KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Donatas Burba

Piršto atspaudo naudojimas šifravimo rakto generavimui

Magistro darbas

Darbo vadovas
doc. dr. Algimantas Venčkauskas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Donatas Burba

Piršto atspaudo naudojimas šifravimo rakto generavimui

Magistro darbas

Recenzentas
doc. dr. Tomas Skersys
2010-05-31

Vadovas
doc. dr. Algimantas Venčkauskas
2010-05-31

Atliko
IFN-8/3 gr. stud.
Donatas Burba
2010-05-31

Kaunas, 2010

Turinys

RESEARCH OF ENCRYPTION KEY GENERATION FROM FINGERPRINT

SUMMARY

Only encrypted data can be treated as secure data and encryption is impossible without encryption key. One of the best known and widely used encryption keys is password, but the main its drawback is necessity to remember it. Biometrics may help to avoid this situation, because everyone has unique characteristics. But the main question is how to extract encryption key from biometric data.

Fingerprints are well known biometric characteristic, used for people identification or authentication and fingerprint readers integrated into USB flash drives or laptops don't cause surprise any more. Every fingerprint can be described using minutiae points' matrix and from this matrix encryption key can be generated. But fingerprints of the same finger aren't identical, so this must be kept in mind as well.

In this research one method of direct encryption key generation from fingerprint is introduced. Minutiae matrix is structured from fingerprint image; parameters are formed and passed to encryption key generators. Two products were used for making matrix and eight generators were produced, generating encryption keys length of 64 and 128 bits. This system was tested with prepared fingerprint set and all the results are given.

Key words: encryption key, password, biometrics, fingerprint, minutiae points.

ĮVADAS

Laikui bėgant vis daugiau žmonių supranta, jog išties informacija – brangiausias turtas ir norint ją išlaikyti būtina pasirūpinti saugumu. Informacijos ir duomenų saugumas jau nėra suprantamas vien kaip jos įdėjimas į patikimą seifą. Spartėjant gyvenimo tempui ir tobulėjant technologijoms vis didesnė dalis duomenų saugoma skaitmeninėje formoje, todėl turi būti taikomi ir atitinkami saugos metodai.

Duomenų saugumas dažniausiai vienareikšmiškai tapatinamas su jų šifravimu. Šifravimas – tai duomenų formos pakeitimas taip, kad ne kiekvienas žmogus galėtų juos suprasti. Užšifruotus duomenis suprasti, t.y. iššifruoti ir perskaityti, galima tik turint teisingą dešifravimo raktą. Kartais šifravimas būna pavadinamas kodavimu, tačiau tai nėra vienas ir tas pats, kadangi kodavimo proceso metu nenaudojamas joks raktas. Taigi duomenų kodavimui užtenka tik tam tikro algoritmo, o šifravimui reikalingas ir specialus algoritmas, ir slaptas raktas.

Būtent šifravimo/dešifravimo raktai ir jų valdymas sukelia daugiausiai problemų visame šifravimo procese. Šiame darbe ir bandyta analizuoti kaip šifravimo raktų naudojimo ir valdymo problema sprendžiama egzistuojančiose sistemose, kur jų stipriosios ir silpnosios pusės, kokios galimos alternatyvos. Atlikus literatūros šaltinių analizę dar kartą pasitvirtino faktas, jog egzistuoja dvi pagrindinės problemos: šifravimo raktą reikia arba atsiminti, arba saugiai laikyti. Apžvelgus šiandien siūlomas duomenų šifravimo priemones įsitikinta, jog šios problemos vis dar nėra išspręstos.

Tolesnė analizė parodė, kad paminėtos problemos gali būti išspręstos naudojant biometriją ir tam tikri žingsniai šia linkme jau padaryti. Nors paprastai biometrinės šifravimo sistemos rūpinasi šifravimo raktų saugojimu ir išdavimu, tačiau yra užuominų apie biometrinių charakteristikų naudojimą tiesioginiam šifravimo raktų generavimui. Būtent toks kelias pasirinktas šiame darbe kaip pagrindą naudojant pirštų atspaudus.

Taigi šio *darbo tyrimo sritis* yra šifravimo raktai, o *tyrimo objektas* – pirštų atspaudų naudojimas tiesioginiam šių raktų generavimui. Būtent pirštų atspaudai pasirinkti ne šiaip sau, o įvertinus, jog tai viena plačiausiai naudojamų, geriausiai ištobulintų ir vartotojus mažiausiai

atbaidančių biometrinių charakteristikų. Todėl šio *darbo tikslas* – sudaryti metodą, pagal kurį šifravimo raktus būtų galima tiesiogiai generuoti iš pirštų atspaudų.

Darbo tikslui pasiekti iškelti pagrindiniai uždaviniai:

- apsibrėžti metodo taikymo sritį ir keliamus reikalavimus
- išsiaiškinti, kaip piršto atspaudas gali būti aprašytas patogiam tolesniam apdorojimui
- sudaryti ir realizuoti patį metodą bei atlikti eksperimentą

Analizuojant pirštų atspaudų aprašymo specifiką išsiaiškinta, jog dažniausiai remiamasi *minutiae* taškais [19], kurių žemėlapis (*angl. map*) sudaromas naudojant kelis etapus [20, 21]. Taip pat apžvelgti praktikoje taikomi pirštų atspaudų palyginimo metodai [22, 23, 24], kurių esmė – geometrinių figūrų konstravimas ir palyginimas. Yra algoritmų, naudojančių ir vieną išskirtinį *minutiae* tašką – branduolį (*angl. core point*). Jam nustatyti taip pat taikomi specialūs metodai, tačiau kaip tai padaryti praktiškai šiame darbe nesigilinta [24, 25].

Stabiliausi *minutiae* taškai (logiška, kad juos ir reikia naudoti) yra išsidėstę arčiausiai minėtojo branduolio, todėl darbe nuspręsta naudoti 200 x 200 vaizdo taškų pirštų atspaudų sritis. Būtent tokio dydžio pirštų atspaudų rinkiniai paruošti metodo testavimui ir eksperimento atlikimui ir tai yra baziniai pradiniai duomenys iš kurių metodas sugeneruoja rezultatus.

Kad nespręsti nereikalingų uždavinių, darbo eigoje metodas suskaidytas į dvi dalis: *a) minutiae* taškų išskyrimo ir matricos formavimo bei *b)* šifravimo raktų generavimo. Pirmajai daliai nuspręsta pasinaudoti jau esančiais įrankiais, kurie suformuotų 200 x 200 dydžio *minutiae* taškų dvejetainę matricą, kurią toliau kaip pradinius duomenis naudos raktų generavimo metodas. Matricos formuotos dviem įrankiais (algoritmais), o projektinė ir realizacijos dalys orientuotos į raktų generavimo iš gautos matricos principų sudarymą ir realizavimą.

Pats pasiūlytas metodas taip pat sudarytas iš dviejų loginių dalių: *a)* parametrų formavimo ir *b)* raktų generavimo. Pirmojoje dalyje formuojamos atkarpos, skaičiuojami jų ilgiai bei posūkių kampai sudarytoje polinėje koordinačių sistemoje. Iš taip sudarytų parametrų jiems pritaikius atitinkamus koeficientus generuojami šifravimo raktai. Pritaikius keletą turimų parametrų ir koeficientų kombinacijų sudaryti 8 raktų generatoriai (keturi iš jų generuoja 64, kiti keturi – 128 bitų ilgio šifravimo raktus).

Eksperimentas atliktas ir metodas tikrintas naudojant jau minėtus pirštų atspaudų rinkinius. Kiekviename rinkinyje buvo po 8 to paties piršto atspaudus. Iš kiekvieno atspaudo

suformuota po 2 matricas (naudojant skirtingus algoritmus), iš kurių generuoti šifravimo raktai ir tikrintas jų pastovumas rinkinyje. Deja rezultatai nuvylė, kadangi visi sugeneruoti raktai buvo skirtingi, net ir gauti iš to paties piršto atspaudų rinkinio.

Rezultatų analizė parodė, kad dėl to pirmiausia kalta ne spraga projektinėje ar realizacijos dalyje, ne matricų formavimo algoritmų invariantiškumo nebuvimas, o naudotų *minutiae* taškų radimo algoritmų netobulumas. Iš tiesų nors naudoti atspaudai rinkiniuose vizualiai skyrėsi labai nežymiai, tačiau juose buvo nustatomi skirtingi taškai, todėl formuojamos skirtingos matricos ir generuojami nesutampantys šifravimo raktai. Taigi naudojant šiame darbe pasirinktas priemones iš pirštų atspaudų generuojami raktai praktikoje negali būti pritaikomi. Nors ir buvo numatytos klaidų tolerancijos priemonės, apsibrėžta, jog *klaidingas priėmimas* (*angl. False Acceptance Rate, FAR*) šiuo atveju yra labiau toleruotinas nei *klaidingas atmetimas* (*angl. False Rejection Rate, FRR*), tačiau matricos buvo per daug skirtingos, kad iš jų sugeneruoti vienodus raktus.

Galbūt darbe pasiūlytas metodas išties būtų visai neblogai pritaikomas praktikoje, tačiau pirmiausia reikalingas patikimas ir pirštų atspaudų neatitikimams tolerantiškas *minutiae* taškų radimo metodas, kurio pagalba vieno piršto atspaudams būtų formuojamos kiek įmanoma mažiau besiskiriančios matricos. Galbūt toks metodas jau egzistuoja, galbūt jį dar reikia sugalvoti, tačiau tai – pirmas ir pagrindinis žingsnis siekiant sėkmingai pakartoti šio darbo eksperimentą ir pasiūlytu būdu piršto atspaudus panaudoti tiesioginiam šifravimo rakto generavimui.

Darbo struktūra:

- Pirmajame skyriuje apžvelgtos šiuo metu siūlomos ir naudojamos duomenų šifravimo priemonės, atkreipiant dėmesį į naudojamus šifravimo raktų valdymo principus. Aptartos ir palygintos galimos biometrinių sistemų alternatyvos ir pačios biometrinės charakteristikos.
- Antrajame skyriuje išsikeltas darbo tikslas, suformuluoti uždaviniai bei apsibrėžta metodo taikymo sritis bei pagrindiniai reikalavimai. Išsiaiškinti pirštų atspaudų požymiai, jų aprašymo kompiuteryje principai. Taip pat aprašyta siūlomo metodo pagrindinė idėja bei pateikta detalizacija, iliustruota pavyzdžiu. Apsibrėžtas parametrų formavimo principas šifravimo raktų generatoriui bei sudaryta po keturis generatorius 64 ir 128 bitų ilgio šifravimo raktams generuoti.

- Trečiajame darbo skyriuje aptarti metodo realizacijai reikalingi įrankiai, apsibrėžta, kas konkrečiai bus realizuojama, pristatytos pagalbinės naudojamos programinės priemonės. Taip pat pateikiamas realizuojamos dalies veikimo algoritmas, testavimo modelis bei pradiniai duomenys eksperimento atlikimui.

- Ketvirtasis skyrius skirtas eksperimentui. Apsibrėžtas eksperimento tikslas, siekiami atsakyti klausimai bei pati eksperimento eiga. Pateikiami gauti rezultatai, jų analizė bei paminėti tolesni galimi (šiuo atveju netgi būtini) darbai.

1. ŠIFRAVIMO RAKTŲ VALDYMO METODŲ IR PRIEMONIŲ ANALIZĖ

1.1. Tyrimo sritis, objektas ir problema

Informacija ir įvairūs duomenų rinkiniai jau seniai tapo vienu iš svarbiausių dalykų šiuolaikinėje visuomenėje. O viena pagrindinių su jais susijusių problemų - saugumas. Anksčiau ši problema paprastai buvo aktuali tik valstybinėms institucijoms, įmonėms ir kitoms, dažniausiai komercinėms, įstaigoms. Tačiau paskutiniu metu ir paprasti vartotojai vis didesnį dėmesį skiria savo duomenų saugumu, suprasdami jo svarbą.

Duomenys laikomi saugiais kai jie yra užšifruoti, o tam reikalingas šifravimo algoritmas ir šifravimo raktas. Algoritmas yra viešoji šifravimo sistemos dalis, jų yra sugalvota daug ir įvairių. Šifravimo algoritmų detalizacijos viešai prieinamos visiems norintiems. Slaptoji šifravimo sistemos dalis – raktas, kurį žinantis vartotojas gali perskaityti užšifruotus duomenis. Taigi pagrindinė šifravimo sistemos problema – saugus raktų valdymas.

Kiekviena duomenų šifravimo paslaugą teikianti sistema turi vienokį ar kitokį raktų valdymo posistemį. O ir patys raktai gali būti kuo įvairiausi – nuo vartotojo pasirinktų iki sistemos pasiūlytų. Saugumo požiūriu šifravimo mechanizmas yra mažiau įdomus nei jo naudojamų raktų valdymas. Pagrindinės ir dažniausiai naudojamos duomenų šifravimo priemonių grupės ir jų pavyzdžiai bus paminėti tolesniuose skyreliuose. Juos apžvelgiant labai nesigilinsime į patį šifravimo/dešifravimo procesą, didesnį dėmesį skirsime šifravimo raktams ir jų valdymui.

Paprasčiausias ir labiausiai įprastas raktas – slaptažodis, t.y. slapta simbolių seka. Slaptažodžiai naudojami prisijungimui prie kompiuterizuotos darbo vietos, banko paslaugų valdymo sistemų, internetinių puslapių bei įvairiose kitose vietose. Tačiau pagrindinė tokių raktų problema pirmiausia ta, jog vartotojas juos turi atsiminti. Todėl neįvertinus galimų pasekmių sukeliamos saugumą mažinančios situacijos: naudojami labai lengvai atspėjami slaptažodžiai, jeigu pasirenkamas sudėtingesnis – toks pat slaptažodis naudojamas keliose vietose arba užsirašomas. Visais atvejais rizika, jog asmeninius ar kitus slaptus duomenis galės peržiūrėti tretieji asmenys, smarkiai išauga, todėl moderniose sistemose slaptažodžių stengiamasi visiškai atsisakyti. Tas pats galioja tiek šifravimo, tiek įvairiems kitiems slaptiems raktams.

Egzistuoja ir tokių sistemų, kuriose jokio rakto atsiminti nereikia. Tačiau tuomet dažniausiai susiduriama su kita problema: jį reikia kažkur saugiai laikyti. Tam dažniausiai

naudojamos *USB* atmintinės, protingosios kortelės (*angl. smart card*) ir kitokios laikmenos. Šiuo atveju niekas negali garantuoti, jog raktu visuomet naudosis tik savininkas, jis nebus pamestas ar nukopijuotas. Taip pat galimas ir šio bei anksčiau paminėto variantų derinys: vartotojas įveda slaptažodį (kurį reikia atsiminti), o sistema jam suteikia raktą (kuris saugomas sistemoje). Tokiu atveju slaptažodis reikalingas tikrojo šifravimo ar kitokio slapto rakto atrakinimui.

Taigi raktų valdyme realiai egzistuoja dvi problemos: raktą reikia atsiminti arba kažkur saugiai laikyti. Nei vienas, nei kitas būdas nėra nei pakankamai patogus, nei patikimas. Todėl stengiamasi ieškoti būdų, kaip vartotojas būtų mažiausiai apkraunamas papildomais veiksmais ir informacija, o jo įsikišimas į sistemos veikimą (įvedant raktą) būtų minimalus.

Viena iš šifravimo rakto alternatyvų, kurių nereikia nei atsiminti, nei saugoti - biometriniai duomenys. Jau seniai žinoma, kad priklausomai nuo pasirinktos biometrinės charakteristikos šie duomenys paprastai yra daugiau ar mažiau unikalūs. Saugumo sistemos, kuriose naudojamas veido, balso atpažinimas, akies rainelė ar piršto atspaudas, pasaulyje vis dažniau naudojamos ir atitinkamose srityse jos puikiai pasiteisina.

Biometrinės sistemos gali būti naudojamos šifravimo rakto „atrakinimui“, t.y. prieigai prie jo užtikrinti. Tačiau šiuo atveju viena minėtųjų problemų išlieka – pats raktas turi būti saugomas sistemos viduje. Kai biometrinės sistemos naudojamos ne raktų išdavimui, o vartotojų identifikavimui/autentifikavimui, rakto atsiminimo ir jo saugojimo problemos išnyksta, tačiau šiame darbe kalbama būtent apie šifravimo raktus.

Nė viena sistema nėra tobula - tą patį galima pasakyti ir apie biometrinius raktus. Čia egzistuoja dvi pagrindinės problemos: klaidingas priėmimas (*angl. False Acceptance Rate, FAR*) ir klaidingas atmetimas (*angl. False Rejection Rate, FRR*), t.y. kai neregistruotas sistemoje vartotojas atpažįstamas kaip registruotas ir atvirkščiai (registruotas vartotojas sistemoje neatpažįstamas). Tai įvyksta todėl, kad kiekvienas naujai nuskaitytas biometrinis pavyzdys nėra visiškai identiškas prieš tai buvusiems – tai daugiausia lemia nevienoda duomenų pateikimo forma (tyliau/garsiau ištarta frazė balso atpažinime, galvos pasukimas į šoną veido atpažinime), žmogaus fiziniai pokyčiai (pakimęs balsas, įsipjautas pirštas). Todėl egzistuoja įvairūs metodai, apdorojantys biometrinę informaciją ir lyginantys naujus biometrinius pavyzdžius su jau esamais. Tačiau iki dar nėra nė vieno, kuris veiktų 100% tikslumu, todėl patikimumą galima vertinti tik nusistačius leistinas *FAR* ir *FRR* ribas.

Taigi pagrindinė šiame darbe sprendžiama duomenų šifravimo (saugumo) problema – šifravimo raktų valdymas. O esminį darbo klausimą galima suformuluoti taip: kaip gauti šifravimo raktą, kurio nereikėtų nei atsiminti, nei saugoti?

1.2. Esamų sistemų apžvalga

Kaip jau minėta, duomenų saugumą galima tapatinti su jų šifravimu. Todėl apžvelkime pagrindines duomenų šifravimo paslaugą teikiančias sistemas (produktus) ir jų grupes. Panagrinėkime kaip jų pagalba galime apsaugoti savo duomenis, didesnį dėmesį atkreipdami į naudojamus raktus ir jų valdymo principus.

Realizacijos požiūriu duomenų šifravimo metodus galima suskirstyti į dvi kategorijas:

- Aparatiniai sprendimai
- Programiniai sprendimai

Iš tikrųjų toks suskirstymas yra sąlyginis, kadangi ne visus egzistuojančius produktus galima vienareikšmiškai priskirti vienai iš šių kategorijų. Dažnai pagrindinis šifravimo mechanizmas realizuotas ir valdomas programinėmis priemonės, o aparatūra naudojama šifravimo/dešifravimo raktų saugojimui. Tačiau tai irgi nėra taisyklė, kadangi rinkoje galima rasti produktų, kur tiek šifravimu, tiek raktų valdymu rūpinasi vien aparatinė arba vien programinė įranga.

1.2.1. Operacinėse sistemose integruotos priemonės

Šiuo metu asmeniniuose kompiuteriuose dažniausiai aptiksime vieną iš trijų operacinių sistemų, t.y. *Microsoft Windows*, *Apple MacOS* ir *GNU/Linux*. Kiekvienoje iš jų yra numatyta duomenų šifravimo galimybė.

EFS (*Encrypting File System*). Tai *Windows* operacinėse sistemose naudojama technologija. *EFS* pasirodė kartu su *Windows 2000* versija ir veikia *NTFS* failų sistemoje. Duomenys šifruojami ir dešifruojami simetriniu raktu, kuris sugeneruojamas kiekvienai šifravimo sesijai. Simetrinis raktas užšifruojamas vartotojo viešuoju raktu ir įdedamas į jau užšifruoto failo antraštės *DDF* (*Data Decryption Field*) lauką. Viešojo ir privataus raktų pora (naudojama šifravimo raktų apsaugai) sugeneruojama sistemos viduje ir saugoma operacinėje sistemoje realizuotomis priemonėmis. [1]

BitLocker – technologija, pateikiama kartu su *Windows Vista* ir *Windows 7 Ultimate* ir *Enterprise* bei *Windows Server 2008* versijomis, šifruojanti visą loginį diską ir naudojanti 128 ar 256 bitų raktą. *BitLocker* geba išnaudoti *TPM* (*Trusted Platform Module*) mikroschemos, kuri bus apžvelgta **1.2.3** skyrelyje, galimybes. Žinoma tam reikalinga ir atitinkama (*Trusted Computing Group (TCG)* – compliant) *BIOS* sistema. Jeigu kompiuteris netenkina šių reikalavimų yra numatyta galimybė naudoti *USB* atmintinę. *BitLocker* naudoja hierarchinę šifravimo/dešifravimo raktų sistemą, o visi raktai saugomi tame pačiame šifruojamame logiciniame diske. [2]

FileVault – Mac OS operacinėje sistemoje (nuo versijos 10.3) realizuota duomenų šifravimo sistema. *FileVault* aktyvuojasi vartotojui prisijungus prie sistemos ir šifruoja jo namų (*angl. home*) katalogą. Duomenų šifravimui/dešifravimui naudojamas simetrinis 128 bitų raktas, kuris sugeneruojamas iš vartotojo slaptažodžio, kuriuo pastarasis jungiasi prie sistemos. Atsargumo sumetimais galima nurodyti ir atskirą slaptažodį, kurio pagalba taip pat bus galima prieiti prie užšifruotų duomenų. Abu slaptažodžiai saugomi operacinės sistemos viduje. [3]

EncFS yra *Linux* distribucijose naudojama ir vartotojo lygmenyje veikianti virtuali šifruota failų sistema. Ji sudaro tarpinį sluoksnį tarp realios failų sistemos ir vartotojo programų. Naudojant *EncFS* reikalingi du katalogai: pagrindinis (angl. *source*) bei prijungimo taškas (angl. *mountpoint*). Failai šifruojami ir dešifruojami pagrindiniame kataloge saugomu simetriniu raktu, kuris iššifruojamas įvedus teisingą slaptažodį. Šifravimui naudojami 128 arba 256 bitų raktai (priklausomai nuo naudojamo algoritmo). [4]

eCryptfs – tai kita *Linux* distribucijų siūloma šifruota failų sistema. Jos veikimo principas analogiškas kaip ir *EncFS* – sudaromas tarpinis sluoksnis tarp realios failų sistemos ir vartotojo programų. Skirtumas tik tas, jog ši sistema veikia branduolio lygmenyje. Kiekvienas failas šifruojamas naudojant atsitiktinai sugeneruotą raktą, kuris taip pat užšifruojamas ir patalpinamas į failo antraštę pagrindinėje failų sistemoje. Raktų generavimu ir valdymu rūpinasi *OpenPGP* pagrindu branduolio lygmenyje veikianti paslauga. [5]

1.2.2. Papildomos programinės priemonės – *TrueCrypt*

Pagal vartotojų atsiliepimus tai geriausia duomenų šifravimui skirta programine tik tarp nemokamų įrankių, tačiau ir apskritai.. Ši atviro kodo (*angl. open source*) programa veikia visose populiariausiose asmeniniams kompiuteriams skirtose operacinėse sistemose. Duomenų šifravimas vykdomas realiu laiku ir nereikalaujama jokio vartotojo įsikišimo. Nau-

dojamas šifravimo/dešifravimo mechanizmas pakankamai sudėtingas, jo aprašymą galima rasti programos tinklapyje. [6]

Šifravimo raktai (paprastai 256 bitų) generuojami pačios programos pagalba, o tam naudojamas vidinis generatorius, kuriuo gaunamos reikšmes priklauso nuo vartotojo aplinkos: t.y. pelės judesių, klaviatūros klavišų paspaudimo, tinklo veiksnumo ir kitų dalykų. Beje, kiekvienoje operacinėje sistemoje generatorius įtakojantys veiksniai skiriasi.

Naudojamas šifravimo raktas užšifruojamas vartotojo įvedamo slaptažodžio ir sistemos sugeneruojamo (ar vartotojo pasirenkamo) raktinio failo kombinacija. Dešifravimui reikalinga informacija sugeneruojama ir pagal tam tikrą struktūrą talpinama į šifruojamo skirsnio antraštę.

Prieš prijungiant šifruotą skirsnį prie sistemos, pagal tam tikrą algoritmą atliekamas šifravimo algoritmo, santraukos skaičiavimo algoritmo, raktų ilgių ir kitų parametrų nustatymas, kadangi konkreti informacija apie šiuos parametrus nėra saugoma. Tik po visų šių procedūrų skirsnis yra prijungiamas ir galima operuoti jame esančiais duomenimis.

Taigi programa tikrai galinga, o joje realizuotas raktų valdymo mechanizmas sudėtingas, tačiau tuo pačiu turėtų būti ir patikimas. Vienintelė blogybė, jog vartotojui visvien reikia atsiminti mažiausiai vieną slaptažodį.

Pagrindines programines duomenų šifravimo priemones aptarėme, toliau pasižiūrėkime kokius aparatinės realizacijos produktus vartotojas šiandien gali pasirinkti.

1.2.3. Aparatiniai sprendimai

TPM - tai *Trusted Computing Group* (*TCG*) organizacijos sukurta (ir vis dar tobulinama) technologija. Šios technologijos produktas - mikrovaldiklis, kuriame saugomi raktai, slaptažodžiai, skaitmeniniai saugumo sertifikatai, jo pagalba šifruojami kompiuteriu apdorojami duomenys. Prie tokiu moduliu saugomų duomenų galima prieiti tik tuo atveju, jei kompiuterio krovimosi metu įvedamas teisingas slaptažodis. Dažniausiai tokie moduliai montuojami kompiuterio pagrindinėje plokštėje, nors realiai jie gali būti naudojami visur, kur yra būtinybė. Šiuos saugumo modulius galima įsigyti tiek atskirai, tiek jau integruotus į kitus produktus. [7]

Pagrindinė vieta, kur vartotojai laiko savo duomenis, yra kompiuterio kietasis diskas (*angl. hard disc*). Kai kurie kietųjų diskų gamintojai (*Fujitsu, Hitachi, Seagate, Samsung*) jau

siūlo vartotojams savo produktus su aparatiniame lygmenyje realizuotu duomenų šifravimu. Dažniausiai siūlomi modeliai skirti nešiojamiems kompiuteriams. Naudojantis tokiu disku paprastai turi būti aktyvuotas BIOS slaptažodis, kuris ir įgalina priėjimą prie šifruotų duomenų [8, 9, 10]. Tačiau egzistuoja ir alternatyvių technologijų, o viena iš jų – gamintojo *LaCie* siūloma *LaCie SAFE* kietųjų diskų serija, kurioje naudojamas aparatinis šifravimas, o prieiga prie šifravimo raktų valdoma integruoto pirštų atspaudų skaitytuvo pagalba. [11]

1.2.4. Kitos priemonės - *IronKey*

IronKey – viena iš šiuo metu rinkoje siūlomų *USB* atmintinių, tenkinanti net karinius saugos reikalavimus. Tai įrenginys su aparatūriniu šifravimo mechanizmu ir patikima identifikacijos sistema. Duomenys *IronKey* atmintinėje visuomet laikomi užšifruoti, jų atrakinimui naudojamas slaptažodis, kuris patikrinamas ir patvirtinamas integruoto aparatinio modulio.

Pirmą kartą prijungus įrenginį prie kompiuterio, paprašoma įvesti slaptažodį, kuriuo naudojantis vėliau bus galima prieiti prie užšifruotų duomenų. Tai padarius sugeneruojami šifravimo raktai bei sukuriama šifruota failų sistema. Reikia paminėti, jog šis raktas veikia tik Windows aplinkoje. [12]

Tokius produktus kaip *IronKey* gamina ir kitos kompanijos (*Kingston*, *Corsair*, *Lexar* ir kt.), jų visų veikimo principas iš esmės toks pat. Beje praktiškai visos duomenų šifravimu aprūpintos atmintinės veikia išskirtiniai tik *Windows* operacinėse sistemose.

1.2.5. Esamų sistemų apibendrinimas

Taigi egzistuoja įvairių paprastiems vartotojams prieinamų ir duomenų saugumą užtikrinančių sistemų. Vienas jų galime gauti kartu su asmeninio kompiuterio operacine sistema, kitas įsigyti atskirai. Visgi nė viena sistema nėra tobula ir kiekvienoje jų galima aptikti didesnių ar mažesnių trūkumų. Aptartų priemonių apibendrintas palyginimas pateiktas *1 lentelėje*.

Žvelgiant į pateiktą lentelę ir grįžtant prie darbo pradžioje iškelto klausimo galima pastebėti, jog visose aptartose priemonėse pagrindinė problema išlieka: šifravimo raktai arba saugomi sistemoje (kas suteikia bent teorinę galimybę jį sužinoti pašaliniams asmenims), arba jie paliekami vartotojo žinioje (reikia juos atsiminti). Dažniausiai naudojama tokia schema: vartotojas įveda slaptažodį ir juo atrakina šifravimo/dešifravimo raktą.

Visos aptartos priemonės būtų saugesnės (ir patikimesnės) jeigu nei jos pačios žinotų, koks šifravimo raktas naudojamas (raktas nebūtų nesaugomas), nei verstų vartotoją jį pri-

siminti. Taigi tolesniame skyrelyje pabandykime pasvarstyti kokiomis priemonėmis būtų galima išvengti šių problemų ir kaip gauti slaptą raktą, kurio nereikėtų nei atsiminti, nei saugoti.

1 lentelė. Šifravimo technologijų (priemonių) palyginimas

Technologija (produktas)	*Veikia operacinėje sistemoje*	*Šifravimui naudojamas raktas*	*Šifravimo raktas saugomas*	*Šifravimo rakto apsaugos priemonės*
EFS	Windows (nuo 2000)	Simetrinis, sugeneruotas OS	+	Vartotojo slaptažodis
BitLocker	Windows (nuo Vista)	Simetrinis, sugeneruotas OS	+	TPM; „atrakinantis“ failas išorinėje laikmenoje; slaptažodis
FileVault	Mac OS X (nuo 10.3)	Simetrinis, sugeneruotas iš vartotojo slaptažodžio	+	Pagrindinis (*angl. Master*) slaptažodis
EncFS	Linux	Simetrinis, sugeneruotas OS	+	Slaptažodis
eCryptfs	Linux	Simetrinis, sugeneruotas OS	+	Slaptažodis
TrueCrypt	Windows, Mac OS X, Linux	Simetrinis, sugeneruotas programos	+	Slaptažodis ir raktinis failas
TPM	Windows, Mac OS X, Linux	-	+	Slaptažodis
Šifravimas kietajame diske	Windows, Mac OS X, Linux	Simetrinis, sugeneruotas aparatinio modulio	+	Slaptažodis (BIOS); piršto atspaudas
IronKey	Windows	Simetrinis, sugeneruojamas aparatinio modulio	+	Slaptažodis

Kadangi problemą iškėlėme tik raktų valdymo srityje, likusiose darbo dalyse kalbėsime apie šifravimo/dešifravimo raktus. Kai kurie teiginiai gali būti pritaikomi ir kitiems slaptiems raktams, tačiau likusios darbo dalys bus rašomos šifravimo raktų kontekste.

1.3. Problemos sprendimo metodų analizė

Praeitame skyrelyje apžvelgėme duomenų šifravimui siūlomus produktus, kuriuos galima naudoti jau šiandien. Toliau pabandykime panagrinėti kokiu būdu galėtume gauti šifravimo raktus ir išvengti iškeltų egzistuojančių problemų.

Problemos nebūtų jeigu kiekvieną kartą prireikus raktas būtų dinamiškai sugeneruojamas. Tačiau pagrindinis tokio proceso reikalavimas yra tas, jog kiekvieno generavimo metu galutinis rezultatas būtų toks pat. Tokiu atveju pradiniai generatoriaus parametrai visuomet privalo būti identiški. Taigi iš kur gauti šiuos parametrus, kad jų nereikėtų nei atsiminti, nei saugoti ir kiekvieną kartą jie būtų vis tokie patys? Vienas iš galimų atsakymų – *biometrija*.

Biometrija – tai technologija, identifikuojanti žmogų pagal jo fizines ar elgsenos savybes. Šios savybės gali būti: rankos geometrija, veidas, piršto atspaudas, balsas, rašysena ir kt. Ši technologija laikoma labai perspektyvia (o kai kurie metodai jau senokai taikomi praktikoje) ir po truputį išstumia kitas saugumo priemones (srityse, kuriose verta tai daryti). Kadangi biometrija yra aktyviai nagrinėjama ir tobulinama sritis, o žmonių biometrinės charakteristikos unikalios, ja remdamiesi ir panagrinėkime kaip būtų galima išspręsti šiame darbe iškeltas problemas.

1.3.1. Biometrinių duomenų panaudojimas saugumo sistemose

Biometrinės technologijos vystosi labai sparčiai ir pagal įvairių autorių prognozes greitu laiku pakeis šiuo metu vis dar populiarius, tačiau daug saugumo problemų keliančius slaptažodžius. Spartesniam biometrinių sistemų skverbimuisi į rinką trukdo keletas dalykų:

- vis dar nėra sudaryta bendrųjų biometrinių duomenų standartų
- ne kiekvienas vartotojas yra linkęs pateikti sistemai savo duomenis

Sritis, kur šios technologijos taikomos jau ilgą laiką – teisė ir visa teisinė sistema. Biometrija čia pasirinkta neatsitiktinai: kiekvieno žmogaus fiziologinės charakteristikos yra unikalios ir būtent tai suteikia sistemai didelį patikimumą.

Kiekvienai naudojamai biometrinei charakteristikai nuskaityti naudojami biometrinių duomenų skaitytuvai, kurie gali dirbti dviem rėžimais:

- 1:1, t.y. tapatybės patvirtinimo (*angl. verification*)
- 1:N, t.y. identifikavimo (*angl. identification*)

Pirmuoju atveju pateiktas duomenų pavyzdys lyginamas tik su vienu duomenų bazėje esančiu įrašu, siekiant įsitikinti, jog sistema bando naudotis tikrai tas asmuo, kuriuo vartotojas prisistato. Šis metodas dažniausiai naudojamas asmeninėse vieno žmogaus sistemose ir pakeičia standartinį slaptažodžio įvedimą. Prieš pateikiant biometrinę charakteristiką, paprastai tenka įvesti vartotojo vardą ar kitą identifikuojančią informaciją, pagal kurią sistema iš

karto atrenka biometrinių duomenų pavyzdį iš saugyklos, kuris bus lyginamas su naujai pateikiamu.

Antruoju atveju pateiktas duomenų pavyzdys lyginamas su visais duomenų bazėje saugomais biometrinių charakteristikų pavyzdžiais. Tuo pirmiausia siekiama išsiaiškinti, ar bandantis prisijungti asmuo apskritai yra registruotas sistemoje ir jeigu taip, bandoma nustatyti jo tapatybę (identifikuoti). Šis metodas naudojamas masinio prisijungimo (daugelio vartotojų) sistemose. Šiuo atveju jau nereikalaujama įvesti jokios papildomos identifikavimo informacijos - pakanka tik pateikto biometrinio pavyzdžio. Vartotojams tokios sistemos naudojimas paprastesnis, tačiau sisteminiu požiūriu šis metodas yra kur kas sudėtingesnis, kadangi reikalingi galingesni ir sudėtingesni algoritmai.

Visose biometrija pagrįstose sistemose egzistuoja viena didelė problema, kuri nusakoma dviem sąvokomis:

- *FRR* (*angl. False Rejection Rate*) – klaidingo atmetimo rodiklis. Jis nusako tikimybę, kad sistemoje registruotas vartotojas bus neatpažintas identifikavimo proceso metu
- *FAR* (*angl. False Acceptance Rate*) – klaidingo priėmimo rodiklis, nusakantis sistemoje neregistruoto vartotojo pripažinimo (priėmimo) tikimybę

Biometrijoje nėra visiško duomenų atitikimo. Tai reiškia, jog kiekvienas naujas to paties vartotojo biometrinių duomenų pavyzdys yra vis kitoks. Todėl biometrinės sistemos negali kiekvieną kartą reikalauti 100% atitikimo šablonui (pradiniam pavyzdžiui). Taigi viena iš esminių biometrinių saugos priemonių problemų – teisingai ir protingai sudėlioti *FRR* ir *FAR* slenksčius, kad tikimybės, jog neregistruotas vartotojas bus priimtas arba registruotas vartotojas bus atmestas, būtų mažiausios. [13]

Šių rodiklių reikšmės priklauso ir nuo pasirinktos charakteristikos tipo. Bet kokiu atveju šioje srityje naudojami sudėtingi algoritmai, padedantys sumažinti klaidų skaičių.

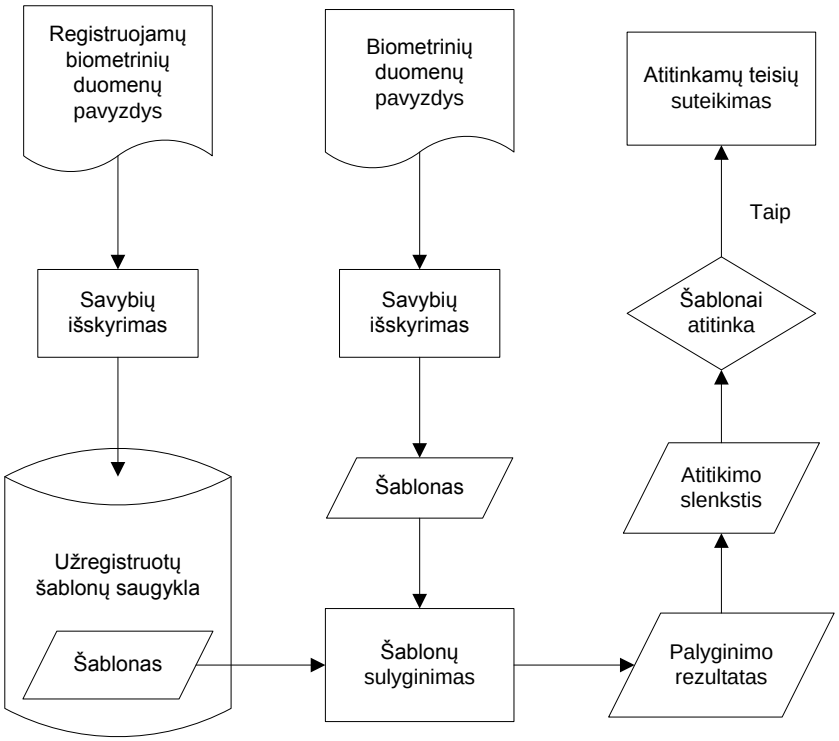
Toliau apžvelkime būdus ir principus kaip biometrija yra ir gali būti pritaikoma saugumo sistemoms.

1.3.2. Tradicinė biometrija

Tradicinė biometrija pradėta taikyti seniausiai ir visų biometrinių saugumo priemonių. Prieš aktyvuojant tokią sistemą, pirmiausia reikia užregistruoti ja besinaudosiančius as-

menis. Tam paprasčiausiai nuskaitomos registruojamų asmenų biometrinės charakteristikos ir patalpinamos sistemos duomenų bazėje.

Asmens identifikavimo procesas taip pat nesudėtingas: pirmiausia nuskaitomas biometrinių duomenų pavyzdys, tuomet duomenų bazėje ieškoma pateikto pavyzdžio atitikmens. Jį radus suteikiamos atitinkamos teisės (leidimas įeiti, prieiga prie duomenų, atitinkamas slaptas raktas ar pan.) [13]. Bendroji tokios sistemos veikimo schema parodyta *1 pav.*



1 pav. Tradicinė biometrinė saugumo sistema

Apskritai tradicinės biometrinės sistemos sprendimas yra paprastas. Sudėtingiausia jo vieta – biometrinių pavyzdžių sulyginimas ir teisingas asmens identifikavimas (tokios sistemos paprastai ir naudojamos identifikavimo, t.y. 1:1 rėžimu). Nepaisant sprendimo paprastumo yra ir nepageidautinų savybių. Viena pagrindinių - sistemos duomenų bazėje biometriniai pavyzdžiai paprastai laikomi atviru formatu, todėl pažeidimo atveju piktavalis galės be problemų pasinaudoti gautais duomenimis.

1.3.3. Rakto išskaičiavimas iš biometrinės charakteristikos

Šio metodo esmė – vartotojo registracijos metu sugeneruoti ir priskirti jam slaptą raktą. Tai atlieka algoritmas, apdorojantis atitinkamų biometrinių duomenų pavyzdžio sričių

bitus. Esminė šiuo principu veikiančios sistemos vieta – rakto generavime naudojamų bitų išskyrimas, tačiau čia slypi ir pagrindinis trūkumas. Paprastai rakto generavime naudojamų bitų vieta biometrinių duomenų pavyzdyje apibrėžiama statiškai, t.y. piktavaliui pakanka išsiaiškinti, kurie bitai įtraukiami į rakto generavimą ir iš sistemos išgavęs konkretaus asmens biometrinį pavyzdį galės pats susigeneruoti vartotojui priskirtą raktą. Taip pat turint skirtingų biometrinių pavyzdžių rinkinį įmanoma išsiaiškinti, kurios sritys yra naudojamos generavime. [14]

Rakto išdavimo metu tokio tipo sistema veikia dviem etapais: pirmiausia vartotojas identifikuojamas pagal tradicinės biometrijos principus. Tuomet iš sistemoje saugomo (nuskaityto vartotojo registracijos metu) biometrinių duomenų pavyzdžio sugeneruojamas konkretus raktas, kuris atiduodamas vartotojui tolesniam naudojimui.

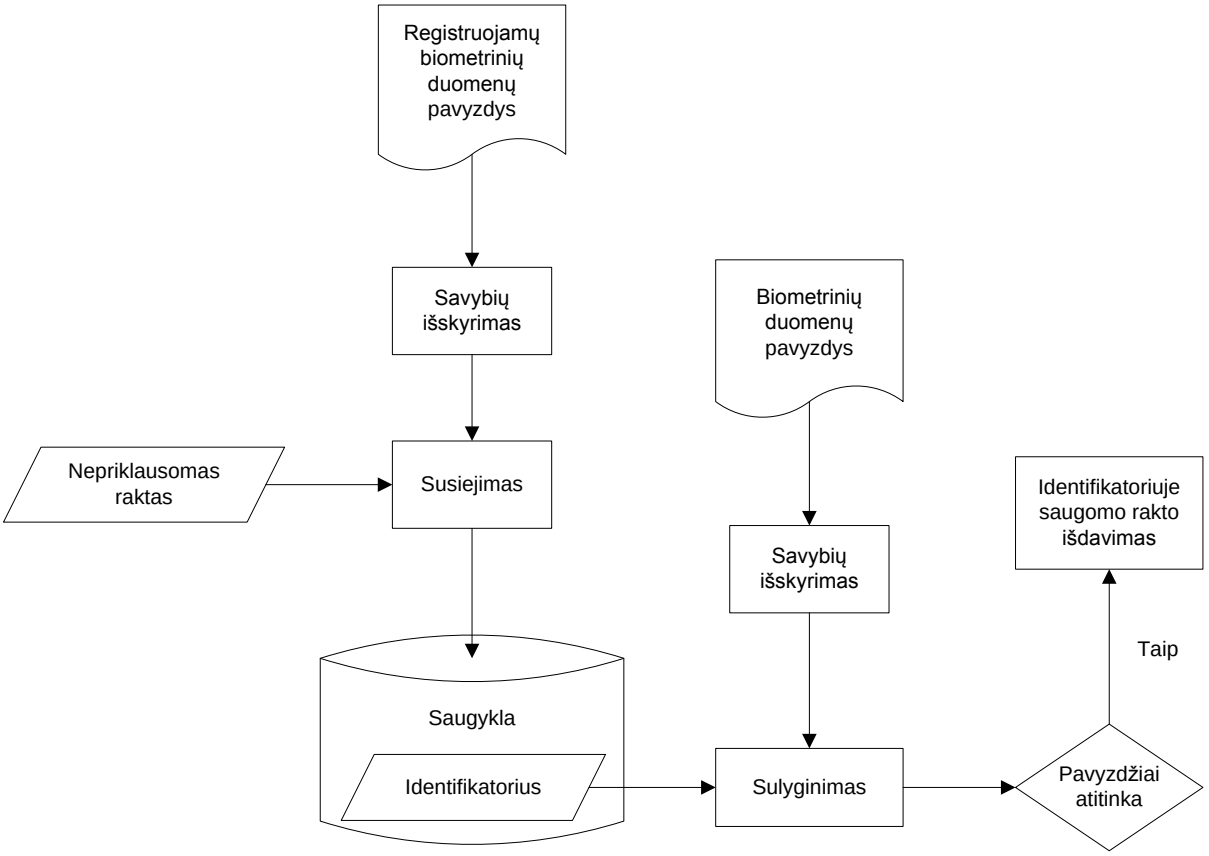
Kartais naudojama ir šio metodo modifikacija: raktas generuojamas ne iš sistemoje saugomo vartotojo biometrinių duomenų pavyzdžio, bet iš pateikto realiu laiku. Tačiau čia susiduriama su kita problema: reikia atrinkti patikimas sritis (naudojamas rakto generavime) arba kitais būdais užtikrinti, jog kiekvieną kartą vartotojui būtų atiduodamas toks pat raktas. Nepriklausomai nuo to, kuri realizacija pasirenkama, vykdant vartotojų identifikaciją visvien reikia turėti atitinkamą duomenų saugyklą, kurioje būtų saugomi biometriniai duomenų pavyzdžiai, pateikti vartotojo registracijos metu, ir kurie būtų lyginami su pateikiamais realiu laiku taip realizuojant vartotojų identifikaciją.

1.3.4. Biometrinis šifravimas

Biometrinis šifravimas (*angl. Biometric Encryption - BE*) – tai technologija, žmogaus biometrinius duomenis naudojanti apsaugant įvairius PIN kodus, slaptažodžius ar kitus slaptus simbolių rinkinius, prieigos teisėms ir kitiems leidimams suteikti. Kaip ir anksčiau aptartais atvejais, prieš pradedant naudotis biometrinio šifravimo sistema taip pat būtina registracija, kurios metu nuskaityta biometrinė charakteristika pagal tam tikrą algoritmą susiejama su nepriklausomai sugeneruotu raktu, kuris ir yra tikrasis šifravimo/dešifravimo raktas. Algoritmo rezultatas - identifikatorius (*angl. BE template*) - išsaugomas tam skirtoje duomenų bazėje (*2 pav.*). [15, 16]

Norint pasinaudoti suteiktu slaptu raktu, pagal naujai nuskaitytą biometrinį pavyzdį iš duomenų bazės atrenkamas atitinkamas identifikatorius, iš kurio, naudojant specialų algoritmą, sugeneruojamas konkretus raktas. Pagrindinis šios sistemos uždavinys – pagal pateiktą biometrinę charakteristiką teisingai atrinkti vartotojo identifikatorių. Kadangi pats raktas ge-

neruojamas tik iš registracijos metu suformuoto identifikatoriaus, kiekvienu atveju gaunamas identiškas raktas.



2 pav. Biometrinio šifravimo sistema

Būtina pabrėžti, jog šiame procese informacija sunaikinama vos tik ji būna panaudota ir nebereikalinga, t.y. nei biometrinės charakteristikos pavyzdys, nei jai priskirtas raktas sistemoje nesaugomi. Be to neturint reikiamo biometrinio pavyzdžio iš identifikatoriaus neįmanoma sugeneruoti teisingo rako. Žinant, jog kiekvienas naujas biometrinių duomenų pavyzdys nėra visiškai identiškas prieš tai buvusiems, tokios sistemos taip pat realizuojamos pakankamai sudėtingai.

Apskritai biometrinio šifravimo sistemos laikomos pakankamai saugiomis ir patikimomis. Nepaisant to, jos visvien neišsprendžia iškeltos problemos – sistemoje turi būti saugomi atitinkami duomenys, šiuo atveju identifikatoriai. Nors aptartasis metodas ir leidžia reikalingą duomenų kiekį sumažinti iki minimumo, tačiau atitinkama saugykla reikalinga. Galbūt palyginus su technologijos privalumais tai nėra didelis trūkumas, tačiau, šioje sistemoje esama problema – duomenų saugojimas - išlieka.

1.3.5. Tiesioginis biometrinių duomenų panaudojimas

Vienas iš tokio tipo metodų buvo užpatentuotas Vokietijoje 1994 metais [17] ir aprašo būdą, kaip biometriniai duomenys gali būti tiesiogiai panaudoti vietoj šifravimo rakto.

Taikant autoriaus siūlomą metodą raktas išskiriamas tiesiogiai iš biometrinės charakteristikos pavyzdžio. Nors tai atrodytų pakankamai saugu ir patikima, visgi pripažįstama, jog metodas turi du didelius trūkumus:

- kiekvienas naujas biometrinių duomenų pavyzdys yra vis kitoks, todėl sunku užtikrinti, jog kiekvieną kartą bus sugeneruojamas toks pat raktas
- iš vieno pavyzdžio gali būti sugeneruotas vienintelis raktas, t.y. jeigu jis nutekės tretiesiems asmenims, biometrinei charakteristikai nebus galima priskirti naujo rakto

Pirmasis trūkumas galioja absoliučiai visoms biometrija besiremiančioms sistemoms, tačiau šio metodo kontekste jis turi ypatingą reikšmę. Antrasis yra taikomas tik šiam konkrečiam metodui, kas smarkiai sumenkina jo patikimumą ir praktinį pritaikomumą.

Apskritai šio metodo realizacija yra pakankamai sudėtinga, kadangi siekiant jog tam pačiam žmogui kiekvieną kartą būtų sugeneruojamas tas pats raktas, reikalinga parinkti stabilius charakteristikos taškus arba naudoti patikimus ir nesutapimams atsparius metodus. Tai ko gero viena pagrindinių priežasčių, kodėl šio metodo praktinis pritaikymas yra pakankamai sudėtingas.

1.3.6. Biometrijos naudojimo raktų valdyme apibendrinimas

Šiame skyrelyje buvo paminėti pagrindiniai raktų valdyme naudojami biometriniai metodai. Nė vienas jų nėra tobulas: vieniems galioja bendrosios biometrinių sistemų problemos, kiti turi dar ir savų unikalių. Viena iš darbo pradžioje aptartų problemų šiuo atveju yra išspręsta – vartotojui nebereikia atsiminti jokių slaptažodžių, nes jo slaptažodis yra jo biometrinė charakteristika. Tačiau kita problema – papildomų duomenų saugyklų naudojimas – visgi išlieka.

2 lentelėje pateiktas nagrinėtų biometrinių saugumo sistemų palyginimas, iš kurio matome, jog biometrija naudojama dviem tikslams: prieigai prie raktų saugyklos arba raktų generavimui. Nors pastarais variantas nėra paplitęs, tačiau visgi buvo bandymų tai daryti.

2 lentelė. Biometrinių saugumo sistemų palyginimas

Veikimo principas	*Paskirtis*	*Reikalinga registracija*	*Saugomi tarpiniai duomenys*	*Pastabos*
Tradicinė biometrija	Prieigos teisėms suteikti	+	+	Veikia identifikavimo ir(arba) autentifikavimo rėžimais
Rakto išskaičiavimas	Prieiga prie slapto rakto	+	+	Rakto reikšmę biometriniai duomenys įtakoja
Biometrinis šifravimas	Prieiga prie slapto rakto	+	+	Rakto reikšmės biometriniai duomenys neįtakoja
Tiesioginis biometrinių duomenų panaudojimas	Rakto generavimui	-	-	Priskirto rakto pakeisti negalima

Biometrinių sistemų patikimumą (o tuo pačiu ir sudėtingumą) galima didinti naudojant ne vieną, o kelias biometrines charakteristikas. Tokiu atveju naudojant atitinkamą algoritmą suformuojamas duomenų rinkinys, kuris toliau gali būti laikomas kaip viena biometrinė charakteristika. [18, 26]

Taigi biometriniai duomenys gali būti puikiai panaudoti slaptų raktų generavime. Pagrindinis tokių sistemų privalumas – vartotojas tokio rakto niekuomet nepames, tačiau svarbiausia – jo nereikia atsiminti. Vis dėl to būtina tobulinti metodus, kurie užtikrintų patikimą raktų generavimą bei atsparumą nedideliems biometrinių duomenų nesutapimams.

Vieną iš iškeltų problemų galima laikyti išspręsta: naudojant biometriją išvengiama būtinybės atsiminti slaptažodį. Beliko viena problema – tarpinių (ir ne tik) duomenų saugojimas. Biometrinės sistemos apskritai dar turi kur tobulėti, o jų pritaikymo galimybės taip pat neišsemtos. Kadangi biometrinius duomenis turi visi žmonės, o jų panaudojimas (ypač atrinkus vartotojui draugišką (*angl. user friendly*) charakteristiką) didesnių nepatogumų taip pat nesukelia, likusias darbo dalis nagrinėsime laikydami, jog slapto rakto generavimui bus naudojama biometrinė charakteristika.

1.4. Biometrinių charakteristikų apžvalga

Toliau trumpai aptarsime pagrindinių šiuo metu naudojamų biometrinių charakteristikų savybes, privalumus ir trūkumus.

1.4.1. Pirštų atspaudai

Pirštų atspaudų skaitymas yra pats populiariausias biometrijos pritaikymo būdas. Tokios sistemos lygina pateiktus pirštų atspaudus su sistemoje saugomais atitikmenimis. Sukurta įvairių nuskaitymo ir palyginimo technologijų, tačiau vartotojams jų specifika nematoma.

Privalumai:

- Paprastas būdas – vartotojų nereikia mokyti naudotis, viskas pavyksta natūraliai
- Naudojami jutikliai yra maži, naudoja nedaug energijos, todėl gali būti įrengiami praktiškai bet kur
- Tai seniausiai taikomas ir geriausiai išnagrinėtas bei paprasčiausiai realizuojamas būdas – rinkoje vartotojui siūloma daug produktų
- Pirštų atspaudų biometrija tapo kai kurių šalių (pvz. JAV) valstybinių įstaigų standartu

Trūkumai:

- Nešvarių ar pažeistų pirštų atspaudai gali būti neatpažįstami
- Kadangi būdas nėra 100% patikimas, dažnai derinamas su kitais (pvz. PIN kodo įvedimu)
- Dėl taikymo kriminalinėse institucijose, vartotojai jaučia psichologinį diskomfortą atspaudo nuskaitymo metu

Šiandien pirštų atspaudų skaitymas tampa vis populiaresnis daugelyje IT sistemų ir produktų. Tačiau norint šią charakteristiką naudoti fizinę prieigą suteikiančiose sistemose, reikia tobulinti esamus metodus, kadangi pastarieji ne visuomet užtikrina reikiamą saugumo lygį.

1.4.2. Veido atpažinimas

Šio būdo esmė – vartotojo 2D ar 3D veido atvaizdas lyginamas su esančiais sistemoje pavyzdžiais. Paprastai taikomi 2D būdai, tačiau pradeda sparčiau populiarėti ir trimačiai.

Privalumai:

- Būdas gali būti naudojamas praktiškai bet kur, kur įmanoma panaudoti vaizdo kamerą

- Veikia per nuotolį, vartotojui pačiam nereikia nieko daryti
- Nauji 3D metodai suteikia ypatingą tikslumą ir patikimumą

Trūkumai:

- 2D sistemoms keblumų gali kelti akiniai, skrybėlės ar kitokie veido pakeitimai - netgi veido pasukimas kitu kampu
- Nors 3D metodai daug tikslesni ir patikimesni, jie vis dar kūrimo stadijoje ir, be abejo, yra (ir bus) žymiai lėtesni - šiuo atžvilgiu pirmauja 2D sistemos
- Paprastai žmonės nemėgsta būti stebimi, todėl kartais manoma, kad vaizdo kamerų naudojimas pažeidžia jų privatumą

Pakankamai didelis netikslumas neleidžia šiam būdui plisti, todėl jis dažniausiai naudojamas kaip pagalbinė priemonė asmenų atpažinimui.

1.4.3. Rankos geometrija

Rankos geometrijos skaitytuvai analizuoja ir lygina delno struktūrą ir kampus. Vartotojui tereikia uždėti delną ant metalinės plokštelės, kurios pagalba nuskaitoma charakteristika.

Privalumai:

- Naudojami skaitytuvai tvirti ir ilgaamžiai, atsparūs nepalankioms išorės sąlygoms
- Paprastas ir intuityvus naudojimas
- Saugomų ir palyginimui naudojamų duomenų kiekis minimalus (apie 20B vienam pavyzdžiui)

Trūkumai:

- Skaitytuvai dideli, užima nemažai vietos, todėl reikalinga iš anksto numatyti ir paskirti jiems vietą
- Būdas gana nepatikimas, kadangi rankos geometrija nėra unikalus kiekvieno žmogaus požymis
- Kai kurių žmonių netenkina sanitarinės sąlygos: jie nenori dėti savo delno ten, kur prieš tai dėjo neaišku kiek ir kokių kitų žmonių

Nors ir dideli, tačiau greiti skaitytuvai gali būti naudojama tose vietose, kur jiems numatyta pakankamai erdvės, tačiau visgi pagrindinę problemą kelia jų patikimumas.

1.4.4. Akies biometrija

Akies obuolyje yra nemažai požymių, galinčių pakankamai tiksliai identifikuoti asmenį. Dažniausiai naudojamos dvi akies biometrinės charakteristikos: tinklainė arba rainelė. Kartais pasitaiko sistemų, kuriose kartu naudojamos abi charakteristikos.

Privalumai:

- Šis būdas yra labai tikslus ir patikimas
- Naudojamasi per atstumą – tarsi žiūrint į veidrodį
- Žmogus su sveiku ir nepažeistu akies obuoliu atpažįstamas praktiškai kiekvieną kartą

Trūkumai:

- Būdas nėra labai intuityvus, vartotojui reikia įgusti juo naudotis
- Naudojamas papildomas akies apšvietimas, kas nėra labai malonu
- Akies biometrija paremtų sistemų vystymas buvo pristabdytas dėl patentų, tačiau tikimasi, jog situacija greitai turėtų pasikeisti
- Reikalinga sudėtinga įranga, todėl būdas pakankamai brangus

Akies biometrijos sistemos yra išties perspektyvios ir daug žadančios. Didesnis jų populiarumas turėtų prasidėti baigiant galioti atitinkamiems patentams, todėl kelerių metų bėgyje galima sulaukti ir labai rimtų sistemų.

1.4.5. Biometrinių charakteristikų apibendrinimas

Šiame skyrelyje paminėtos tik populiariausios ir dažniausiai taikomos biometrinės charakteristikos. Jų, be abejo, jų yra žymiai daugiau, pvz. parašo skenavimas, elgsenos ypatybių, balso analizavimas, kitų įvairių kūno vietų geometrija, kraujagyslių tinklo analizė ir kt. [19]

Kai kurios paminėtos technologijos gali būti puikus pakaitalas ar net geresnė alternatyva šiuo metu naudojamoms. Tačiau kadangi dauguma jų dar tik kūrimo stadijoje, nėra rimtų, naudojimui paruoštų produktų, todėl tenka naudoti tai, kas jau išbandyta ir patikrinta.

Biometrinių charakteristikų palyginimo apibendrinimas pateiktas *3 lentelėje.* Tikslumas ir sudėtingumas įvertinti renkantis iš trijų galimų pasirinkimų, t.y. *mažas, vidutinis, didelis.*

3 lentelė. Biometrinių charakteristikų palyginimas

Charakteristika	*Tinka identifikavimui*	*Tinka autentifikavimui*	*Tikslumas*	*Sudėtingumas*	*Maža kaina*
Piršto atspaudas	+	+	Didelis	Mažas	+
Veido atpažinimas	-	+	Vidutinis	Vidutinis	+
Rankos geometrija	-	+	Vidutinis	Mažas	-
Akies tinklainė	+	+	Didelis	Didelis	-
Akies rainelė	+	+	Didelis	Vidutinis	-
Parašo atpažinimas	-	+	Mažas	Mažas	+
Balso atpažinimas	-	+	Mažas	Mažas	+

Iš šiandien naudojamų technologijų populiariausia yra pirštų atspaudų skaitymas. Populiarumą lėmė paprastumas, didelis pritaikomumas, pasiteisinęs naudojimas kriminalinėse institucijose, maži skaitytuvų gabaritai ir nesudėtinga realizacija – tai matyti ir *3 lentelėje.* Kadangi ši technologija yra geriausiai žinoma ir atidirbta, būtent ją ir pasitelksime tolesnėse darbo dalyse.

1.5. Išvados

Išanalizavus duomenų šifravimo priemonių pasiūlą nustatyta, jog tokių priemonių yra daug ir įvairių. Vienos pateikiamos kartu su operacinėmis sistemomis, kitas (tiek aparatines, tiek programines) galima įsigyti atskirai.

Esminė ir daugiausia problemų kelianti šifravimo sistemų dalis – slaptų raktų valdymas. Geriausiai žinoma ir plačiausiai naudojama šifravimo rakto forma – slaptažodis.

Populiariausi ir dažniausiai naudojami raktų valdymo būdai turi du pagrindinius trūkumus: vartotojui reikia atsiminti slaptą simbolių seką ir (arba) ją kažkur saugoti. Įprastas scenarijus: vartotojas įveda įsiminta slaptažodį, kuriuo atrakina sistemoje saugomą slaptą raktą.

Biometrinių charakteristikų panaudojimas supaprastina raktų valdymo problemą, kadangi yra charakteristikų, kurios gali būti patikimai naudojamos žmogaus identifikavimui, taip jį išlaisvinant nuo būtinybės atsiminti slaptažodžius.

Praktikoje taikomi biometriniai metodai paprastai veikia dviem etapais: pirmiausia identifikuoja vartotoją, tuomet išduoda sistemos jam priskirtą ir duomenų bazėje saugomą raktą.

Bet kuris paminėtas slaptas raktas gali būti laikomas ir šifravimo raktu, todėl visi skyriuje paminėti teiginiai tinka ir šifravimo raktams.

2. ŠIFRAVIMO RAKTO GENERAVIMO IŠ PIRŠTO ATSPAUDO METODAS

2.1. Darbo tikslas ir uždaviniai

Taigi išnagrinėjus duomenų šifravimui skirtus produktus, dar kartą įsitikinta, jog sudėtingiausia ir daugiausia dėmesio reikalaujanti tokių sistemų dalis – šifravimo raktų valdymas. Paprastai šifravimo sistemose vienam ar kitam tikslui naudojamas slaptažodis, kurį vartotojas turi atsiminti. Kaip jau išsiaiškinome - tai nėra nei patogu, nei praktiška, o ir sukelia daug papildomų saugumo problemų. Todėl reikia vengti būtinybės atsiminti bet kokią šifravimo sistemoje naudojamą informaciją, ypač šifravimo raktą.

Kita jau minėta problema – tarpinių duomenų saugojimas. Tais atvejais kai slaptažodis naudojamas tik šifravimo rakto išdavimui, pats raktas vienokia ar kitokia forma yra saugomas sistemoje, kas vėlgi dažniausiai sukelia papildomų saugumo problemų. Todėl bet kokių tarpinių duomenų saugojimo taip pat reiktų atsisakyti.

Kaip alternatyvą įvairiems slaptažodžiams ir kitokiems raktams galima naudoti žmogaus biometrinius duomenis. Aptarėme pagrindinius biometrijos pritaikymo duomenų šifravime būdus. Nė vienas iš jų neišsprendė iškeltų problemų: arba visvien reikėjo tarpinių duomenų saugyklų, kas komplikuoja realizacijos ir panaudojimo paprastumą (o taip pat ir saugumą), arba pats būdas yra pakankamai sudėtingas, kad būtų plačiau naudojamas.

Taip pat palyginome skirtingas biometrines charakteristikas ir išsiaiškinome, kad populiariausia ir lengviausiai realizuojama technologija – pirštų atspaudų skaitymas. Įvertinus technologijos privalumus ir trūkumus nuspręsta, jog būtent pirštų atspaudus galima panaudoti sprendžiant iškeltas problemas - kas ir bus daroma tolesnėse darbo dalyse. Susisteminus išanalizuotus šaltinius bei padarytas išvadas, galime apsibrėžti darbo tikslą ir jam pasiekti keliamus uždavinius.

2.1.1. Darbo tikslas

Sudaryti metodą, leidžiantį šifravimo raktą generuoti tiesiogiai iš piršto atspaudo.

2.1.2. Uždaviniai

Kadangi literatūra jau išnagrinėta, naudojami šifravimo metodai ir biometrinių charakteristikų pritaikymas aptarti, uždavinius formuluojame tolesnėms šio darbo dalims.

- Apsibrėžti sudaromo metodo taikymo sritį
- Apsibrėžti, kokius reikalavimus turi tenkinti sudaromo metodo realizacija
- Išsiaiškinti, kaip piršo atspaudas gali būti aprašomas tolesniam jo apdorojimui
- Sudaryti paprastą, lengvai realizuojamą, tačiau apsibrėžtus reikalavimui tenkinantį metodo matematinį pagrindą
- Programiniu būdu realizuoti sudarytą metodą eksperimentui atlikti
- Eksperimento metu nustatyti sudaryto metodo efektyvumą pagal užsiduotus kriterijus
- Pagal gautus rezultatus parašyti darbo išvadas

2.2. Taikymo sritis ir reikalavimų apibrėžimas

Prieš pradedant sudarinėti rakto generavimą iš piršto atspaudo metodą, būtina apsibrėžti metodo taikymo sritį ir pagal tai nustatyti jam keliamus reikalavimus. Tai padės išvengti darbo eigoje susidarančių neaiškių situacijų ir leis susikoncentruoti į esmines metodo savybes.

Taikymo sritį galima numanyti iš darbo pavadinimo ir analizės dalyje nagrinėtos problemos. Kadangi iki šiol darbe nagrinėjome šifravimo sistemas ir jų raktus, taip elgsimės ir toliau. Kertinė šifravimo sistemų dalis yra šifravimo/dešifravimo raktas, todėl mūsų iš biometrinių duomenų sugeneruotas raktas turėtų būti taikomas būtent duomenų šifravimui.

Rakto saugumas yra tiesiog proporcingas jo ilgiui ir skirtingų simbolių, iš kurių jis sudarytas, skaičiui. Vadinasi kuo ilgesnis ir kuo iš įvairesnių simbolių sudarytas raktas, tuo jis saugesnis. Iš biometrinės charakteristikos sudaryti ilgą ir sudėtingą raktą nėra lengvas uždavinys, juolab, kad dar nėra aišku, ar tai apskritai įmanoma pritaikyti praktikoje. Tolesnėse dalyse nesieksime nei rakto ilgio nei jo sudėtingumo rekordų, nes mūsų tikslas – patikrinti, ar iš piršto atspaudo sugeneruotas raktas apskritai gali būti panaudojamas duomenų saugumui užtikrinti.

Bene svarbiausias uždavinys – iš to paties piršto visuomet gauti tą pati raktą. Priešingu atveju užšifruotų duomenų negalės perskaityti net pats jų savininkas (užšifravęs asmuo). Žinoma būtų idealus variantas jeigu kiekvienam pirštui būtų sugeneruojamas vis kitoks raktas, tokiu atveju būtų užtikrinta, jog niekas negalėtų perskaityti svetimų duomenų.

Apsibrėžę taikymo sritį ir norimą gauti rezultatą, nustatykime pagrindinius reikalavimus, kuriuos turi tenkinti sudaromas metodas. Reikalavimų šioje dalyje labai nedetalizuojame – tai padarysime projektinėje dalyje. Taigi sudaromas metodas turi tenkinti šiuos reikalavimus:

- rakto generavimui turi būti naudojami keli atrinkti piršto atspaudo taškai, konkretus jų skaičius priklausys nuo metodo specifikos
- iš to paties piršto atspaudo kiekvieną kartą turi būti gaunamas toks pat raktas
- svarbi metodo sparta, todėl reikia vengti ypatingai sudėtingų iteracijų ir skaičiavimų
- atspaudo ar rakto „nutekėjimo“ klausimas nenagrinėjamas, t.y. galimybė tam pačiam atspaudui priskirti kitokį generuojamą raktą nėra būtina
- sugeneruotas raktas turi būti galutinis, t.y. jis iš karto (be papildomo apdorojimo) gali būti perduotas jo reikalaujančiai sistemai kaip šifravimo raktas
- metodui neturi reikėti jokių papildomai kaupiamų duomenų, viskas turi vykti realiu laiku, o po rezultato pateikimo visi naudoti duomenys privalo būti sunaikinti

Beje, atskirai verta pažymėti, jog klaidingas atmetimas (*FRR*) mūsų atveju yra daug blogiau nei klaidingas priėmimas (*FAR*). Daug svarbiau, kad duomenis savo piršto atspaudu užšifravęs asmuo juos vėl galėtų perskaityti, priešingu atveju duomenys bus nebeatstatomi, iš jų neliks jokios naudos ir tokios sistemos panaudojimas bus bevertis. Apskritai kol nebus atlikti dideli ir patikimi eksperimentai bei ištobulintas pats metodas, tokiu būdu sugeneruotą raktą vertėtų naudoti nebent nelabai svarbiems duomenims arba jų kopijoms šifruoti. Būtent dėl šių aplinkybių galima teigti, jog galimybė perskaityti ne savo užšifruotus duomenis šiuo atveju yra mažesnė blogybė nei negalėjimas to padaryti su „teisingo“ piršto atspaudu.

2.3. Piršto atspaudo aprašymas kompiuteryje

Norint generuoti šifravimo raktą iš piršto atspaudo, pirmiausia atspaudą reikia apsirašyti patogia ir suprantama forma. Tai galioja ne tik nagrinėjamu atveju – kiekviena biometrinė charakteristika turi būti vienaip ar kitaip aprašyta, kad ją būtų galima toliau apdoroti matematiniais (ir ne tik) metodais.

Kalbant apie pirštų atspaudus - kiekvienas žmogaus pirštas išvagotas įvairių griovelių ir iškilimų, kurių visuma ir sudaro atspaudą. Manoma, kad pasaulyje nėra dviejų vienodų

pirštų atspaudų, jų nepasitaikė ir nuo 1924 metų FTB kaupiamoje duomenų bazėje. Šis faktas tik patvirtina teiginį, jog pirštų atspaudai yra unikali asmens identifikavimo priemonė.

2.3.1. Piršto atspaudą identifikuojančios charakteristikos

Pirštų atspaudų rašto požymiai standartiškai skirstomi į tris lygius:

- I lygis – iškilios linijos, matomos plika akimi
- II lygis – taškai, kuriuose linijos išsišakoja (susijungia) arba prasideda (pasibaigia)
- III lygis – iškilimų matmenys, forma, kryptis ir pan.

Kiekvieno lygio požymiai aprašomi tam lygiui būdingais parametrais, kurių atskirai neaptarinėsime. Pakankamai tiksliam žmogaus identifikavimui pilnai užtenka I ir II lygio požymių. Kompiuterizuotose sistemose paprastai naudojamas II lygis - taškai, vadinami *minutiae* taškais arba *Galtono* charakteristikomis, kurie pakankamai nesudėtingai aprašomi. *Minutiae* tašką paprasčiausiu atveju apibūdina jo koordinatės ir tipas (susikirtimas, nutrūkimas).

Piršto atspaudo aprašymui patogu naudoti matricas, ypač jeigu nuskaitytas atspaudas turi būti perduotas tolesniam kompiuteriniam apdorojimui. Tokia matrica sudaroma nuskaityto atspaudo atvaizdą apdorojant specialiais algoritmais. Standartiškai tai vyksta tokia eilės tvarka:

- piršto atspaudo atvaizdo paruošimas (išryškinimas, skaitmenizavimas, segmentavimas)
- *minutiae* taškų išskyrimas (linijų ploninimas iki 1 pikselio pločio, pačių taškų radimas)
- papildomos operacijos (netikrų taškų pašalinimas ir kt.) [20, 21]

Kadangi *minutiae* taškų naudojimas pirštų atspaudų aprašymui ir palyginimui yra pakankamai tikslus, nesudėtingai realizuojamas ir vienas plačiausiai naudojamų ir literatūroje aprašytų būdų, sudaromas metodas šifravimo raktą generuos remdamasis būtent šiomis charakteristikomis.

2.3.2. Piršto atspaudo aprašymo principas sudaromame metode

Taigi išsiaiškinome, jog *minutiae* taškus galima aprašyti koordinačių matrica. Kaip jau minėjome kiekvienas taškas priklauso ir vienam iš dviejų tipų, t.y. susikirtimas ir nutrū-

kimas. Šio darbo sudaromame metode taško tipo nevertinsime - tą paliksime ateities darbams, o mums pakaks tik matricos, nurodančios *minutiae* taškų vietas.

Teisingas piršto atspaudo apdorojimas ir matricos sudarymo technologija yra pakankamai sudėtingi dalykai, apie juos galima rasti daug įvairios literatūros. Kadangi ši sritis verta atskiro magistro darbo, matricos sudarymo principų nenagrinėsime, tik apsibrėšime kokią matricą naudosime savo metode. Taigi, sudaromam metodui pradiniai duomenys bus paduodami *minutiae* taškų koordinačių matrica, kuri sudaryta tokiu principu:

- kiekvienas matricos elementas nusako vieną piršto atspaudo atvaizdo pikselį
- jeigu konkrečioje vietoje yra *minutiae* taškas (kaip jau apsibrėžėme - nesvarbu kokio tipo), matricos elementas lygus *1*, priešingu atveju lygus *0*

2.4. Metodo esmė

Pagrindinis sudaromo metodo tikslas – iš to paties piršto atspaudo kiekvieną kartą sugeneruoti tokį patį šifravimo raktą. Tai įmanoma tik tada, jeigu kiekvienu tokiu atveju generatorius bus inicijuojamas tais pačiais parametrais, kitaip tariant - jeigu metodui paduodami pradiniai duomenys kiekvienu atveju bus identiški. Tačiau kaip jau išsiaiškinta analizės dalyje, tai labai sudėtingas uždavinys, kadangi kiekvienas naujas piršto atspaudo pavyzdys nėra identiškas anksčiau buvusiems, vadinasi ir suformuota matrica kiekvienu atveju bus šiek tiek kitokia. Todėl pirmasis uždavinys – nustatyti, kaip apdoroti turimą piršto atspaudo matricą, kad generatorius gautų reikiamus parametrus.

Šiam uždaviniui išspręsti reikia:

- nustatyti stabiliausius piršto atspaudo *minutiae* taškus, kurie kiekvieną kartą bus (arba kurių buvimo tikimybė didžiausia) sudarytoje matricoje
- iš sudarytos matricos suformuoti parametrų sąrašą, kurie bus perduoti šifravimo rakto generatoriui
- įvesti klaidų kontrolę, kuri ištaisytų nors nedidelius gautų parametrų nukrypimus

Paskutinieji du žingsniai gali būti vykdomi bet kokia tvarka, t.y. klaidų kontrolė gali būti įvedama formuojant parametrus generatoriui, arba realizuota pačiame generatoriuje. Be abejo, galima ir abiejų variantų kombinacija, kurią ir naudosime.

Taigi išanalizavę piršto atspaudą ir apdoroję sudarytą matricą, turime gauti stabilius parametrus, kuriuos paduosime generatoriui ir gausime šifravimo raktą.

2.5. Metodo detalizacija

2.5.1. Stabilių minutiae taškų išskyrimas

Kadangi pirštų atspaudų analizei nusprendėme naudoti *minutiae* taškus, reikia išsiaiškinti, kurie jų stabiliausi ir patikimiausia, bei kuriais galima remtis generuojant šifravimo raktus. Literatūroje apie pirštų atspaudus ir jų analizės specifiką vienareikšmiškai teigiama, jog pačios stabiliausios *minutiae* charakteristikos yra piršto pagalvėlės centre. Kaip atskaitos taškas joms nustatyti naudojamas *minutiae branduolys* (*angl. core point*).

Šio branduolio radimui taip pat naudojami specialūs algoritmai, tik juose paprastai apdorojama ne *minutiae* matrica, o pradinis piršto atspaudo atvaizdas [22]. Sudaromame metode taip pat remsimės branduolio tašku, todėl vienas iš uždavinių – jį teisingai nustatyti. Šio taško nustatymo uždavinys - sudėtinga ir atskira sritis, kurios šiame darbe nenagrinėsime, o laikysime, kad jo vieta gautojoje matricoje yra žinoma. *Minutiae* branduolį naudosime kaip atskaitos tašką, nuo kurio pradėsime konstruoti parametrus šifravimo rakto generatoriui. Tiksli šio taško paskirtis bus aprašyta truputį vėliau.

Kaip jau minėta, eksperimentais nustatyta, kad stabiliausi *minutiae* taškai išsidėstę būtent aplink minėtąjį branduolį, t.y. šie taškai turėtų būti randami kiekvieną kartą nuskaitant piršto atspaudą. Žinant šį faktą, šifravimo rakto generavimui turi būti naudojami būtent arčiausiai branduolio esantys taškai. Be abejo metodui paduodamoje matricoje turi būti ir šie taškai, ir pats branduolio taškas, tačiau nėra garantijų, jog taip bus kiekvieną kartą. Lygiai taip pat negalima laikyti, jog visų šių taškų pozicijos kiekvieną kartą išliks tos pačios. Vadinasi šioje vietoje didelę reikšmę turės atspaudo skaitytuvas bei nuskaitytą atvaizdą apdorojantis ir reikiamą matricą formuojantis algoritmas.

Taigi kai išsiaiškinome dominančią piršto atspaudo sritį galima apibrėžti ir naudojamos matricos dydį. Standartinė populiariausių pirštų atspaudų skaitytuvų raiška siekia apie 500 dpi, todėl įvertinus dominančios srities matmenis bei šiek tiek supaprastinus situaciją, laikykime, jog metodas naudos 200 x 200 elementų dydžio matricas.

Apibendrinant situaciją: generatoriui paduodamus parametrus formuojame iš 200 x 200 taškų matricos, kurioje *minutiae* taškų vietos pažymėtos *„1“*. Kaip atskaitos taško naudojamo *minutiae* branduolio koordinatės matricoje taip pat yra žinomos iš anksto.

2.5.2. Parametrų formavimas

Literatūroje yra aprašyta metodų, kaip turint išskirtus *minutiae* taškus konstruojamos įvairios figūros. Paprastai jos naudojamos pirštų atspaudų palyginimo ir atitikmens paieškos uždaviniuose [23]. Viena populiariausių tokiu principu konstruojamų figūrų – trikampis, kurio kraštinių ilgis ir kampai naudojami pirštų atspaudų palyginimui [24]. Siūlomame metode taip pat bus panaudota keletas iš nagrinėtuose šaltiniuose pateiktų idėjų. Tačiau mes nekonstruosime jokių geometrinių figūrų, jų tarpusavy nelyginsime, o ir *minutiae* taško tipas (priešingai nei kai kuriuose iš naginėtų metodų) mums visiškai nesvarbus.

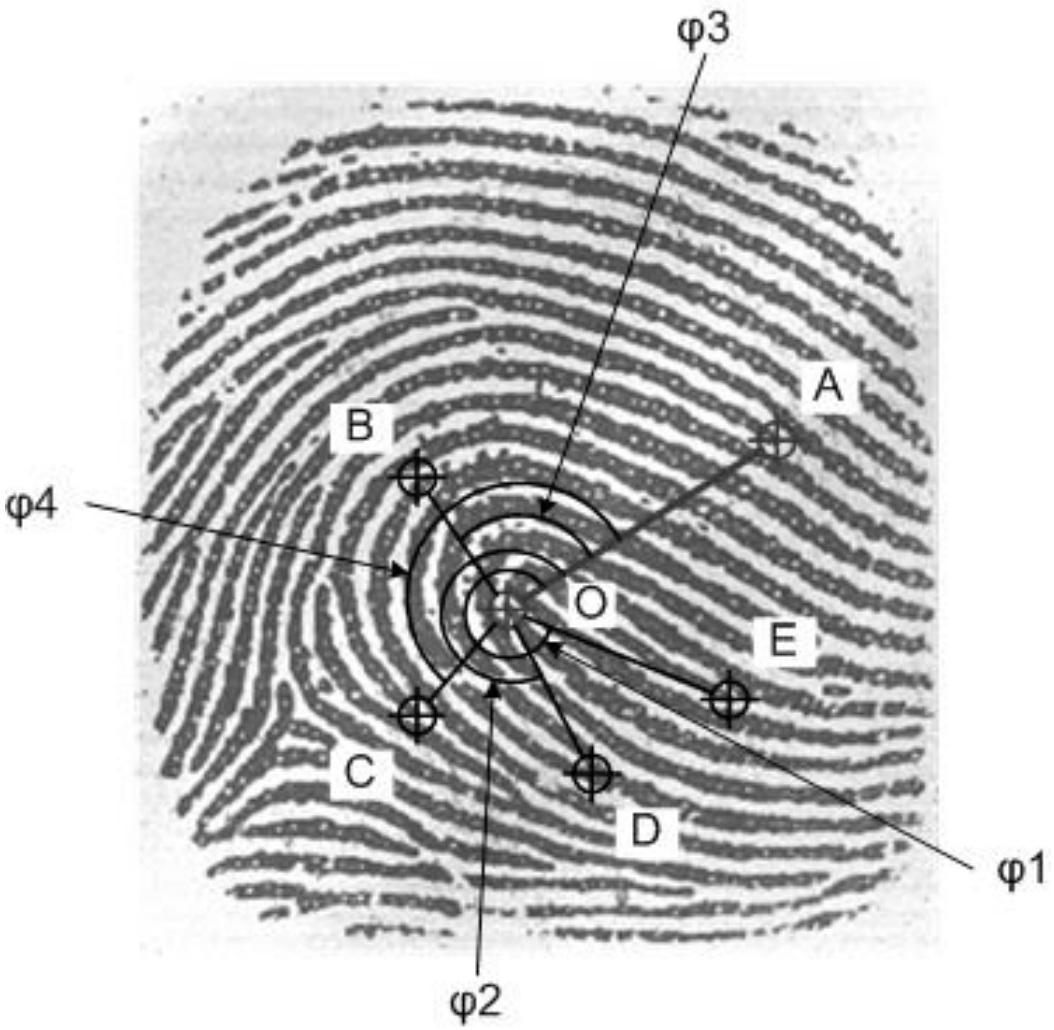
Kadangi paruošiamuosius darbus jau atlikome, t.y. apsibrėžėme situaciją, laukiamus rezultatus, išsiaiškinome pradinių duomenų formą ir jų reikšmę, galime išsiaiškinti ir patį šifravimo rakto generavimo principą. Šiame skyrelyje aptarsime generatoriui paduodamų parametrų formavimą, o aprašydami remsimės maksimaliu reikalingų parametrų skaičiumi (ne visais atvejais visi paduoti parametrai bus panaudoti). Kada, kaip ir kokie parametrai bus generatoriaus panaudoti, išsiaiškinsime kitame skyrelyje. Taigi, turėdami pradinę matricą ir branduolio taško [25] koordinates joje, toliau atliekame šiuos žingsnius:

- surandame 17 taškų, esančių arčiausiai branduolio
- atkarpa, jungianti branduolį su labiausiai nutolusiu iš 17 surastų taškų, tampa polinės koordinačių sistemos atskaita
- sudaroma pirmoji skaičių seka iš mažėjimo tvarka surikiuotų likusių 16 taškų atstumų nuo branduolio
- sudaroma antroji skaičių seka iš likusių 16 atkarpų (jungiančių branduolį ir vieną iš atrinktų taškų) teigiamų posūkio kampų, nusakytų susidarytos atskaitos atkarpos atžvilgiu (seka rikiuojama taip: kurioje pirmosios sekos pozicijoje yra konkrečios atkarpos ilgis, toje pat antrosios sekos pozicijoje yra tos atkarpos posūkio kampas, kitaip tariant. ilgiausios atkarpos posūkio kampas bus pirmas, trumpiausios - paskutinis)
- sudarytos sekos nariai suapvalinami iki sveikosios dalies

Atlikę šiuos žingsnius turime du 16 elementų masyvus, kuriuos sudaro iš piršto atspaudo *minutiae* charakteristikų gauti parametrai. Paskutinis paminėtas žingsnis įveda klaidų kontrolę, kas turėtų padėti sugeneruoti tokį patį šifravimo raktą esant nedideliems atspaudo neatitikimams. Žinoma, klaidų kontrolė taip pat yra labai daug dėmesio reikalaujanti sritis, kadangi įvedus per didelę klaidų toleranciją, padidėja tikimybė, jog dviem visiškai skirtin-

giems atspaudams bus sugeneruotas toks pat raktas. Beje, dar visiškai neaišku, ar taip nebus ir mūsų konstruojamu atveju. Tačiau bent minimalią kontrolę įvesti būtina, ką ir padarėme.

Taigi po visų šių veiksmų pirmajame masyve turime sveikųjų skaičių seką, kurios nariai yra iš intervalo [0; 359]. Kadangi naudojame 200 x 200 taškų matricą, tai didžiausias galimas skaičius antrojoje sekoje (atstumas nuo vieno matricos kampo iki kito) yra lygus $\sqrt{200^2 + 200^2} \approx 283$, tačiau turint omenyje, kad *minutiae* branduolys turėtų būti apytiksliai nagrinėjamos matricos centre, o patys tolimiausi taškai praktiškai nebus įtraukiami į parametrų sąrašus, galime laikyti, jog maksimalus galimas atkarpos ilgis yra 128. Taigi antrojoje sekoje yra sveikieji skaičiai iš intervalo [1; 128].



3 pav. Parametrai suformuoti iš piršto atspaudo.

3 paveiksle pateiktas realus parametrų formavimo pavyzdys, tačiau paprastumo sumetimais čia naudojami tik 5 taškai. Piršto atspaudo atvaizde branduolys pažymėtas tašku *O*, aplink jį surasti 5 arčiausiai esantys taškai (*A, B, C, D, E*). Taškas *A* yra toliausiai nutolęs nuo branduolio (iš visų 5 nagrinėjamų taškų), todėl atkarpa *OA* tampa polinės koordinačių sistemos atskaita. Turime dar 4 reikalingas atkarpas, t.y. *OE, OD, OB, OC*. Užrašome šių atkarpų ilgius mažėjimo tvarka: l_E, l_D, l_B, l_C. Analogiškai (pagal apsibrėžtą tvarką) užrašome ir šių atkarpų posūkių kampus atskaitos *OA* atžvilgiu: $\varphi 1$, $\varphi 2$, $\varphi 3$, $\varphi 4$. Gautus skaičius suapvaliname iki sveikosios dalies ir turime parametrus, kuriuos jau galime paduoti generatoriui.

Konkrečiu atveju išmatavę atkarpų ilgius ir jų posūkių kampus bei atlikę reikiamą apvalinimą turėsime tokius 2 parametrų masyvus: [*60, 46, 38, 34*] ir [*305, 263, 90, 196*]. Esant realioms sąlygoms viskas atliekama identiškai, išskyrus tai, kad masyvuose bus ne po 4, o po 16 elementų.

2.5.3. Rakto generavimas

Turėdami suformuotus parametrų masyvus toliau galime generuoti ir patį šifravimo raktą. Šiame darbe naudosime nesudėtingus raktų generatorius, kuriais generuosime 64 ir 128 bitų ilgio raktus. Šiuo metu 64 bitų šifravimo raktai niekur nebenaudojami, o 128 bitų ilgio raktus išstumia 256 bitų, tačiau kadangi mes šiuo darbu nepretenduojame į rimtą komercinį produktą, išvadoms paradyti užteks ir pasirinktų 64 ir 128 bitų ilgių raktų.

Šifravimo raktą generuosime keliais būdais t.y. naudosime skirtingus generatorius. Kaip jau minėjome, generatorius ne visais atvejais panaudos visus gaunamus parametrus – tokiu būdu galėsime patikrinti rakto stabilumą priklausomai nuo generavimui panaudotų piršto atspaudo taškų charakteristikų ir jų skaičiaus. Iš kokių ir kiek parametrų konkretus generatorius formuos šifravimo raktą, pateikta *4 lentelėje*.

Taigi iš viso turime 8 generatoriaus variantus, kiekvienas jų generuoja 64 arba 128 bitų raktą tam naudodamas 8 arba 16 taškų (neįskaitant branduolio taško ir papildomo atskaitos sistemos atkarpos taško, t.y. iš tikrųjų naudojami atitinkamai 10 ir 18 taškų). Raktas generuojamas arba naudojant tik atkarpų posūkių kampų parametrus, arba naudojant ir atkarpų ilgių, ir jų posūkių kampų parametrus. Antruoju atveju abiejų tipų parametrų svoris rakte yra vienodas (t.y. sudaro vienodą bitų skaičių). Vertinant tik vieno tipo parametrus pasirinkti posūkių kampai, kadangi didesnė jos reikšmių sritis.

4 lentelė. Šifravimo rakto generatorių variantai

Generatoriaus variantas	*Rakto ilgis, b*	*Naudojamų taškų kiekis*	*Taško atstumo nuo branduolio svoris rakte, b*	*Atkarpos posūkio kampo svoris rakte, b*
1	64	8	0	8
2			4	4
3		16	0	4
4			2	2
5	128	8	0	16
6			8	8
7		16	0	8
8			4	4

5 lentelėje pateikti atitinkamų parametrų koeficientai kiekvienam generatoriaus variantui. Konkretus parametras padauginamas iš jam priskirto koeficiento ir gautas rezultatas suapvalinamas iki sveikosios dalies. Pastarasis žingsnis įveda papildomą atsparumą klaidoms ir neatitikimams. Gautoji reikšmė užrašoma dvejetaine forma ir taip suformuojamas visas reikiamo ilgio raktas.

5 – ojo generatoriaus atveju iš taško atstumo gauta dvejetainė reikšmė užrašoma tiesiogine ir pakartojama atvirkštine tvarka – taip gaunamas reikiamas bitų skaičius iš 1 taško (konkrečiu atveju – 8 bitai).

5 lentelė. Generatoriaus parametrų koeficientai

Generatoriaus variantas	*Rakto ilgis, b*	*Naudojamų taškų kiekis*	*Taško atstumo nuo branduolio koeficientas*	*Atkarpos posūkio kampo koeficientas*
1	64	8	0	32 / 45
2			1 / 8	2 / 45
3		16	0	2 / 45
4			1 / 32	1 / 90
5	128	8	0	32 / 45
6			2	32 / 45
7		16	0	32 / 45
8			1 / 8	2 / 45

Žemiau pateiktos formulės, pagal kurias sugeneruojamos atitinkamo ilgio rakto dalys. *3* ir *4* generatorių atvejais pritaikius pateiktą formulę gaunami 4 bitai, *1, 2, 7* ir *8* generatorių atvejais – 8 bitai, o *5* ir *6* generatorių atvejais – 16 bitų. Gautieji bitai surašomi nuosekliai apdorojant parametrų masyvų elementus kol gaunamas reikiamo ilgio raktas. Formulėse taško atstumas nuo branduolio žymimas l, o atitinkamos atkarpos posūkio kampas – φ. Ženklas „+“ reiškia ne sudėtį, o gautų bitų sekų sujungimą.

1) $^{32}/_{45} * \varphi$
2) $^{1}/_{8} * l + {}^{2}/_{45} * \varphi$
3) $^{2}/_{45} * \varphi$
4) $^{1}/_{32} * l + {}^{1}/_{90} * \varphi$
5) $^{32}/_{45} * \varphi$ (seka parašoma tiesiogine tvarka ir pakartojama atvirkštine)
6) $2 * l + {}^{32}/_{45} * \varphi$
7) $^{32}/_{45} * \varphi$
8) $^{1}/_{8} * l + {}^{2}/_{45} * \varphi$

Kaip galima pastebėti *1, 5* ir *7* bei *4* ir *8* generatorių formulės sutampa. Tačiau jie generuoja skirtingo ilgio rakto dalis, rezultatai užrašomi skirtinga tvarka, arba generavime naudojamas skirtingas parametrų kiekis. Todėl negalima iš karto teigti, kad šių generatorių rezultatų stabilumo rodikliai būtiniai sutaps – tuo dar reikia įsitikinti.

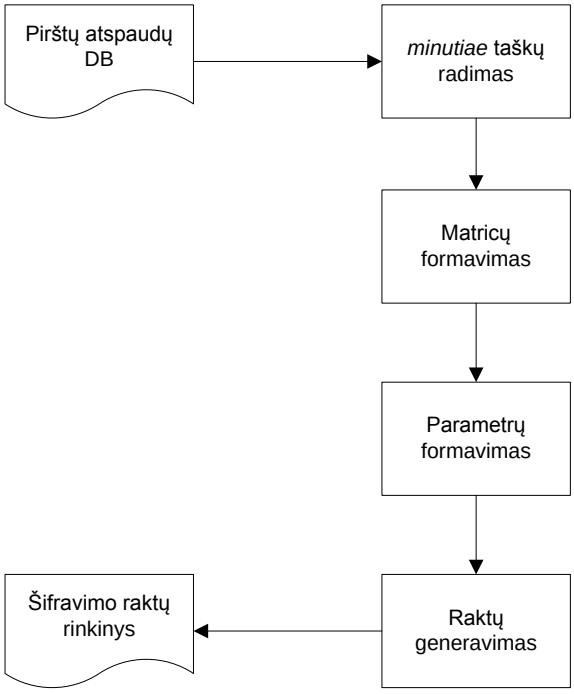
2.5.4. Pasiūlyto metodo apibendrinimas ir savybės

Sudarytas metodas iš suformuotos piršto atspaudo *minutiae* taškų matricos suformuoja parametrų seką ir iš jos sugeneruoja šifravimo raktus. Pasiūlyti viso 8 generatoriaus variantai, generuojantys 64 ir 128 bitų ilgio raktus, tam naudodami 8 ar 16 taškų bei įvertinantys posūkių kampus arba posūkių kampus ir atkarpų ilgius. Parametrams formuoti naudojami patys stabiliausi taškai – branduolys ir arčiausiai jo esantys kiti taškai.

Rakto generavimui naudojami parametrai sudaromi iš atkarpų, jungiančių branduolį su 17 arčiausiai jo esančiais taškais, ilgių ir jų posūkių kampų. Atkarpų ilgis ne absoliutinis, o santykinis (pikseliais), kampai nusakomi polinėje koordinačių sistemoje, kurios atskaita laikome atkarpą, jungiančią branduolį ir arčiausiai jo esantį atitinkamai 9-ąjį arba 17-ąjį tašką. Šifravimo raktas generuojamas apdorojant šiuos parametrus ir gautus rezultatus nuosekliai surašant į bitų seką.

Siekiant įvesti atsparumą klaidoms ir atspaudų neatitikimams naudojami tik sveikieji skaičiai, o rakto generavimui parenkami tik arčiausiai branduolio esantys taškai ir atitinkamų atkarpų posūkio kampai. Vadinais galutinis rezultatas priklauso nuo konkretaus atspaudo taškų išsidėstymo, o ne nuo ženklų po kablelio skaičiaus. Kita vertus, nors ir kiekvienas atspaudas yra unikalus, dar nereškia, jog pasirinkta klaidų kontrolė pasiteisins ir nepablogins rezultatų. Taip pat verta dar kartą paminėti, jog formuojant parametrus atskaitos pradžia yra dinaminė, t.y. kiekvieną kartą nustatomi iš naujo pagal gautus parametrus.

Bendroji pasiūlyto metodo schema pateikta *4 pav.* Joje parodyti pagrindiniai šifravimo raktų generavimo iš pirštų atspaudų etapai. Turint atspaudų duomenų bazę, juose surandami *minutiae* taškai, pagal kuriuos suformuojamos matricos. Iš gautųjų matricų suformuojami parametrai, pagal kuriuos sugeneruojami šifravimo raktai.



4 pav. Pasiūlyto metodo principinė schema.

Kiek sprendimo metodas yra efektyvus, kokius rezultatus pateikia bei kokie jo privalumai bei trūkumai išsiaiškinsime eksperimentinėje dalyje.

2.6. Išvados

Šiame skyriuje apsibrėžtas darbo tikslas, iškelti uždaviniai, nustatyta sudaromo metodo taikymo sritis ir jam keliami reikalavimai.

Pirštų atspaudų požymius galima nesudėtingai aprašyti skaitmenine forma. Vienas populiariausių variantų – *minutiae* taškų matrica. Tokią matricą pasirinkome šiame skyriuje, ja pateiksime pradinius duomenis sudaromam metodui.

Svarbiausias reikalavimas tiesioginiam rakto generavimui iš piršto atspaudo – stabilių parametrų parinkimas. Stabiliausi *minutiae* taškai yra piršto pagalvėlės branduolys ir aplink jį išsidėstę kiti taškai. Būtent šie taškai ir turėtų būti naudojami šifravimo rakto generavimui.

Rakto generatoriui parametrai formuojami iš atkarpų, jungiančių *minutiae* taškus su branduoliu, santykinių ilgių ir šių atkarpų posūkio kampų.

Siūlomi 8 generatoriaus variantai iš gautų parametrų suformuojantys 64 arba 128 bitų ilgio raktus naudojant 10 ar 18 matricos taškų (įskaitant branduolį ir papildomą tašką atskaitos sistemai sudaryti). Pateiktos formulės rakto dalims (bitų sekoms) generuoti. Nuosekliai sujungus šias sekas gaunamas reikiamo ilgio raktas.

Kadangi parametrai formuojami atskirai kiekvienam piršto atspaudui, o atskaitos sistema taip pat parenkama individualiai, galima teigti, jog galutinis rezultatas priklauso tik nuo piršto atspaudo (iš jo suformuotos matricos).

3. ŠIFRAVIMO RAKTŲ GENERAVIMO IŠ PIRŠTO ATSPAUDO METODO REALIZACIJA

3.1. Reikalingi įrankiai

Šio darbo eksperimentui atlikti reikalingas programinis įrankis, galintis iš nuskenuoto piršto atspaudo sugeneruoti reikiamo ilgio šifravimo raktus pagal ankstesniame skyriuje aprašytus žingsnius. Siekiant paprastumo realizacijos dalis suskaidyta į 2 dedamąsias. Taigi reikalingi 2 dalykai:

- programa, apdorojanti piršto atspaudo atvaizdą (paveikslėlį) ir iš jo sugeneruojanti *minutiae* taškų matricą
- programa apdorojanti sudarytą matricą ir iš susiformuotų parametrų sugeneruojanti šifravimo raktus

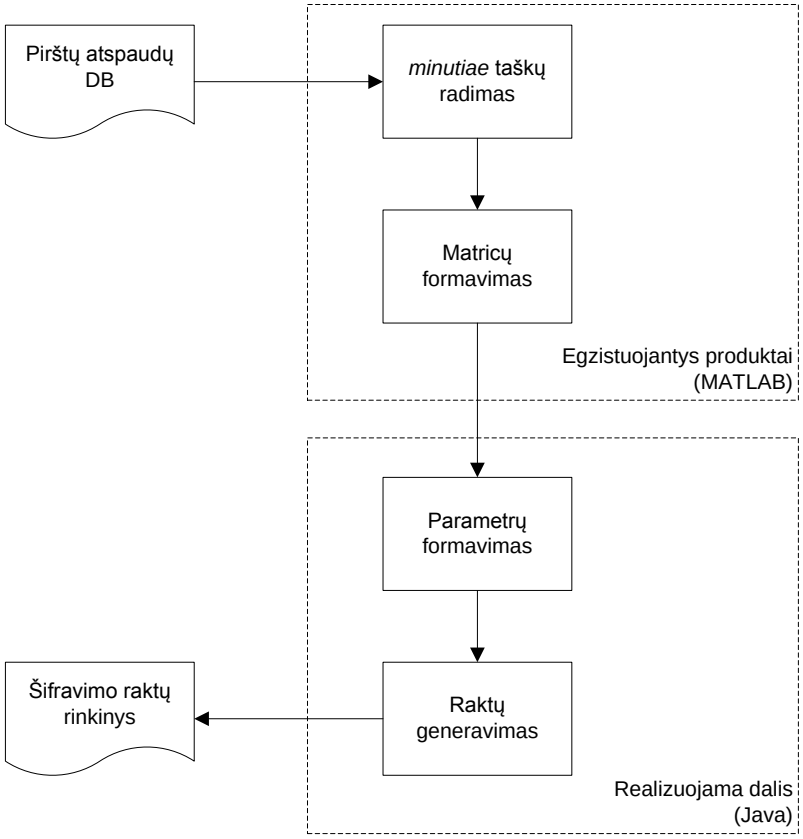
Kaip jau minėta anksčiau, piršto atspaudo apdorojimas ir *minutiae* taškų nustatymas jau savaime yra pakankamai sudėtingas uždavinys. Kadangi darbo tikslas nėra tiesiogiai susijęs su pirštų atspaudų apdorojimu ir reikiamų charakteristikų išskyrimu, šiam darbui atlikti nutarta pasinaudoti jau egzistuojančiomis priemonėmis.

Siekiant bent minimaliai įvertinti naudojamo algoritmo įtaką matricos formavimui, nuspręsta naudoti du produktus, abiems parenkant kiek galima vienodesnius vykdymo parametrus. Šiam tikslui pasirinktos dvi atviro kodo priemonės: *MATLAB* aplinkoje realizuota programa *FingerPrint Application* [27] bei pirštų atspaudų apdorojimo biblioteka *Fingerprint Verification System* [28]. Pirmojoje realizuotas tiek reikalingas *minutiae* charakteristikų išskyrimas iš piršto atspaudo, tiek keletas papildomų galimybių, kurių šiame darbe nenagrinėsime. Antroji (pasirinkome taip pat *MATLAB* aplinkai parašytą kodą) atlieka tik mums reikalingą funkciją – išskiria *minutiae* taškus.

Taigi išsiaiškinus *FingerPrint Application* ir *Fingerprint Verification System* veikimą ir atitinkamai modifikavus jų išeities kodus, gauti du algoritmai, apdorojantys piršto atspaudo atvaizdą ir rastus *minutiae* taškus išsaugantys 200 x 200 matricoje tekstiniame faile. Deja nė viena iš pasirinktų programų negali atlikti dar vieno mums reikalingo veiksmo - nustatyti branduolio vietos matricoje. Šis uždavinys, kaip jau minėjome praeitame skyriuje, yra pakankamai sudėtingas, jam spręsti naudojami specialūs algoritmai, kurie realizuoti komerciniuose produktuose. Deja šių produktų įsigyti nebuvo galimybės, todėl branduolio pozicijai matricoje

nustatyti panaudosime euristinį sprendimo būdą, t.y lyginsime gautą matricą su pradiniu piršto atspaudo atvaizdu.

Antroji realizacijai reikalinga dalis ruošiama savarankiškai. Ji apdoroja pirmąja programa suformuotą matricą ir sugeneruoja galutinus rezultatus Duomenimis programos apsikeičia tekstinių failų pagalba, o iš vieno duomenų failo sugeneruojami 8 šifravimo raktai.



5 pav. Šifravimo raktų generavimo iš piršto atspaudo realizacijos schema.

5 pav. pateikta realizacijos schema (papildyta principinė metodo schema). Kaip ir buvo minėta, pirštų atspaudų apdorojimas, *minutiae* taškų radimas ir matricų formavimas atliekamas pasinaudojant jau esančiais *MATLAB* produktais. Savo jėgomis realizuoti parametrų formavimo (iš turimų matricų) ir raktų generavimo algoritmai.

3.2. Programos veikimo aprašymas

Šį skyrių toliau nagrinėsime laikydami, jog tekstiniame faile turime 200 x 200 elementų matricą, sudarytą iš *0* ir *1,* kur *1* reiškia, jog šioje piršto atspaudo vietoje yra vienas iš *minutiae* taškų, *0* – kad tokio taško toje vietoje nėra. Failo pabaigoje taip pat įrašytos branduolio (atskaitos) taško koordinatės. Pastarasis taškas matricoje taip pat pažymėtas *„1“*.

Realizuota programa valdoma komandinės eilutės pagalba. Kadangi tai tik pagalbinė priemonė, nebuvo tikslo kurti grafinės sąsajos. Komandinėje eilutėje iškviečiamas reikiamas vykdomasis failas, nurodomi reikiami parametrai ir programa pateikia galutinį rezultatą jau be vartotojo įsikišimo. Programos veikimo algoritmo labai nedetalizuosime, o jo vykdymo metu atliekami šie žingsniai:

- iš duomenų failo nuskaitoma matrica ir patikrinama, ar gautas reikiamas taškų skaičius (18)
- apskaičiuojami taškų santykiniai atstumai (atkarpų ilgiai) nuo branduolio taško (*A[]*)
- masyvas *A* surikiuojamas didėjimo tvarka ir iš jo paimami pirmieji 17 elementų (*B[]*)
- suformuojamos 2 polinių koordinačių sistemų atskaitos: pagal 9-ąją atkarpą (*B[8]*) – s_1 ir pagal 17-ąją atkarpą (*B[16]*) – s_2
- randami atkarpų *B[0..15]* teigiami posūkių kampai atkarpos *B[16]* atžvilgiu (*K16[]*) bei atkarpų *B[0..7]* teigiami posūkių kampai atkarpos *B[8]* atžvilgiu (*K8[]*)
- kviečiami generatorių metodai, kurie naudodamiesi *B[], K8[]* ir *K16[]* duomenimis bei pritaikę *5 lentelėje* pateiktus koeficientus sugeneruoja šifravimo raktus
- gautieji raktai šešioliktaine forma pateikiami ekrane ir atspausdinami rezultatų faile

Masyvas *B[]* realizacijoje yra duomenų struktūra, kurioje saugomos taško koordinatės branduolio taško atžvilgiu bei abu šiuos taškus jungiančios atkarpos ilgis, todėl aprašyme jis iš pradžių paminėtas kaip taškų atstumų, o vėliau ir kaip atkarpų masyvas. Struktūrose *B[], K8[]* ir *K16[] i*-ojo elemento reikšmė apibūdina to paties taško (atkarpos) parametrus.

3.3. Testavimo modelis

Pasiūlytos metodo realizacija turi padėti atsakyti į 2 pagrindinius klausimus:

- ar pagal šią metodiką generuojant šifravimo raktą iš to paties piršto atspaudo visuomet bus gautas tas pats rezultatas
- kokia tikimybė, jog iš dviejų skirtingų atspaudų bus sugeneruoti vienodi šifravimo raktai

Atsakant į šiuos klausimus reikia atsižvelgti į skirtingų naudotų generatorių įtaką rezultatams. Vadinasi tuo pačiu bus įvertintas ir naudotų charakteristikų stabilumas bei jų patikimumas priklausomai nuo generuojamo rakto ilgio.

Kad ir kokie rezultatai paaiškėtų atlikus eksperimentą, vienareikšmiškai teigti, jog jie priklauso tik nuo pasiūlytos metodo, negalima. Didelį vaidmenį vaidina pats atspaudo nuskaitymas, apdorojimas ir matricos suformavimas. Jeigu šie elementai dirbtų idealiai, o ir pats piršto atspaudas kiekvieną kartą būtų identiškas, tuomet galutinis rezultatas tikrai priklausytų tik nuo metodo. Tačiau kadangi taip nėra, tikėkimės, kad įvesta klaidos tolerancija padės pasiekti gerų rezultatų.

3.3.1. Duomenys, naudojami matricų formavimui

Kadangi matricos, kurias naudosime raktų generavimui, *FingerPrint Application* ir *Fingerprint Verification System* pagalba bus formuojamos iš realių pirštų atspaudų atvaizdų, reikia susidaryti eksperimentui naudojamų atspaudų duomenų bazę. Kad rezultatai būtų kuo artimesni realybei, nutarta pasirinkti *FVC* (*angl. Fingerprint Verification Competition*) 2000, 2002 ir 2004 metų konkursuose naudotus pirštų atspaudus (2006 metų konkurso duomenų bazių rasti nepavyko). Iš šių bei Vilniuje įsikūrusios kompanijos *„Neurotechnology“* siūlomų pirštų atspaudų duomenų bazių [29] atsirinkti 8 pirštų atspaudai. Kiekvienas pirštas nuskaitytas 8 kartus (pateikiama po 8 vieno piršto atspaudus), vadinasi eksperimentui naudosime duomenų bazę, sudarytą iš 64 atspaudų.

Verta pažymėti, jog atrinkti atspaudai iš karto negalėjo būti panaudoti. Pirmiausia metodas pritaikytas dirbti su 200x200 dydžio matrica, todėl situacija būtų paprastesnė, jeigu ir atspaudo dydis būtų toks pat – tai palengvintų matricos formavimą. Taigi iš atrinktų atspaudų atvaizdų buvo iškirptos 200x200 vaizdo taškų sritys, kiekviename rinkinyje vaizduojant tą pačią (kiek tai įmanoma) sritį. Taip suformuotą duomenų bazę iš 64 pirštų atspaudų atvaizdų ir naudosime eksperimentui atlikti.

3.4. Pasiūlyto metodo ir jo realizacijos apibendrinimas

Pasiūlytas metodas iš tekstiniame faile pateiktos dvejetainės matricos sugeneruoja 8 šifravimo raktus: po keturis 64 ir 128 bitų ilgio. Faile taip pat pateikiamas matricoje esančio pagrindinio taško (branduolio) koordinatės.

Realizuota *Java* programa šifravimo raktus generuoja be vartotojo įsikišimo: pastarajam reikia tik nurodyti tekstinį failą su pradiniais duomenimis (suformuota matrica) ir sulaukti rezultato.

Realizacija nesiriša konkrečiai prie piršto atspaudo: ji gali sugeneruoti rezultatą iš bet kokiu būdu (iš bet kokių pradinių duomenų) suformuotos matricos. Tačiau kadangi darbe nagrinėjame pirštų atspaudus, jais remiantis pateikiami visi pavyzdžiai.

Piršto atspaudo nuskaitymas, apdorojimas, matricos suformavimas – atskiro darbo tema, čia šių uždavinių nenagrinėjame.

Eksperimentui atlikti pasiruošti 8 pirštų atspaudai (po 8 pavyzdžius kiekvienam). Kiekviename rinkinyje vaizduojama ta pati sritis, kurios dydis 200x200 vaizdo taškų.

4. ŠIFRAVIMO RAKTŲ GENERAVIMO IŠ PIRŠTO ATSPAUDO METODO EKSPERIMENTINIS TYRIMAS

Šiame skyriuje atliksime eksperimentą, naudodami pasiruoštus pirštų atspaudų rinkinius ir realizacijos dalyje aprašytas programas (tarp jų - pasiūlyto metodo realizaciją, t.y. 8 šifravimo rakto generatorius).

4.1. Eksperimentinės dalies tikslas

Eksperimento metu palyginsime realizuotus raktų generatorius, įvertinsime pasirinktų parametrų stabilumą bei, kaip minėta anksčiau, pabandysime atsakyti į du pagrindinius klausimus:

- ar iš to paties piršto atspaudų visuomet bus gaunami tokie patys šifravimo raktai
- ar iš skirtingų pirštų atspaudų visuomet bus sugeneruoti skirtingi šifravimo raktai

Atsakius į šiuos klausimus galėsime daryti išvadas apie galimybę pirštų atspaudus naudoti šifravimo ir kitiems slaptiems raktams generuoti.

4.2. Eksperimento eiga

Eksperimentas atliktas tokia eilės tvarka:

- iš pasiruoštos pirštų atspaudų duomenų bazės suformuotos pradinės matricos
- matricos apdorotos *Java* aplinkoje realizuotais šifravimo raktų generatoriais
- generatorių rezultatai atspausdinti tekstiniuose failuose

Viso suformuoti 2 pradinių matricų rinkiniai: vienas su *FingerPrint Application,* kitas su *Fingerprint Verification System.* Abiem atvejais gauti 64 matricų rinkiniai, kurie apdoroti realizuotais generatoriais, rezultatus išspausdinant tekstiniuose failuose, kad būtų paprasčiau analizuoti. Viename faile pateikiami iš vieno piršto atspaudų sugeneruoti šifravimo raktai. Kadangi kiekvienam pirštui paruošta po 8 atspaudus ir naudojami 8 generatoriai, vienam pirštui sugeneruoti 64 raktai. Įvertinant tai, kad eksperimente naudoti 8-ių pirštų atspaudai ir pats eksperimentas pakartotas 2 kartus, iš viso sugeneruoti 1024 šifravimo raktai.

4.3. Eksperimento rezultatai

Eksperimento daliniai rezultatai pateikiami *6 lentelėje*. Joje surašyti šifravimo raktai, gauti iš vieno piršto matricų rinkinio (8 matricų). Bendrą vaizdą galima susidaryti ir iš pateiktųjų rezultatų, o likusieji pateikti *1 priede*.

Taigi pirmiausia verta dar kartą pabrėžti, jog visi pateikti šifravimo raktai (jų *6 lentelėje* yra 128) sugeneruoti iš vieno piršto atspaudų. Ir galima aiškiai matyti, jog sutampančių raktų nėra. Jeigu skirtingi raktai būtų gauti tik iš matricų, suformuotų naudojant skirtingus algoritmus, tokią situacija būtų galima suprasti. Tačiau konkrečiu atveju skirtingi raktai gauti ir iš to paties piršto skirtingų atspaudų kai matricos formuojamos naudojant vieną ir tą patį metodą. Pabandykime pasiaiškinti, kodėl taip atsitiko.

Pirma kilusi mintis – galbūt projektinėje dalyje palikta spraga, kuri leidžia iš tų pačių duomenų (matricos) gauti skirtingus rezultatus. Tačiau atsirinktos 5 skirtingos matricos ir su kiekviena jų, naudojant realizuotą raktų generavimo įrankį, generavimas pakartotas po 100 kartų. Hipotezė nepasitvirtino – rezultatai gauti iš vienos matricos visais atvejais buvo identiški, vadinasi problemos reikia ieškoti kitur.

Kitas variantas – galbūt matricas formuojančios programos iš vieno piršto atspaudo gali sugeneruoti skirtingas matricas, kas ir lemtų tokius eksperimento rezultatus. Padarytas analogiškas bandymas kaip ir anksčiau aprašytu atveju: atrinkti 5 pirštų atspaudai ir kiekvienas jų po 100 kartų apdorotas viena programa (*FingerPrint Application*) ir po 100 kartų kita programa (*Fingerprint Verification System*). Prielaida nepasitvirtino ir šįkart – iš to paties piršto atspaudo visuomet buvo suformuojamos vienodos matricos.

Reikia paminėti, jog ir šio bandymo, ir viso eksperimento metu abi matricų formavimui naudotos programos tam pačiam piršto atspaudui sugeneruodavo skirtingas matricas. Deja nebuvo galimybės pasinaudoti nė vienu komerciniu praktikoje naudojamu produktu šiai užduočiai (*minutiae* matricos formavimui) atlikti, todėl negalima daryti išvadų, ar naudotos programos teisingai atlieka savo funkciją.

Įsitikinus, jog ir raktų generatoriai, ir matricų formavimo algoritmai iš gautų duomenų kiekvieną kartą suformuoja identiškus rezultatus, beliko viena vieta, dėl kurios gali būti toks eksperimento rezultatų variantiškumas – naudoti pirštų atspaudai.

6 lentelė. Eksperimento rezultatų (sugeneruotų šifravimo raktų) ištrauka

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	1ce971bb036ae03c 248e3dd600871772 1b0dfe17e3242cc1 3840c51efdf48504 ddb8faae79d104c1 1de437a8198819a7 0cecc6e23c204404 1e3316c9b2de2d75	7b3ea2cc3bae2389 240d0254236c9171 56d9bc5533d2a460 b6b9ec6d87f7551b f8414767604042c6 660ca80629a2a571 69c5317b667eecbe fbcec9d9d227e722
2	514e473b20161e13 5248333d30281117 61505f514e42322c 63646c515f4f3820 8d8b7f7a675d403c 414e434a3128111a 403e3c3e33322410 4143413c3b2d2227	47434a3c333a1208 5250404532362917 453d3b35231d1a06 4b4b3e36281f1501 5f5444464634240c 46303a20222a1a07 363c333736272e0b 5f5c5c5d4d423e12
3	4656732ceb38c3a0 8ca83c535c703b4a 3a93103f10f1e22c 78a6d5f5891644d5 55eb7370ebfb7d0c b4f93d3574907e70 2c38f3689746cac8 da590a20232dbe38	ecfa83bc406907e4 565422b465496bdb 0554688b1970e851 661fdadd88b35c2e b677d6dbaff11ff7 354687b6d717911e feda6dd07c3868fc b31887b11eeff404
4	9955544776423020 abaa875457100212 8aa8444744747403 9aa9b57566455531 99ba9898ba765743 a576474555641310 8b46745661113232 7656464444432302	7776646754520131 5555446555561232 4555566642103210 5547767766201303 a999757667744331 4555656571012003 7776577413021233 a886656447777501
5	1c38e997718ebbdd03c06a56e0073c3c 24248e713dbcd66b000087e117e8724e 1bd80db0fe7f17e8e3c724242c34c183 381c4002c5a31e78fdbff42f85a10420 ddbbb81dfa5fae75799ed18b0420c183 1db8e42737eca815199888111998a7e5 0c30ec37c663e2473c3c200444220420 1e7833cc1668c993b24dde7b2db475ae	7bde3e7ca245cc333bdcae7523c48991 24240db00240542a23c46c369189718e 566ad99bbc3d55aa33ccd24ba4256006 b66db99dec376db687e1f7ef55aa1bd8 f81f418247e267e6600640024242c663 66660c30a81506602994a245a5a5718e 6996c5a3318c7bde66667e7eec37be7d fbdfce73c993d99bd24b27e4e7e72244
6	5e1c4ce9407132bb2e031e6a1ae0103c 5c24488e3e3d3ad630002a871c171472 661b5e0d5afe56174ee34624302c24c1 6c38684064c5561e52fd4ef432852e04 8edd8eb876fa74ae6a7954d148043ac1 4c1d48e4483746a836192c88181914a7 400c3eec3cc63ae2363c302028441204 4c1e4833401630c930b228de262d2675	4c7b423e42a238cc343b30ae1a230e89 5624520d4a0240543823306c2a911671 405636d932bc325522331ad210a40c60 4ab640b93eec346d24871ef718550c1b 56f852414e474a674260324026420ec6 46663c0c38a82406222920a212a50c71 3e693ec53a31387b30662c7e24ec0cbe 5efb5cce58c950d94ad248273ee71a22
7	42655b607f3f25c7e4b13882cb32a804 8ec4ac8d3ac0533559c3720b35bd4ca7 30a7993a1c0e31ff1a0cfd16e2242bc0 7882a463dc59f7518991166f4e45d656 5753e5b8783d7c02e0bbfdb17cd307c4 b144fe9e3cd03b5d7b42950677e67704 23c93687fd376e8591724c67c2a5c98a dea9529d00a3220b293e21d5bde93880	e5c5f5a78c3db8c5400467910174e94f 5d6658492d28b4446851469867b1d5b5 035159446a8787ba1193760fed8c5e1b 616919f3dda2d7d08689bd3d57c725eb b564767ad669d1b1aaf3f81912f2f378 3b5d4b6c8776bb6fd67b1876981215e0 f8e2dca661d6d80872ce39836e87f5c7 b83710898878b11f1aeee9f8f1470642
8	848675767773626c5e4b43382c131a10 989c8a88837c6563554c3730332b141a 938a89837170636f61505f514e42322c 97989a868d757f756869615654443d25 a5a5ae9b979397908e8b7f7b675d403c 8b746f69635d534547444940372e1710 828c73686f636658493734363c3a2c18 7d7a7579605a52504243423d3b2e2328	6e6c6f6a58535b4c4440463930371e04 6566656462525b5456554449363b2d1b 706565645648484b413937302e181501 7676616f6d6a5d5d48483b33251c120e 8b8687877d665d5b5a5f4f41413f2f07 7365645658574b464d3731272921110e 6f6e5d5a565d4d40373c333836282f0c 8b83817878777b61515e5e5f4f443014

Iš pradžių reikia priminti, jog eksperimentų naudojamų pirštų atspaudų duomenų bazė buvo kruopščiai paruošta: atrinkti atspaudų atvaizdai ir iš jų iškirptos 200x200 vaizdo taškų sritys, kurios ir buvo naudojamos *minutiae* taškų radimui ir matricų formavimui. Vieno rinkinio (vieno piršto atspaudų) ribose visuomet buvo naudojama viena ir ta pati piršto atspaudo sritis, taigi programoms buvo paduodamas visuomet tos pačios vietos vaizdas.

Nors pirštų atspaudų rinkiniai vizualiai atrodė vienodai (nebent nežymiai skyrėsi atvaizdų kokybė), tačiau panagrinėjus kaip programos juos apdoroja, situacija tapo aiškesnė. Pabandžius palyginti kaip *minutiae* taškai išsidėstę skirtinguose vieno piršto atspauduose, pastebėta, jog randamas skirtingas taškų skaičius, o ir atitinkantys taškai ne visuomet aptinkami tose pačiose vietose. Panagrinėjus atidžiau, pastebėti esminiai kriterijai, dėl kurių skyrėsi piršto atspaudų atvaizduose surastų *minutiae* taškų išsidėstymas:

- ne visuose rinkinio atvaizduose sutampa linijų storis: kadangi algoritmas jas suplonina iki 1 vaizdo taško, skirtingas tų pačių linijų storis iškraipo galutinį suplonintą (*angl. thinned*) atspaudo atvaizdą
- ne visuose rinkinio atvaizduose sutampa linijų (ypač jų pabaigų) ryškumas; algoritmai linijos pabaigą „mato“ anksčiau negu ji iš tikrųjų pasibaigia
- pačiose linijose pasitaiko mažesnio ryškumo (kartais netgi visiškai išblukusių) atkarpų, kurios interpretuojamos kaip linijos trūkiai, dėl ko atsiranda papildomų *minutiae* taškų

Abejose matricų formavimui naudotose programose buvo įjungta „netikrų“ taškų šalinimo funkcija, t.y. jeigu atstumas tarp dviejų aptiktų taškų yra mažesnis už tam tikrą dydį, abu taškai atmetami. Ši funkcija turėjo padėti išvengti taškų, kurie atsirado dėl atspaudo nekokybiškumo, aptikimo ir padidinti atsparumą klaidoms, tačiau realiai ji nesuveikė taip, kaip tikėtasi.

Taigi būtent tokių eksperimento rezultatų priežastis – pirštų atspaudai, tiksliau jų atvaizdų neatitikimas, kuris matricas formuojančioms programoms buvo „neįkandamas“. Norint eksperimentą pakartoti ir tikintis geresnio rezultato, pirmiausia reikia sugalvoti (ar rasti jau sugalvotą) algoritmą, kuris būtų tolerantiškesnis pirštų atspaudų atvaizdų neatitikimams. Vadinasi, pirmiausia reikalingas patikimas *minutiae* taškų suradimo algoritmas, kuris to paties piršto atspaudams formuotų jeigu ne identiškas (kas praktiškai neįmanoma) tai bent kiek įmanoma mažiau besiskiriančias matricas. Tačiau ši užduotis – atskira magistro baigiamojo darbo tema.

4.4. Išvados

Eksperimentas turėjo padėti atsakyti į klausimą, ar pirštų atspaudus galima naudoti tiesioginiam šifravimo (ir kitokių) raktų generavimui, t.y. ar vieno piršto atspaudams visuomet būtų sugeneruojamas tas pats raktas ir ar skirtingų pirštų atspaudams visuomet būtų sugeneruojami skirtingi raktai.

Eksperimento metu iš pasiruoštų pirštų atspaudų atvaizdų pirmiausia buvo sugeneruojamos matricos (naudotos dvi programos), kurios vėliau buvo paduodamos raktų generatoriams. Naudota po 8 kiekvieno piršto atspaudų atvaizdus, kiekvienam jų suformuota po 2 matricas (atskiromis programomis). Iš vienos matricos sugeneruoti 8 šifravimo raktai, iš vieno piršto atspaudų – viso 128 raktai.

Gauti rezultatai nebuvo tokie, kokių tikėtasi – vieno piršto atspaudams nepavyko sugeneruoti vienodų šifravimo raktų. Pagrindinė to priežastis – naudotų pirštų atspaudų atvaizdų neatitikimai vienas kitam, dėl ko iš jų buvo sugeneruojamos visiškai nepanašios *minutiae* taškų matricos. Branduolio taškui rasti taip pat buvo naudojamas ne patikimas algoritmas, o euristinė analizė.

Tikintis sėkmingai pakartoti eksperimentą pirmiausia reikalingas pirštų atspaudų atvaizdų apdorojimo algoritmas, tolerantiškas jų neatitikimams, kas leistų suformuoti minimaliai besiskiriančias *minutiae* taškų matricas.

5. IŠVADOS

Duomenų saugumo problema darosi vis aktualesnė ne tik įstaigoms bei organizacijoms, bet ir privatiems asmenims. Ir šiandieną jau kiekvienas supranta, jog saugūs duomenys – tai užšifruoti duomenys.

Literatūros šaltinių ir egzistuojančių sistemų analizė parodė, jog duomenų šifravimo priemonių yra pakankamai: vienos integruotos naudojamose operacinėse sistemose, kitas, aparatines ar programines, galima įsigyti atskirai.

Bet kokia šifravimo sistema susideda iš dviejų dalių: šifravimo algoritmo ir šifravimo rakto, o esamų priemonių analizė tik patvirtino faktą, kad daugiausia problemų kelianti sritis – šifravimo raktų valdymas. Taip pat išsiaiškinta, jog raktų valdyme egzistuoja dvi pagrindinės iki šiol neišspręstos problemos: šifravimo raktą reikia atsiminti arba kažkur saugiai laikyti.

Šių problemų galimų sprendimo būdų analizė parodė, jog biometrija gali padėti pakeisti situaciją ir šiokių tokių užuomazgų ta linkme jau yra. O atlikus biometrinių charakteristikų analizę įsitikinta, jog bene paprasčiausias ir labiausiai išnagrinėtas būdas tai padaryti – naudoti pirštų atspaudus.

Projektinėje darbo dalyje pasiūlytas metodas iš piršto atspaudo leidžiantis tiesiogiai generuoti 64 ir 128 bitų ilgio šifravimo raktus. Metodo esmė: iš pirštų atspaudų atvaizdų suformuojamos *minutiae* taškų matricos, pagal kurias generuojami šifravimo raktai. Viso pasiūlyti 8 raktų generatoriaus variantai, generavimo procese naudojantys taškų atstumus nuo pagrindinio (branduolio) taško, bei suformuotų atkarpų posūkių kampus.

Realizuota programinė priemonė, apdorojanti pateiktas matricas ir sugeneruojanti šifravimo raktus. *Minutiae* taškų radimui ir matricų formavimui pasitelktos trečiųjų šalių programos.

Remiantis eksperimento rezultatais galima teigti, jog naudotų programinių priemonių rinkinio šiandien realiam raktų generavimui naudoti negalima – generuojant raktus iš to paties piršto atspaudų, visi gautieji raktai buvo skirtingi. Detalesnė rezultatų analizė parodė, jog didžiausią įtaką tokiems rezultatams turėjo pirštų atspaudų atvaizdus apdorojančios ir matricas formuojančios programos, nesusitvarkančios su atvaizdų nesutapimais. Įtakos neabejotinai turėjo ir branduolio taško radimui naudotas euristinis metodas.

Taigi norint pakartoti eksperimentą ir įsitikinti, ar tikrai pirštų atspaudai gali būti naudojami tiesioginiam šifravimo raktų generavimui, pirmiausia reikia turėti patikimą algoritmą, apdorojantį pirštų atspaudų atvaizdus ir formuojantį *minutiae* taškų matricas. Kaip įsitikinta darbo metu, šis algoritmas – pagrindinė ašis, nuo kurios ir reikia pradėti sprendžiant panašias problemas.

Apjungus šį ir būsimą kieno nors darbą apie reikiamo algoritmo paieškas, galima tikėtis sėkmingai pakartoti atliktą eksperimentą, o gauti rezultatai gali tapti pagrindu tiesioginiam šifravimo raktų generavimui iš pirštų atspaudų ir jų praktiniam pritaikymui šifravimo sistemose.

NAUDOTA LITERATŪRA

[1] The Ecrypting File System [interaktyvus], [žiūrėta 2008-12-06]. Prieiga per internetą: <http://technet.microsoft.com/en-us/library/cc700811.aspx>

[2] BitLocker Drive Encryption Technical Overview [interaktyvus], [žiūrėta 2008-12-06]. Prieiga per internetą: <http://technet.microsoft.com/en-us/library/cc732774.aspx>

[3] Mac OS X 10.4 Help: About FileVault [interaktyvus], [žiūrėta 2008-12-09]. Prieiga per internetą: <http://docs.info.apple.com/article.html?path=Mac/10.4/en/mh1877.html>

[4] EncFS Encrypted Filesystem [interaktyvus], [žiūrėta 2008-12-10]. Prieiga per internetą: <http://www.arg0.net/encfs>

[5] eCryptfs - Enterprise Cryptographic Filesystem [interaktyvus], [žiūrėta 2008-12-10]. Prieiga per internetą: <http://ecryptfs.sourceforge.net>

[6] TrueCrypt - Free Open-Source Disk Encryption Software [interaktyvus], [žiūrėta 2008-12-15]. Prieiga per internetą: <http://www.truecrypt.org/docs/intro.php>

[7] Trusted Computing Group: TPM [interaktyvus], [žiūrėta 2008-12-15]. Prieiga per internetą: <https://www.trustedcomputinggroup.org/groups/tpm>

[8] Encryption HDD FAQ: Fujitsu [interaktyvus], [žiūrėta 2009-01-05]. Prieiga per internetą: <http://www.fujitsu.com/us/services/computing/storage/hdd/encryption-faq.html>

[9] Seagate Technology – Momentus Hard Drive Family [interaktyvus], [žiūrėta 2009-01-05]. Prieiga per internetą: <http://www.seagate.com/www/en-us/products/laptops/momentus/>

[10] Hitachi – Safeguarging your data with Hitachi bulk data encryption [interaktyvus], [žiūrėta 2009-01-05]. Prieiga per internetą: <http://www.hitachigst.com/tech/techlib.nsf/techdocs/74D8260832F2F75E862572D7004AE077/$file/bulk_encryption_white_paper.pdf>

[11] LaCie – d2 Safe Hard Drive [interaktyvus], [žiūrėta 2010-0426]. Prieiga per internetą: <http://www.lacie.com/intl/products/product.htm?pid=11301>

[12] IronKey: Technology [interaktyvus], [žiūrėta 2009-01-09]. Prieiga per internetą: <https://www.ironkey.com/technology>

[13] BOLLE, Ruud M., et al. Guide to Biometrics. New York, 2004. 353 p. ISBN 0-387-40089-3

[14] ULUDAG U., et al. Biometric Cryptosystems: Issues and Challenges. *Proceedings of the IEEE*, 2004, Vol. 92, No. 6, p. 948–960.

[15] ADLER, Andy. Vulnerabilities in biometric encryption systems: Audio- and Video-based Biometric Person Authentication. *AVBPA 2005*. New York, 2005.

[16] CAVOUKIAN, Ann; STOIANOV, Alex. Biometric Encryption: A Positive-Sum Technology that Achieves Strong Authentication, Security AND Privacy. *IPC – Office of the Information and Privacy Commissioner/Ontario* [interaktyvus]. 2007, kovas [žiūrėta 2009-05-15]. Prieiga per internetą: <www.ipc.on.ca/images/Resources/bio-encryp.pdf>

[17] BODO, A. Method for producing a digital signature with aid of a biometric feature. German patent DE 42 43 908 A1, 1994

[18] JAGADEESAN, A.; DURAISWAMY, K. Secured Cryptographic Key Generation From Multimodal Biometrics Feature level Fusion Of Fingerprint And Iris. *International Journal of Computer Science and Information Security*, 2010 January, Vol. 7, No. 1, p. 396-305. ISSN 1947, 5500.

[19] Biometrics Catalog [interaktyvus], [žiūrėta 2009 05 17]. Prieiga per internetą: <http://www.biometricscatalog.org/Introduction/default.aspx>

[20] The NSTC Subcommittee on Biometrics and Identity Management. Fingerprint Recognition. *Biometrics.gov* [interaktyvus]. 2006 rugpjūčio 7 [žiūrėta 2009 05 14]. Prieiga per internetą <http://www.biometrics.gov/Documents/FingerprintRec.pdf>

[21] THAY, Raymond. Fingerprint Image Enhancement and minutiae Extraction. The University of Western Australia. 2003. 63 p.

[22] BORO, R.; DUTTA, Roy S. Fast and Robust Projective Matching for Fingerprints Using Geometric Hashing. Department of Electrical Engineering, IIT Bombay, Powai, Mumbai [interaktyvus], [žiūrėta 2009-05-19]. Prieiga per internetą: <http://www.cse.iitb.ac.in/~sharat/icvgip.org/icvgip2004/proceedings/br6_289.pdf>

[23] ZHANG, W.; WANG, S.; WANG, Y. Structure matching algorithm of fingerprint minutiae based on core point. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China [interaktyvus], [žiūrėta 2009-05-19]. Prieiga per internetą: <http://www.ic.sunysb.edu/Stu/sewang/papers/JAAS03-matching.pdf>

[24] BHANU, B.; TAN, X. Fingerprint indexing based on novel features of minutiae tripletas. *IEEE Transactions*, 2003 gegužė, vol. 25, issue 5, p. 616 – 622.

[25] JULASAYVAKE, A.; CHOOMCHUAY, S. An algorithm for fingerprint core point detection. Faculty of Engineering, Research Center for Communication and Information Technology (ReCCIT), King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand [interaktyvus], [žiūrėta 2009-05-28]. Prieiga per internetą: <http://www.kmitl.ac.th/~kchsomsa/somsak/papers/isspa_07.pdf>

[26] COSTANZO R. Ch. Biometric Cryptography: Key Generation Using Feature and parametric Aggregation. Shool of Engineering and Applied Sciences, Department of Computer Science, The George Washington University [interaktyvus], [žiūrėta 2009-05-28]. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.329&rep=rep1&type=pdf>

[27] MATLAB Central – File detail – FingerPrint Application [interaktyvus], [žiūrėta 2010-01-26]. Prieiga per internetą: <http://www.mathworks.com/matlabcentral/fileexchange/16728-fingerprint-application>

[28] Fingerprint Verification System by Shivang Patel [interaktyvus], [žiūrėta 2010-01-26]. Prieiga per internetą: <http://fvs.sourceforge.net/index.html>

[29] Advanced Source Code . Com – Fingerprint Database [interaktyvus], [žiūrėta 2010-03-30]. Prieiga per internetą: <http://www.advancedsourcecode.com/fingerprintdatabase.asp>

PRIEDAI

1 priedas. Sugeneruoti šifravimo raktai

Vienas sugeneruotų šifravimo raktų rinkinys buvo pateiktas pagrindinėje darbo dalyje (*6 lentelėje*), likusieji pateikiami toliau. Ne tiek svarbu iš kokio atspaudų rinkinio gauti konkretūs raktai, todėl pateikiamos tik lentelės. Kiekviename naujame puslapyje esančioje lentelėje surašyti raktai, sugeneruoti iš vieno piršto 8 skirtingų atspaudų.

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	700f2209ddb59139 b0ef4088d1955842 aeea0e716e82024c fb65000ff971ae73 673760d027bda405 606651cbfa60422c e46efd5650c5dddf 9347142ea3870d51	33cc77812dd5a289 91e2da5d18841362 45d6a9df2ca838ec 34cfdec840692442 9d1cabe0369818f6 7d6774afbeaca7b3 f41b6e15fd8807ff 15a2ad3188a439aa
2	777072603d3b1913 7b6e64684d292524 4a4e303726281014 4f4630303f271a17 5633363d222b1a10 6666554c3f262412 6e666f55453c3d1d 494441323a282015	333c3738323d2a18 493e3d3531382116 343d3a3d222a131e 433c3d3c24261214 39313a3e3339211f 4746373a3b3a2a1b 4f4136313f38302f 313a3a33282a231a
3	696d6ec7e898520a 40d8846a59e273fe 538afe77268eef7c 487c9ebff6f0f6a6 f70cd630746d3ca1 e9a37852887e1864 fe0d44d9809fe677 15f7000d62e076e2	a7adf428b500b521 e8900066f43b7e7c 6c5745e270d05d61 f73c2cc69342ac8a cd0ab792b3c05b31 c743640c42378667 4acd23a436b54d44 5ccecd39b44c24d4
4	9a9b9bb976661002 98ba655656745033 5466775545233313 9657676775303121 7547754451130320 baa8565466570211 b747557664677111 8575444754701130	6567754621002100 7664445571021313 5755557410301310 7547477560102322 7746656020301200 7554554754012111 5677446545211311 5777774621130131
5	700e0ff022440990ddbbb5ad9189399c b00deff740028811d18b95a9581a4242 ae75ea570e70718e6e76824102404c32 fbdf65a600000ff0f99f718eae7573ce 67e637ec6006d00b27e4bdbda42505a0 60066666518acbd3fa5f600642422c34 e4276e76fdbf566a500ac5a3ddbbdffb 93c947e214282e74a3c587e10db0518a	33cccc3377ee81812db4d5aba2458991 9189e247da5b5dba1818842113c86246 45a2d66ba995dffb2c34a815381cec37 342ccff3de7bc8134002699624244242 9db91c38abd5e007366c98191818f66f 7dbe67e6742eaff5be7dac35a7e5b3cd f42f1bd86e7615a8fdbf881107e0ffff 15a8a245adb5318c8811a425399caa55
6	7e70700f702264093cdd34b518911639 72b06cef6440628840d12c9526582642 42ae40ea3c0e3671266e24821a02144c 4cfb40653c00360f30f92a7116ae1273 54673c373c6038d02e2726bd16a41405 6a606666565148cb38fa2e602e421a2c 6ce4646e62fd5256405038c536dd14df 4c9348474214342e32a32487200d1651	3c3338cc38773281302d30d528a21889 40913ce23ada3a5d381830842c131462 3c4536d636a930df282c24a81c3814ec 40343ccf36de30c82a40286918241242 3c9d3c1c38ab38e0363632982618l6f6 407d40673e743caf3cbe32ac24a71ab3 46f4421b3e6e381538fd3688320728ff 3e1536a230ad30312a8824a4243916aa
7	699f6dde69ecc776e6859880532b07af 4b04d3858c4567a15191e2297236f9e3 563e8da8fbe47876256185e7e5f979c3 448c78ce9deeb6fbf661fc0bf56caa6e f97207c9d8693f0a72426bdb32c8ae10 eb9ba03d7b825921818773ec1b81644d f7e90cdb4f49d19d810c9bf3ed627a7c 1b51fc7b07010cd86c20ed077c60e62a	af7aacdefa4d2c89bd56010bb75f2c13 e680980c04066d62f3443cbf7ae775c4 64c8577b4c59ea2c7102d60b58d46418 fa703dc32bcec16196314029a1cb85a3 cad305a5b1759121be3dcc0257ba3a17 ce73433d694e02c2402a3772816f6976 47a8c9d6293da0473c62b65d45d04f47 51c2ccecc7d3379db3404bcf2642d748
8	9699969d868e8c877e7879683532101a a4909d887874767a75696e6247232f2e 7573787a6f6e57474246383e2e2f171c 8478777c796e6b5f4f463f303f261a16 6f67606c5d5653505734363d232c1a11 8e898a83777875626868574e31282614 8f7e707d74747d796860695f4e363717 81757f776060605d46424e3037262e12	7a776a6d5f5442483b3530303b352211 6e686960605046464f34333b373e271c 565c555744454e4237303d30252d1611 5f57535c424c4c46493334322a2c181a 5c5d404a4b4749323b333c30353b2311 6c5754534644404c4442333738362617 646a6c6d52534a4443463b35343d3424 756c6c6e6c6d53593b34343c22242d14

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	4367f12e69884a6a 2c19d11d4e981e72 7b926f1c008ed53a 6259ea4ea0f8ac51 1bec3cb102dd7515 5d8327977a6a5642 7184fc8ceb04aaf2 60e9f80f72ee33c7	c780004e374efd60 29cb219fc0cb3bfb ecd023a7baba1246 560ec5b1fc071927 9728f521e6a75595 b0c71445713b8a76 075f072777d4c12d 914326b60426359d
2	54565f3236383426 92817d7174292127 7769565140483d33 46454e443a3f2a15 414e333b302d2711 5548423937262514 87685f583e303a2f 765e5f50473e231c	4c38303423141f16 424c42493c3c232f 4e3d322a2b1b1114 45302c2b2f101102 39322f221e1a1519 3b3c312427131817 50454032271d1c02 6954525b40423329
3	892349948a37ac9b 8f1810c254f47c49 e2f6e9ae6750e7b2 a89181cb11a05b60 eba9bacab8d5a71b f28f8b99f2c310fe 22bbc253bc3c24e3 a893f824a345c381	7cce11bea6e313e4 4453a05ad7c467ea fcdf7610ed2abb14 06b4ea4d3ea8def0 f00d52893c9b84f3 0ede9644f058b7cb cdd5709de3e05ba0 452e41e0943c034a
4	aa88966566412322 ab8a88b899755312 b8b9ba6755547520 aaa4647644641210 7666667666312102 7467666674700033 88aab898a7470130 aaa8b64564557020	5777446761300031 5554645675751132 b777554473022201 8565765743223330 7447542203222130 4777655530122132 bbb9546774701220 9987547465474412
5	43c267e6f18f2e74699688114a526a56 2c341998d18b1db84e7298191e78724e 7bde92496ff61c3800008e71d5ab3a5c 6246599aea574e72a005f81fac35518a 1bd8ec373c3cb18d0240ddbb75ae15a8 5dba83c127e497e97a5e6a56566a4242 718e8421fc3f8c31ebd70420aa55f24f 6006e997f81f0ff0724eee7733ccc7e3	c7e3800100004e7237ec4e72fdbf6006 2994cbd321849ff9c003cbd33bdcfbdf ec37d00b23c4a7e5ba5dba5d12484662 566a0e70c5a3b18dfc3f07e0199827e4 97e92814f5af2184e667a7e555aa95a9 b00dc7e3142845a2718e3bdc8a51766e 07e05ffa07e027e477eed42bc1832db4 918943c22664b66d0420266435ac9db9
6	5c43546754f13c2e3a693688364a206a 942c88197ed17c1d724e2c98221e2072 747b60925e6f561c4800468e32d5323a 4e62465942ea404e34a030f820ac1051 481b42ec363c36b134022cdd20751015 545d428340273c973a7a2e6a2a561242 82716a8458fc568c3eeb300430aa2cf2 74605ce952f8500f447234ee2a3314c7	42c73a803600344e28371c4e1afd1a60 4c294ccb4a21489f3ec03ecb283b28fb 44ec36d036232aa722ba16ba12121046 44563a0e2ac528b126fc1c0718190a27 349732282af520211ce618a712551095 3eb03ac73214284524711e3b1c8a1676 5607465f44073627227718d410c10a2d 60915c43542652b6480442263835269d
7	809e2231409f934589ac3774aece90b0 84f811871d07cc295542fa4777c2479b eb29fb6eef9ba1e7627a5704e876bd22 ab8e93108718cdb91b13a40759b1660b ebb5a99fbbabc2a0bb8cdd52a27d16b6 f62188ff8fb8989ffc22c6361909f6e2 2a23b0bcc22a523fb1c43bcb2a43e931 aa849c3df1812f4eae38475dc03d8216	7fc9c4ec1619bee2a963e3311a31e042 444d5438a90658a9d375cb496975e4a4 f6c4d4f8736a1702eed225a9bcbc1448 066dbd44edae46db31e9a08cd7e2f402 f00800d35c298d9d35c693bf8445f333 08eedeee906e4942f2095687b37dccb8 c9d6dc567a0c9ddfe73ee70656b3a00c 4e5624e44212ea0a9c4e31c00e3140a7
8	8889828384797974585a53373a3c392b a8afa1a891909c9295847f74772c2429 ae929f968e897a7e766755504e473b32 8a88897178615c5b41414a40353b2610 7e7b6a595b5a5c4a4b483d353a27211b 7f72686f686b59595f424c3331202f1e a2a2abab9c9295938b6c535c32343e23 8a8889838f7872747a5354554c332811	777c7c7e61615b4e4a363e3321131e14 747475635a50554a4d474c4436372e2a 8f6c6d5f575651404e3d322a2b1b1114 80766b545e5a444d433e2a282d1e1f00 5f50404d45423839333c292b18141f13 605e5d4e494644443f3035282b171c1b 9c9d9d857770696d5e434e30251b1a00 8485827e74716e606954535c4043342a

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	c9f80e11efe0eca2 644d9fa8b974c076 b8a4a913c3898431 83fa7283cc2a558b 07fba0b804b21bfc 516b5767152ebcab 1c6992e00f6d51ae fa29eca41dd76940	2520c4a2ee27ed22 0928d21bd324e8f9 0571439142d67985 81fcec5ee727b8e2 f1e25af1463d53b8 9a2e772a48dd8cc5 e2d2b174c01610d3 86a842987d0446ec
2	7c6f60515e4e2e1a 4644494a3b272c17 7b7a7a616c686813 686f67583c322518 505f4a4b403b212f 8586858671727b2a 6166594e3026252a 6f625e5a512d2614	62626c6a5e524e22 40423d312d221e1f 50474449343d2728 585f4e454e323b2e 5f4e453f3433252b 59423732343d281c 5e5d4b473c31212d 383a34393720241e
3	fedaae20cf11feea 9a7f478176bbc8d8 a2f1011621183ff9 e3d94cdd5d45a026 ab8d4ee00fab0b1f b4904882798945ed 05a573bf168d064a bdfab8a87a62a5ec	fea22678ba527b7a 52042dd89b6a6b78 01cddf244b8d81cc e0ebe6f5d54b3703 946ffa5432a3889f f632dd292c0bd725 a1d0d01421fb0551 4d5935b09b4a815f
4	bbbaab8477447732 aa97556455663232 a8b8888544464772 b8b6577757552001 aa67577447664203 a9a89aa89aaa5573 8965546745670112 abbaaa6656546133	bb64455666545652 5445477666121212 4477774556672033 b876757575564100 6557765544602223 b984774647023101 6474744544760110 5756456422122013
5	c993f81f0e701188eff7e007ec37a245 64264db29ff9a815b99d742ec003766e b81da425a99513c8c3c389918421318c 83c1fa5f724e83c1cc332a5455aa8bd1 07e0fbdfa005b81d0420b24d1bd8fc3f 518a6bd657ea67e615a82e74bc3dabd5 1c3869969249e0070ff06db6518aae75 fa5f2994ec37a4251db8d7eb69964002	25a42004c423a245ee7727e4edb72244 09902814d24b1bd8d3cb2424e817f99f 05a0718e43c291894242d66b799e85a1 8181fc3fec375e7ae7e727e4b81de247 f18fe2475a5af18f46623dbc53cab81d 9a592e7477ee2a544812ddbb8c31c5a3 e247d24bb18d742ec00316681008d3cb 8661a815424298197dbe04204662ec37
6	70c96af86a0e5e1158ef4ce028ec12a2 4e644a4d469f44a836b9287424c01276 76b872a472a96c136cc3688962841231 688364fa6272588330cc302a2255188b 520752fb4ea04eb8440436b22e1b2efc 8e518e6b865782677e157c2e76bc26ab 641c646958924ce03c0f2a6d245120ae 6afa60295cec58a4501d2ed72e691840	6c25642062c460a256ee542744ed2422 420940283cd2341b26d322241ee81cf9 58054c7146434491384236d626792085 5c8152fc4aec465e42e7302730b822e2 54f14ce24a5a3cf13a46303d2c5328b8 5e9a4a2e3c773a2a3a483add228c18c5 5ae252d24cb148743cc036162e1024d3 3e863ca83c423c98387d280428461eec
7	f9eadea3aeeb2202ccfb1114f2e2eea5 9bac74f84f7584137760b2bbcc87d389 a229fb1100101b6e2612188231f8f29f ee38d89441cedfd95dd34c5da6042e64 a5bb84d747e2ec0209fda2bb06b41dfe bd4195094489832c7d978393415be8d7 0d57a95a7135befd19678fdd0c6a4eac b8d7f9afb882ae837dad6f27a05becc3	fce9ab2b236f738cb1ac502e7bb379ae 5e2c094c25d3d18d97b660a961b17687 0c13ced2d8f125464cb889d7891dc0cc e304e9b5e467f056d85342b53d7d0e39 9a476ef8fda05a463828a0388c8499fe fd613a24d3d928932dc10abddb712058 a911d907d50a10422414f3b602585216 4ed657973f54b80c93b54fa5891153f9
8	9f9e9d8a8a8e82707c6f61515f4e2e1a 898a877f7477785147464b4b3c282d18 aa929f9190818176727171686 36f6f19 8e838d79747c7d6d656d64553a302216 8a8b787d646e6e50505f4a4b403b212f aba4a9a0a4989892878988897 4757e2d 80857a7577736b6f6166584d3026242a 9b9d9f9a8b887a78676a56525a252e1c	8f8e7a72727677686b6a6562575b472a 65626054525d4d48494b363a262b1718 60616c6d6d6f5254544b484d38312c2c 8e807e6b6e666f655d55444b43373023 7974766f6f6a655453424a333838292f 8f8683726d6d6259524c303b3d372215 7a716d606d60515452514f4b30352521 646d655943454b40393b343a3821251f

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	2f49560861f984ac f1d700168d9bf8c9 2508997db1a5fe7c 85be3672d333aea0 c901c5e14c5bf6a1 062149d6a427a2ca 8a3609aa93a0b6d1 bf9b910e741b5e08	e035986aa1dcdd88 181209db4cdc65ae 2896872741768f87 1fd873a8c9c0df68 b8e1b32e5149c36e e2fdccb80f8269b0 b3aab618cac913bd f68a825afd9f0351
2	42444530363f282a 6f5d404138291f1c 626069575b4a2f17 484b33372d232a1a 6c504c4e44453f2a 4042443d3a221a1c 5853404a393a1b1d 7b69595047311510	8e7379765a4d3d28 7171706d544d362a 5259584234372828 515d474a4c4c2d16 4b4e4b4235343c26 4e4f4c2b2018161b 5b5a4b313c2c212b 3f3838252f191015
3	29187b27acc8d7f2 fa9feedaa8ac34a7 89a2809d0e7588d5 98e82dbe7a25b198 7a9d4f295957de83 7eade30779b41913 3cf8b350941baace d5efccad976e4f3d	0ed5f5f75a0e155f 61409422332f6f8c 6ec57d3e07602576 912008adfa4799b3 0ce587c2d0d576e9 21d0fae101ed2a8c 8ed9d112dcd3ee3d 3152facdd653d7d2
4	4646564567723130 baa7777666670121 6664646747556631 6676476756012022 9aa7574656557720 5767744556610200 4776645465462233 b9bbbb6765575303	8bb9b9b996474513 9898a94444475723 5775574745540111 6444466776556620 8bb9657474751132 4474767444730223 6776744477703303 4454767731103130
5	2ff44992566a08106186f99f8421ac35 f18fd7eb000016688db19bd9f81fc993 25a4081099997dbeb18da5a5fe7f7c3e 85a1be7d366c724ed3cb33ccae75a005 c9930180c5a3e1874c325bdaf66fa185 066021844992d66ba42527e4a245ca53 8a51366c0990aa5593c9a005b66dd18b bffd9bd991890e70742e1bd85e7a0810	e00735ac98196a56a185dc3bddbb8811 181812480990dbdb4c32dc3b65a6ae75 2814966987e127e44182766e8ff187e1 1ff8d81b73cea815c993c003dffb6816 b81de187b3cd2e74518a4992c3c36e76 e247fdbfcc33b81d0ff082416996b00d b3cdaa55b66d1818ca53c99313c8bdbd f66f8a5182415a5afdbf9ff903c0518a
6	4a2f424942563e083a6138f92e8428ac 6af156d74e004616368d2e9b1cf814c9 6825660860995e7d50b140a528fe1a7c 4e854abe363634722cd3243322ae10a0 64c958014cc546e1444c405b32f62aa1 4c064a2140493ed636a422271ea218ca 568a54364c0942aa389334a01eb61cd1 70bf669b5c915a0e4e74301b165e1008	80e07a357898746a5ca14edc3cdd2888 78187212720966db5e4c48dc3a652cae 5828549650874a273e4130762e8f2487 5c1f52d84c7348a842c940c02adf1c68 4cb84ce144b3402e3c513a4930c3286e 4ce24afd42cc28b8240f1c82166914b0 5ab358aa4eb63e183cca2ac9281328bd 3af6388a3282225a22fd169f16031251
7	2e95198a7cb82d78a8c2cf81da72fd25 f8a795fcece4dcaea086afc53c49a778 809dae20840e94da00e273588b80d856 9884ea8b29dbbbea70a9205dbe1e988b 70af9dd140f12d915b925672ddec8732 76eaa5dde53d02727893bb491799143d 3dc7fa80b538560e994418b9a2afc4e0 d25cebfdc2cea2d6957167e44af134de	06edd554f852f87656ab0ee0175253fe 6f134c099b482320383229fb6cfc85ce 64eec2537bd138e40c7a6b0c255b736c 9e1d24020486acd2f1aa457b9b92b13b 0ecbe4538a78c923dc04d651756ce792 2310d609ffa3ed1e011cebd62ea088ce 80eed894dd1f1a20d3cbd638ebea33dd 331f542ef9afcddad1645c34d779dd2c
8	62696158575b52474a4c4c383d372f22 8f8a897f7e7e7d6a6a584a4c33241a17 78797a727870696d606e675558482d15 79787e68625d5b5e474a32352b212918 a78a897d747f7269655945474d4e3823 776e6a6d5e53505747494b3431291113 737c7f686b6365505954414b3a3a1c1e ad959e9f9c8c7a7d7967565e443f131d	909e9d959f858f87857a707e5145352f a691848089847272737372 6f564f382c 767e6c65676d535e5057564032352726 7971726060686a5d5f5a444749492b13 908c8e8578776c524d404d4537363e29 72717d606f6a6e5140414e2d221a181c 787e7d797d6161625d5c4d333e2e232d 737155525f5a4c4d3d3635232d171d12

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	134c981aead673d1 d3c487b9c3a7dd5d bdf5844debf6c488 e71b4e45c03d1841 da8162ba4256091d d6b2e74684a5f33d f00a011b7642de8a 4e491349fd613e07	5505c35b73c4784f 70a754c091b63b89 d47d5b552477c8e5 c28807a0b4b64470 f135a8ae22a4a758 2b3016fa31dc3d71 efe705b62328bbc4 e2d0c4351ae33838
2	414449413e2d271d 8d6c684b4c3a2d05 5b4f48444e3f1c18 7e6164544c333114 5d58563b34352021 7d7b6e64484a3f23 5f50404147343d38 746451343f363310	75706c65676c5734 877a756c694b4328 7d67656562372c1e aca8807a5b4b4437 bfa3aa9a828a6a25 9283817f636d6327 7e7e706b62523b3c 6e5d3c23212e1313
3	f346951f1491ed7c 40e1b7becb7ab9c4 5cdf8d1ead63cda6 c9ecf89647ba2a7a 2d650ea1f97d5623 af2cfc20dbe48af4 a2317d08798afc61 8592bb4498584a84	66d8e5fc1c823831 187847de583a7916 f696fe7db53305ac 03395e0c85d68813 f3ccd2398d44b44f f06cc979cca9c7d0 5ceffb01002d44de fe6b009ca980ea00
4	b855654745643313 98b8a9abb6566231 9b77674767547321 babbba6555664212 4755476476531100 ab8bbb8876756631 6444574656667310 a9a8aa9566521221	99bab9b747644640 8a9a99bb96465641 b9a9bb5765444123 888a9b8ba9b56640 fcffb88aab99a953 f89bba9abba67570 9bbbba8844475533 bb56446766203200
5	13c84c3298191a58ea57d66b73ced18b d3cbc42387e1b99dc3c3a7e5ddbb5dba bdbdf5af84214db2ebd7f66fc4238811 e7e71bd84e7245a2c0033dbc18184182 da5b81816246ba5d4242566a09901db8 d66bb24de7e746628421a5a5f3cf3dbc f00f0a5001801bd8766e4242de7b8a51 4e72499213c84992fdbf61863e7c07e0	55aa05a0c3c35bda73cec423781e4ff2 700ea7e5542ac0039189b66d3bdc8991 d42b7dbe5bda55aa242477eec813e5a7 c243881107e0a005b42db66d4422700e f18f35aca815ae752244a425a7e5581a 2bd4300c1668fa5f318cdc3b3dbc718e eff7e7e705a0b66d23c42814bbddc423 e247d00bc42335ac1a58e3c7381c381c
6	4e134e4c4698421a38ea2ed6287310d1 80d36cc46a874eb94cc33ca72edd0c5d 5ebd4af54484444d40eb38f61ac41088 72e76e1b684e544540c0363d32181641 52da508150623cba3a4230562e09281d 7ad672b268e762464e844aa53cf3283d 56f0540a4601401b40763e423cde3c8a 724e604952133c493cfd3a61383e1007	725570056ec36e5b627362c45e78324f 827074a772546ec0629144b6403b2c89 70d46e7d665b625562243a7728c81ee5 acc2a4888c077ca054b446b640443070 b2f1ae35a8a898ae8e228ca466a72458 942b8e308e167cfa6c3168dc603d2871 7cef72e770056cb662235e283cbb38c4 64e252d03ec42a35281a28e31c381038
7	f0334f62965110fd114a9618e8d371cf 4b0fe515bf74b8eec2b275a7b196cb4b 54c6d3f88cd813e3a0d86731cedaa86c c39de2c8f68f9c62497eb1a822a07ba4 2bd462580be9a318f2997ad25b6e2135 adfb2ccffdc02803dab6eb4987a9f740 a923321674d30488789389a4ffca6713 87539922b7b14e42908c558c40a38049	6c69d889e958f8c71ccc8a223a8b3f16 1e8276864279d8e0508734a071961b69 f8609663fbe27bdbb05836310053a4c1 0a3e30915aed0ccb8d53d36b8082103b fc30cac7d42c339c8dd1444abe4043f4 fd0962c4c79d7698c4c9af93ca75d60a 56cde0fdf7ba071d0d0423d34146d9e2 f3ed6cba050993caad9b8f00e4ae0302
8	9f8374767965515f414449413e2d271c a4a0aea19b978b8e8c6b674a4b392c04 858c7d7f787d716e5a4d46434c3d1a16 ac999e8c8f88797674676b5a423a371a 727d6665605e5a515f59573d35362223 baaf929c9f9c82807d7b6e64484a3f24 7a727361676d60585759484a4f3c3631 a8a5a9929b8b847479685538343a3814	b6b6bd989e858f7c717c686263685331 a198979894978d8e8578736a67494126 bfa6a9a69f9e777d7b65636360352a1c b0b3b3b9b5beb0bca8a58d7658484133 cfc3ccccbdb2b3b9b8ada4948b84642f cfb0b6bcaca9a7999c8c8a796c676d20 a5acaeafafab90817070726d64543d3e 9f8e767b7070797c6a5938202e2a1010

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	6c647e2704d44a31 9f67ecc5e057c0c7 5e46e3c8259bb283 561376d07b594be1 bd6cc6591a53f036 536b039b5bc93a84 a4a962c9d37eaf6a f9335e2972b64e20	d21d4d51161b962c 7e4ff1f716e03012 d3181d7baf94f17e 60eb1053312207b8 9ec1f68929fba5c9 63b76a2ef109fed0 10f28262692dd398 f9335ed3ccb64ed3
2	56564742302d2413 59464e3c3e352c0c 45343e2c22292b18 5551573d3735241e 7b766c6551452f03 45463039352c2318 6a6a565c4d372a16 6f535542373b3422	8d81747561612922 57545f5f313e3321 5d5141473a292f27 565e41353322201b 494c4f38323f1a1c 464b36323f303f2d 515f584646322d19 6f53555d5c3b342d
3	cc5e9498fe0a85db aa11bc960d534c23 c9035494a8216efc 869e9a29ea061fe7 4a323129506fbe8d 3e062617de81d4b0 23cb9f2833f5603f 450fdef659c8d1b8	509e8d5dae12ee6f 531eda45da446386 041d0780d117b9f8 8be2e227d58ca972 71a464782481b835 a0d11795b1c84652 8f5e3b5efd6441b7 10a0b8a1047eec6e
4	bb9b656677462132 6644676547501300 7644556562001333 6567664676410331 9a88888654576723 4745454577203120 88ba674644755003 9987777556763022	98abab9bab447713 9447765576551021 4547456474452232 aab4744575632210 5465555645602201 6474456564321110 6757465777555021 88a8a66445577313
5	6c3664267e7e27e40420d42b4a52318c 9ff967e6ec37c5a3e00757eac003c7e3 5e7a4662e3c7c81325a49bd9b24d83c1 566a13c8766ed00b7bde599a4bd2e187 bdbd6c36c663599a1a5853caf00f366c 53ca6bd603c09bd95bdac9933a5c8421 a425a9956246c993d3cb7e7eaff56a56 f99f33cc5e7a2994724eb66d4e722004	d24b1db84db2518a16681bd896692c34 7e7e4ff2f18ff7ef1668e007300c1248 d3cb18181db87bdeaff59429f18f7e7e 6006ebd7100853ca318c224407e0b81d 9e79c183f66f89912994fbdfa5a5c993 63c6b7ed6a562e74f18f0990fe7fd00b 1008f24f8241624669962db4d3cb9819 f99f33cc5e7ad3cbcc33b66d4e72d3cb
6	586c56644e7e4a273c0428d4244a1e31 549f4e6748ec3ac532e0305728c00cc7 405e384636e32ac82625249b20b21a83 5856561350763cd0347b30592e4b18e1 7abd766c6cc66a595c1a485326f00e36 4c53446b3c03329b305b2ac9263a1084 6ca464a9586250c94ed3327e2aaf106a 64f95c335c5e44293a7238b6304e2e20	82d2821d7a4d74516c16681b2696262c 5a7e564f56f152f73a163ae034302a12 56d35018461d407b36af2a9426f1207e 5c6056eb4a10345330312e22260718b8 429e42c140f63c89362930fb1ea516c9 426340b73e6a322e32f1300930fe28d0 561050f2508246624269302d2ad31898 64f95c335c5e54d354cc38b6304e2ed3
7	c7c951e79b439887f3ec05ae8c5bd1b8 ada1181eb6ce9d6e0ed65b344ec62f36 c495053c5e4f9448a78e2c116ee3fbcc 846f95e79ba72298eeac0e6914f2e47a 4ca5362f3f162c9b580761f4b5ee8bd1 34e1016c2662147cd0e78017d845b600 2338c9b69af52c8d3137f056600c3df8 455a07fdd1ebfa645d98c28dd61bb384	5d0f9cec84d45dd1a3ee1e22e7ec67fd 533317e0d4ac4c51d0a1424968318264 04491add057e8301d4191e7db196f380 89b1ee21e1212673d35e83c6a4967b2c 7611ac4e6348728b294c8215b4863155 a604d21318729758bc10c38749625729 80fb56ef32b25ce4f5d767474e12b87d 1f0aa102b881aa140e4873e7e1cb63e7
8	9c8c858e797469685f5e404a38252d1b 7a7a61616b5c5956504d4533343c2203 7c797063655459444a383221262e2f1c 7876796e696a52595e5a5036313f2e17 949a9392838182797570666f5b4e280d 737e6056525651474d4e38313d242b10 92939c9b797f726863635f554630231f 9485807f7d7e6f6665595c483d313b28	a5a0a9ae989d958d8a8e71726e6e262f 8573717e7d6a64655d5a545436333826 7074716d606768505d5141473b292f28 888b8e726e6262675d55483c3a292712 67616a64564447484244483 13b381315 7a706d61616759454b413c3834363522 787f757e737b655e5f5d564444312b17 81808a808b786a616054575e5e3c362e

Generatoriaus Nr.	*Matrica formuota su*	
	FingerPrint Application	*Fingerprint Verification System*
1	7f1693bbfe348bdd ccf5aeae4e68c0c2 10ddb09a76a4f769 373b7bb0cc809db4 12953907f3d25129 a9f111e9c1607767 6ef1c178c0b6da3e 70c5381557643631	4c290e223d5d2211 443f2019b6bfc5b1 adf2f9111aac2e21 37bb2af73ad1d7b7 528e943dae74857d 7620db7bd4788d7e beed3afab50c8f59 28e7d2e209e8f970
2	5751595b4f43483d 6c5f5a4a44362c1c 615d5b59574a3f36 5343473b3c38290b 615953403f2d2522 5a5f414e4c363726 666f6c474c3b3d13 676c535155463323	6462403233251211 645342414b4b1c1b 4a4f3f31313a3232 737b726f433d2d1b 757869434a473837 67525d573d373827 8b8e737f4b402805 524e4d4e403e3f37
3	de949745d6e058d3 ef895cd2f2dd79ee 0471b36230dcad29 7ead7667aae24f12 984046aef71edb30 1a191920bf1fc687 e814e07180d9ddf5 a2d19543a0749976	44d8b1cb0ecef1ed aff005537755fffe c2e1c2ba59abc5dc 912345426e526f0e 59b9eea16aa5c899 050c655b2d938243 cfe1a132e162e3b8 a05eac0cea9acab3
4	bb65655575745670 7766577474775233 8998a85444776702 5767555566701300 aa98956775473200 8646464467477121 ba89b85464767331 a8b4655464556611	99baa47647733033 abb8855455557733 b474746656223133 a888955457545303 9aaab76456657622 898b955647642010 bbb8a888b8547422 6457674776667220
5	7ffe166893c9bbddfe7f342c8bd1ddbb cc33f5afae75ae754e726816c003c243 1008ddbbb00d9a59766ea425f7ef6996 37ec3bdc7bdeb00dcc3380019db9b42d 124895a9399c07e0f3cfd24b518a2994 a995f18f1188e997c183600677ee67e6 6e76f18fc183781ec003b66dda5b3e7c 700ec5a3381c15a857ea6426366c318c	4c3229940e7022443dbc5dba22441188 44223ffc20041998b66dbffdc5a3b18d adb5f24ff99f11881a58ac352e742184 37ecbbdd2a54f7ef3a5cd18bd7ebb7ed 524a8e7194293dbcae75742e85a17dbe 766e2004dbdb7bded42b781e8db17e7e be7dedb73a5cfa5fb5ad0c308ff1599a 2814e7e7d24be2470990e817f99f700e
6	547f5216509350bb4cfe4834468b3cdd 62cc58f556ae46ae404e38682ac01ac2 601058dd56b0509a507644a436f73069 5437463b447b3ab034cc3280209d0cb4 60125e9558394c0738f324d220512029 56a952f14e1148e948c1386036772067 6a6e6af164c1487848c03eb63cda103e 687064c55a38501550574c643e362431	6a4c68294c0e3a22323d2c5d1c221c11 6044563f4e204e194cb64abf1cc51cb1 44ad44f23ef93c113a1a38ac362e3021 783778bb762a6ef74c3a3ad12ed71eb7 7252708e66944a3d4aae40743a853a7d 64765e205adb587b3ed43878308d2c7e 88be86ed7c3a7cfa4cb54a0c2e8f0e59 52284ee74ed246e2440938e838f93670
7	d1e2964e9c784053d36ae70e5288df31 e8f980935ccede2cf822dbdb7b95ecee 07407117b1316b2e3f0cdfc9a4d32698 79e3a9db7e6c6173abafef2440f31128 98804e0c416ea5e3f6781debd6b6350d 13a318941f9b2f0ab3fc1cf3cc6b8271 e784144fe404721a890bdb93dbd0f458 ae24d518975c413aaa00724f919e706b	4441d687bb16ccbf0ce8cee1fd1de1d0 adf6fa0c025853397d785a53f0f8ffeb c42ee911c022b3ac599ea5bdc658dacd 9b152c3d44514a2c63e7562366fd03e3 5397b697eee9aa1366a2a851c2889991 025d05cc675e54b52cd59130892e4233 c2fae41ca416322eec1b6828e33abd87 ab035ce4acc802c0e8a792a2c9a8b930
8	8d8e7964696764555d565e5045484d33 7e7f7879656c6d626f525d4d47392e1e 908487818b83766263505d5c5a4d3239 777e7a6d676656575a4a4e32343f2102 9998948084767a6e6f57514e3d2b2320 817a7179616952505b5f414f4c363827 9e8881848e80777168606d494d3d3f15 8a828d71797564636a60575459493726	94948d888b717c7b606e4c3e3f211e1d 8a8f8f8080757573675745454f4f1f1e 8c626e616c524b4a45493a3b3c353d3c b9b1a2a3a4757472767e7562463f201e 95998b898e7e7a71767a6a454c483939 a0a5a09c8675656b625d595338323423 9c9f9e919a8183828e8176724e432b08 7a60655e5a5c505c5e4a494a4c3a3b33