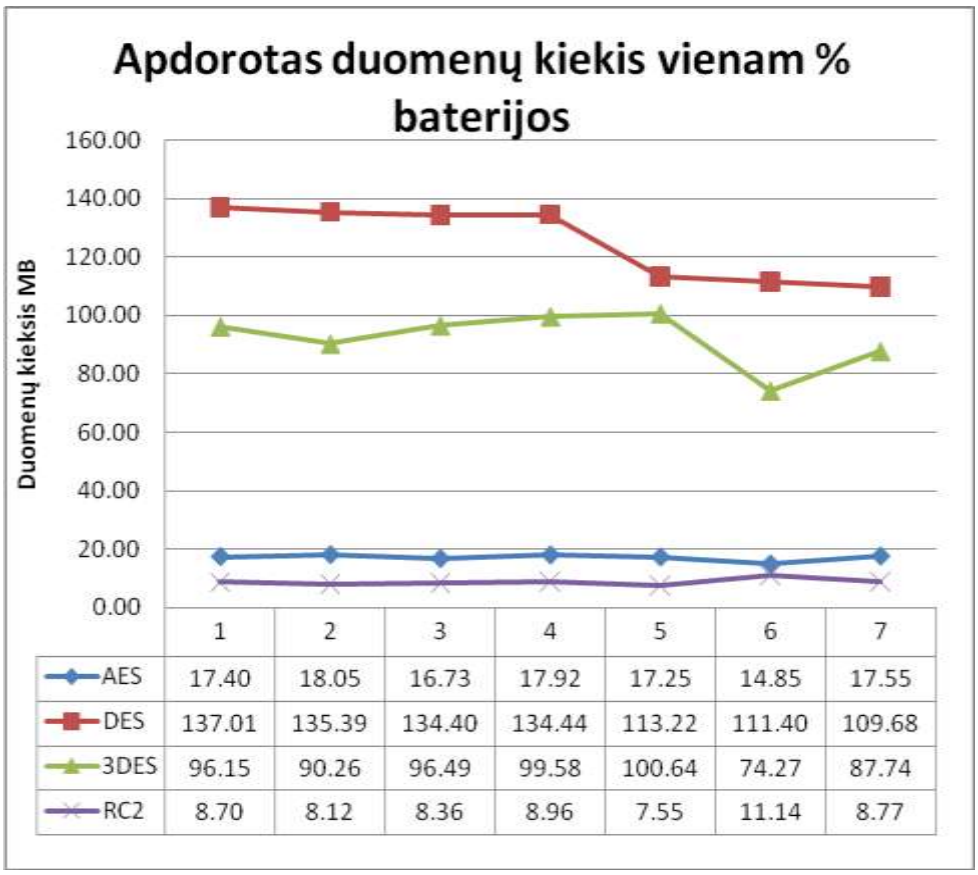
41 pav. Duomenys gauti iššifruojant



42 pav. Duomenys gauti užšifruojant

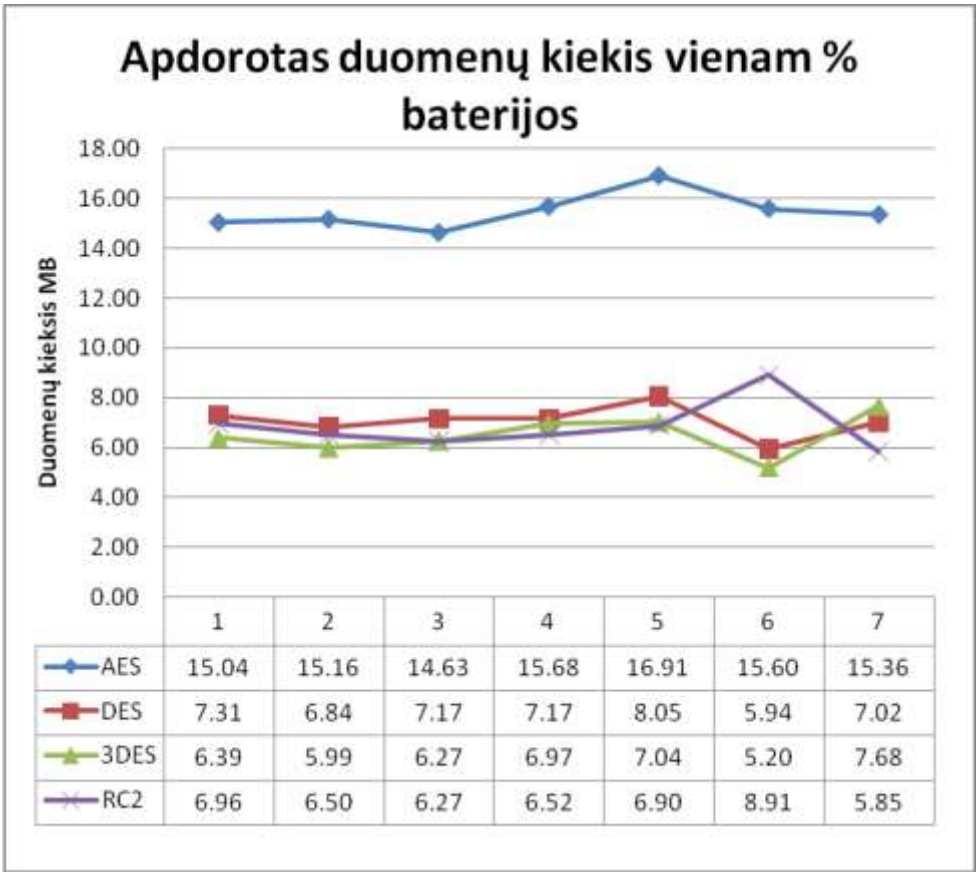


43 pav. Duomenys gauti iššifruojant



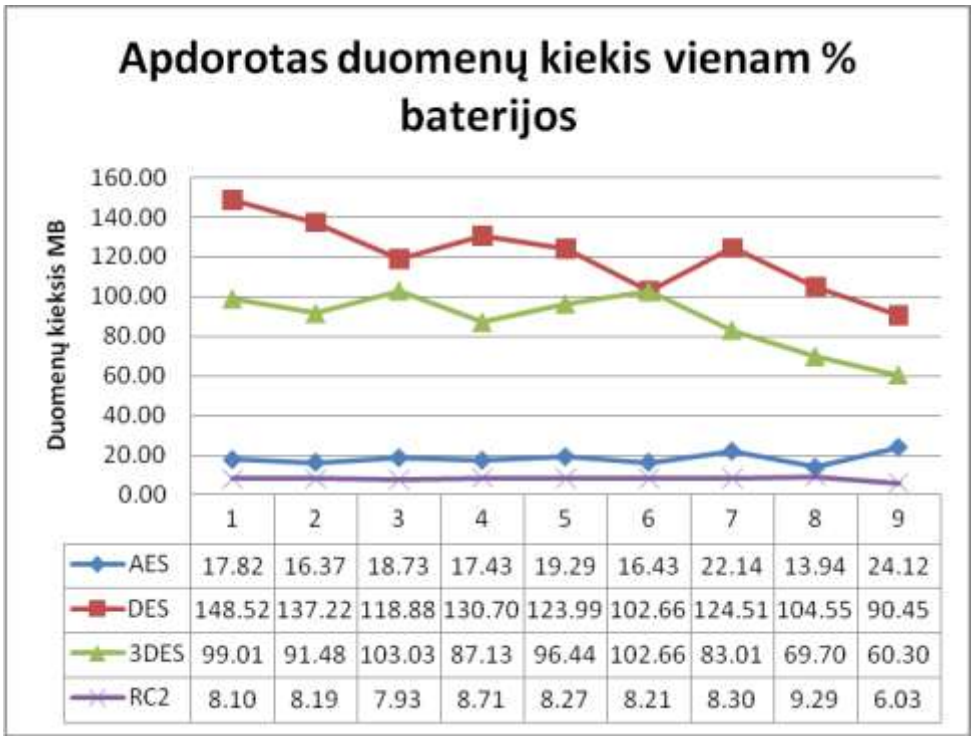
	1	2	3	4	5	6	7
AES	17.40	18.05	16.73	17.92	17.25	14.85	17.55
DES	137.01	135.39	134.40	134.44	113.22	111.40	109.68
3DES	96.15	90.26	96.49	99.58	100.64	74.27	87.74
RC2	8.70	8.12	8.36	8.96	7.55	11.14	8.77

44 pav. Duomenys gauti užšifruojant grafinius-jpg tipo failus

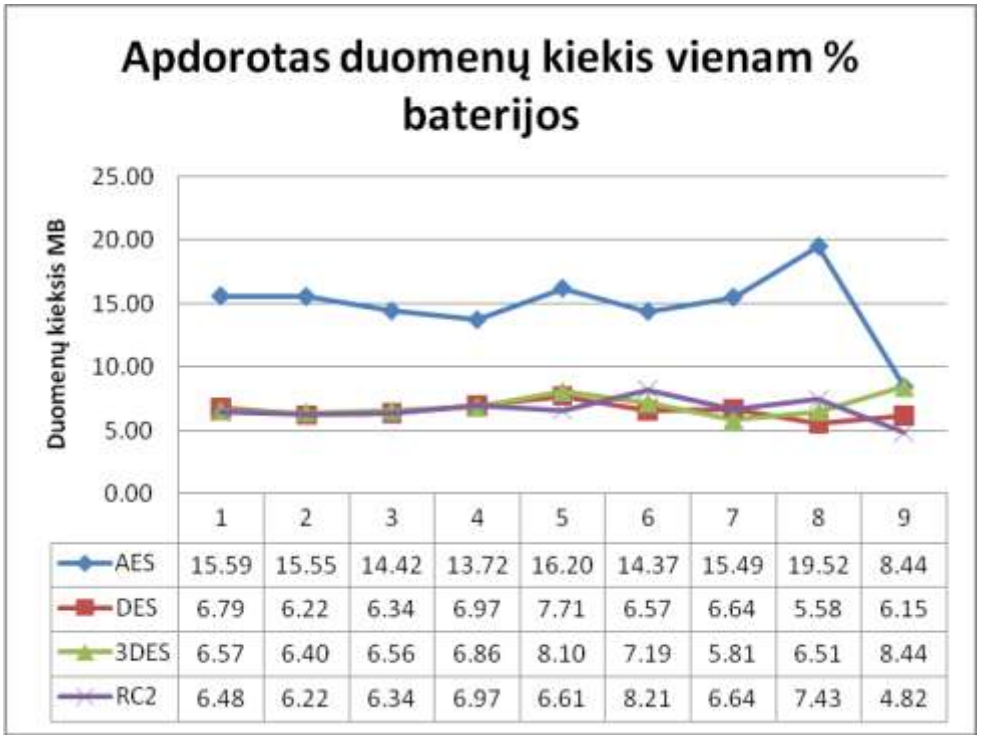


	1	2	3	4	5	6	7
AES	15.04	15.16	14.63	15.68	16.91	15.60	15.36
DES	7.31	6.84	7.17	7.17	8.05	5.94	7.02
3DES	6.39	5.99	6.27	6.97	7.04	5.20	7.68
RC2	6.96	6.50	6.27	6.52	6.90	8.91	5.85

45 pav. Duomenys gauti iššifruojant grafinius-jpg tipo failus



46 pav. Duomenys gauti užšifruojant tekstinius-doc tipo failus



47 pav. Duomenys gauti iššifruojant tekstinius-doc tipo failus

8.4. Informacinė sistema

Šiame priede yra įdėti tyrimui naudotos informacinės sistemos failai, tyrimo skaitiniai bei grafiniai rezultatai *.csv, bei *.xls tipo failuose.