Kauno technologijos universitetas

Informatikos fakultetas

Bio kriptografijos raktų generavimo metodas

Baigiamasis magistro projektas

Paulius Kazlauskas

Projekto autorius

Prof. Algimantas Venčkauskas

Vadovas

Kaunas, 2022

Kauno technologijos universitetas

Informatikos fakultetas

Bio kriptografijos raktų generavimo metodas

Baigiamasis magistro projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Paulius Kazlauskas

Projekto autorius

Prof. Algimantas Venčkauskas

Vadovas

Doc. Nerijus Morkevičius

Recenzentas / Recenzentė

Kaunas, 2022

Kauno technologijos universitetas

Informatikos fakultetas

Paulius Kazlauskas

Bio kriptografijos raktų generavimo metodas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Paulius Kazlauskas

Patvirtinta elektroniniu būdu

Kauno technologijos universitetas

Informatikos fakultetas

Bio kriptografijos raktų generavimo metodas

Kazlauskas, Paulius. Bio kriptografijos raktų generavimo metodas. Magistro baigiamasis projektas / vadovas prof. Algimantas Venčkauskas; Kauno technologijos universitetas, Informatikos fakultetas.

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

Reikšminiai žodžiai: biometrija, kriptografija, šifravimas, piršto atspaudai.

Kaunas, 2022. 47 p.

Santrauka

Šio projekto tikslas yra sukurti ir išbandyti biometrinį šifravimo raktų generavimo metodą. Tikslui pasiekti buvo iškelti uždaviniai:

1. išanalizuoti šifravimo raktų generavimo problemas ir metodus;
2. išanalizuoti biometrijos panaudojimo šifravimo raktų generavimui galimybes ir metodus;
3. sukurti biometrinį šifravimo raktų generavimo metodą;
4. įgyvendinti sukurtą biometrinį šifravimo raktų generavimo ir valdymo metodo prototipą;
5. ištirti sukurtą biometrinį šifravimo raktų generavimo metodo prototipą ir įvertinti jo veikimą.

Projekte procesas aprašytas 3 pagrindinėmis dalimis: esamų raktų generavimo metodų analizė, bandomo realizuoti metodo aprašymas, metodo prototipo aprašymas ir eksperimentai. Darbo gale pateikiamos išvados apie išbandytą metodą.

Lentelių skaičius (4 vnt.), paveikslų skaičius (22 vnt.), literatūros šaltinių skaičius (29 vnt.), informacijos šaltinių skaičius (1 vnt.).

Kazlauskas, Paulius. Bio Cryptographic Key Generation Method. Master's Final Degree Project / prof. Algimantas Venčkauskas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics Engineering (Computing).

Keywords: biometrics, cryptography, cyphers, fingerprints.

Kaunas, 2022. 47p.

Summary

The aim of this project is to develop and test a biometric method for generating cryptographic keys. To achieve this goal, the following goals were set:

1. analyze the methods of generating and managing cryptographic keys and their problems;
2. analyze the possibilities and methods of using biometrics to generate cryptographic keys;
3. develop a method for generating biometric cryptographic keys;
4. implement the prototype of biometric cryptographic key generation method;
5. investigate the developed prototype of cryptographic key generation method and evaluate its correctness.

The project describes the process in 3 main parts: analysis of existing key generation methods, description of the method to be implemented, description of the method prototype and experiments. Conclusions on the tested method are presented at the end of the work.

Number of tables (4 units), number of pictures (22 units), number of literature sources (29 units), number of informational sources (1 unit).

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Įvadas

Bendrai yra keli biometrinių kriptografijos raktų generavimo metodų tipai. Vieni metodai kombinuoja gautus biometrijos duomenis su jau esamais raktais, o kiti generuoja raktus naudodami tik biometrinius duomenis. Taigi, galima išskirti 2 pagrindinius tipus:

- raktų generavimas;
- raktų pririšimas prie biometrinių duomenų.

Taip pat, patys biometriniai duomenys gali būti skirstomi į dvi grupes:

- vienos dalies (Unimodal) – šie kriptografiniai duomenys sudaryti naudojant tam tikrą vieną sritį, pvz.: piršto atspaudas, kuomet paimami tik odos griovelių susikirtimo taškai;
- kelių dalių (Multimodal) – šie duomenys yra sudaryti iš keleto skirtingų sričių ar matavimų, pvz.: piršto atspaudai, akies rainelės atvaizdas ir veido parametrai.

Abu biometrinių duomenų tipai turi savų teigiamų ir neigiamų savybių, kurių palyginimai pateikti 1 lentelėje. Matome, kad biometriniai duomenys sudaryti iš kelių sričių yra brangiau ir sudėtingiau apdorojami, tačiau jų saugumas ir panaudojimo lankstumas yra didesnis. Vienintelis neigiamas dalykas, kad kuo daugiau skirtingų duomenų surenkama, tuo sudėtingiau yra užtikrinti, kad visi jie atitiks sekantį kartą nuskaitytus duomenis.

1 lentelė. Biometrinių duomenų įvertinimas[4]

Parametrai	**Vienos dalies (Unimodal)**		**Kelių dalių (Multimodal)**	
	Žemas	**Aukštas**	**Žemas**	**Aukštas**
Kaina	X			X
Užtikrintumas		X	X	
Atpažinimo tikslumas	X			X
Saugumas	X			X
Lankstumas	X			X
Sudėtingumas	X			X

Nagrinėjant kriptografinių raktų generavimo ir valdymo metodus, iškyla saugaus rakto sukūrimo ir valdymo/saugojimo problema. Sunku sugeneruoti saugų raktą, bei jį persiųsti kitai šaliai ar išsaugoti, taip, kad rakto neperimtų piktavaliai asmenys.

Taigi, šio darbo tikslas yra sukurti ir išbandyti biometrinį šifravimo raktų generavimo metodą.

Tikslo pasiekimui iškelti uždaviniai:

1. išanalizuoti šifravimo raktų generavimo problemas ir metodus;
2. išanalizuoti biometrijos panaudojimo šifravimo raktų generavimui galimybes ir metodus;
3. sukurti biometrinį šifravimo raktų generavimo metodą;
4. įgyvendinti sukurtą biometrinį šifravimo raktų generavimo ir valdymo metodo prototipą;
5. ištirti sukurtą biometrinį šifravimo raktų generavimo metodo prototipą ir įvertinti jo veikimą.

1. Bio kriptografijos raktų generavimo metodų analizė

1.1. Šifravimo raktų generavimo ir valdymo problemų ir metodų analizė

1.1.1. Bendros raktų generavimo ir valdymo metodų problemos

Viena iš pagrindinių šifravimo raktų problemų yra nepakankamai sudėtingas raktas, kurį lengva atspėti ar nulaužti. Daugeliu atvejų, rakto generavimui yra panaudojami tam tikri duomenys, pvz.: jei vartotojas nori užšifruoti tam tikrus duomenis, šifravimo raktui generuoti gali būti panaudotas vartotojo įvestas slaptažodis ar PIN kodas. Taip pat, slaptažodis ar PIN kodas gali būti tiesiogiai panaudotas šifravimui. Tiesiogiai slaptažodžių ar PIN kodų šifravimui naudoti nereikėtų, nes tai yra ganėtinai trumpas šifravimo raktas.

Kuomet naudojami vartotojo įvedami duomenys, sugeneruojamas reikiamo ilgio šifravimo raktas. Modernūs šifravimo algoritmai naudoja ilgus ir pakankamai saugius raktus, tačiau tokia pati duomenų įvestis, tam pačiam algoritmui dažniausiai duos tuos pačius raktus. Kuomet perimama vartotojo paslaptis – galima išgauti naudojamus šifravimo raktus.

Vartotojo žinomi duomenys panaudojami šifravimo rakto generavimui turi dar vieną trūkumą. Daugelis vartotojų susikuria tokius slaptažodžius ar PIN kodus, kurie jiems asocijuojasi su tam tikrais dalykais. Socialinės inžinerijos metodais, šiuos slaptus duomenis galima išgauti iš vartotojų ir taip išgauti norimus duomenis iš norimų sistemų.

Kita problema yra susijusi su raktų valdymu. Sistemose kur dvi pusės turi apsikeisti kriptografiniais raktais yra sudėtinga užtikrinti, kad apsikeitimas įvyks saugiai ir niekas neperims siunčiamų raktų. Saugiam apsikeitimui raktais, naudojamos kelių metodų kombinacijos. Prieš apsikeičiant šifravimo raktais, pirma yra vykdoma abiejų pusių autentifikacija. Vėliau raktais apsikeičiama kuomet abi busės jau autentifikavosi viena su kita. Tokiu metodu siunčiami raktai, papildomai gali būti šifruojami panaudojant autentifikacijos raktus. Tačiau net ir autentifikuotas raktų apsikeitimas negali garantuoti visiško saugumo.

1.1.2. Galimų būdų stipresniems raktams generuoti ir juos valdyti analizė

Rakto generavimas naudojant „atsitiktinio miško kirtimo“ algoritmą

Srautinis šifras yra naudojamas ilgoms duomenų sekoms šifruoti, kaip pavyzdžiui vaizdo transliacijos ar ilgi duomenų failai, kuriuos reikia šifruoti perdavimo metu. Naudojantis šiuo šifravimo metodu, sugeneruojama begalinė skaičių eilė, kuri yra naudojama duomenų šifravimui.

Ši sugeneruota seka turi kelis trūkumus, pvz.: sudėtinga užtikrinti, kad kiekvienam duomenų rinkiniui bus generuojamas visiškai skirtingas šifravimo raktas, sudėtinga generuoti nesutampančias sekas, net ir naudojant skirtingus pradinius raktus.

Pasiūlytas srautinio rakto generavimo algoritmas [1] naudojasi „atsitiktinio miško kirtimo“ (angl. Random forest mining) algoritmu.

- pirma, algoritmui yra pateikiama reikšmė n, kuri nurodo, kiek duomenų rinkinių bus generuojama, generuojamas skaičius gaunamas 2^n;

- šis algoritmas gautus duomenų rinkinius sudalina į 3 dalis. Dvi dalys panaudojamos sudaryti „pakavimo“ (angl. Bagging) rinkiniui, o likusi dalis sudaro „nesupakuotą“ (angl. Out of bag) duomenų rinkinį;
- abu duomenų rinkiniai panaudojami sudaryti duomenų medžiams. Kiekvienas duomenų blokas rinkinyje, sugeneruoja vieną medį;
- panaudojus duomenų kombinavimo algoritmus, apjungiami duomenys esantys „pakavimo“ medžiuose, gautas rezultatas yra ilga dvejetainių skaitmenų seka;
- duomenys, kurie bus šifruojami yra konvertuojami į dvejetainį formatą;
- jei raktas gaunamas trumpesnis, nei duomenų seka, kurią reikia užšifruoti, grįžtama į generavimo pradžią ir padidinamas n.

Gavus tinkamo ilgio raktą, duomenys gali būti užšifruojami.

Norint duomenis iššifruoti. Procesas pakartojamas, su antruoju „nesupakuotu“ medžių rinkiniu. Iš šio rinkinio išgaunamas raktas skirtas duomenų iššifravimui.

Diffie-Hellman

Pagrindinė ataka nukreipta prieš apsikeitimo raktais protokolus yra „vidurinio žmogaus“ (angl. Man-in-the-Middle) ataka. Stengiantis išvengti šios atakos buvo pasiūlytas toks metodas [4]:

Naudojantis Diffie-Hellman raktų apsikeitimo protokolu, yra apsikeičiama reikšmėmis, kurios vėliau naudojamos bendro rakto sugeneravimui. Norint apsisaugoti nuo siunčiamos reikšmės perėmimo atliekami tokie žingsniai:

1. abejoms pusėms yra žinomas skaičius e;
2. Alisa sugeneruoja slaptą skaičių M ir suskaičiuoja K1= e^(M+e), jei M1=M+e tai K1=e^M1;
3. Bobas sugeneruoja slaptą skaičių N ir suskaičiuoja K2=e^(N+e), jei N1=N+e tai K2=e^N1;
4. Alisa ir Bobas apsikeičia reikšmėmis K1 ir K2, N ir M išlieka slapti ir jais neapsikeičiama;
5. Alisa apskaičiuoja bendrą raktą pagal Key=(K2)^M1=e^(M1N1);
6. Bobas apskaičiuoja bendrą raktą pagal Key=(K1)^N1=e^(M1N1);
7. abi pusės suskaičiuoja reikšmę log e (e^(M1N1))=M1N1;
8. Alisa suskaičiuoja R1=(M1N1/M1)-e;
9. Bobas suskaičiuoja R2=(M1N1/N1)-e.

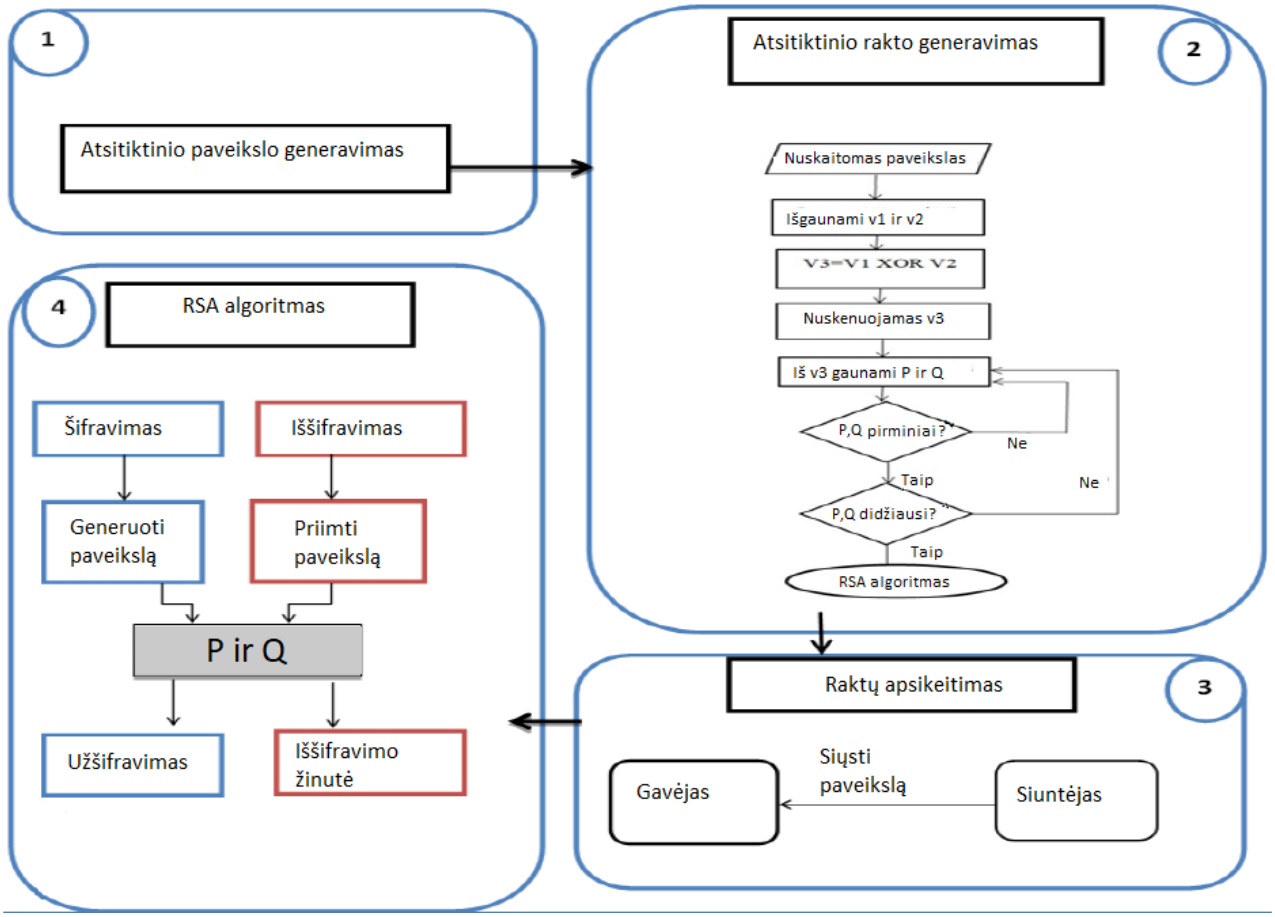
Jei R1 ir R2 yra pirminiai skaičiai – tuomet raktas yra saugus ir nebuvo atakuota. Jei gaunamas skaičius yra pirminis, tuomet daroma išvada, kad į raktų apsikeitimą buvo įsiterpta trečio asmens ir raktai sunaikinami. Raktų apsikeitimo procesas pakartojamas iš naujo.

Kitas metodas skirtas pagerinti Diffie-Hellman rakto saugumui, naudoja atsitiktinio paveikslo generavimą [3]. Sugeneruojamas numatyto dydžio atsitiktinis paveikslas, tiesiog spalvotų pikselių visuma. Iš šio paveikslo išgaunamos reikšmės rakto generavimui. Visas procesas pavaizduotas 1 paveiksle:

1. sugeneruojamas NxN dydžio, bet ne mažesnis nei 200x200 atsitiktinių pikselių paveiksliukas;
2. pereinama per visus paveikslo pikselius ir kiekvieno jų RGB vertė yra paverčiama į dvejetainį formatą;

3. paveikslas nuskenuojamas iš kairės į dešinę ir iš viršaus į apačią, taip gaunami du vektoriai V1 ir V2;
4. pritaikoma pasirinkta loginė operacija, kad iš dviejų vektorių būtų išgautas trečiasis vektorius V3;
5. iš vektoriaus V3 išgaunamos dvi didžiausios įmanomos pirminių skaičių reikšmės p ir q;
6. šios reikšmės paduodamos RSA rakto generavimo algoritmui, kuris sugeneruoja šifravimo raktą.

Šis algoritmas leidžia sugeneruoti šifravimo raktą iš paveikslo, tačiau apsikeitimas tarp dviejų pusių, vyksta beveik taip pat kaip standartiniame Diffie-Hellman protokole, tačiau apsikeičiama ne jau sugeneruotomis raktų reikšmėmis, o paveikslais iš kurių vėliau išgaunami raktai.



1 pav. Rakto generavimas panaudojant atsitiktinį paveiksliuką[3]

Rakto apsikeitimas naudojant vietos informaciją[2]

Sistema skirta naudoti pastatų viduje. Daroma prielaida, kad naudojamas belaidis tinklas turi galimybę nustatyti vartotojo buvimo vietą patalpose. Dėl žemo tikslumo, sudėtinga nustatyti tikslią vartotojo buvimo vietą, tačiau galima ganėtinai tiksliai nustatyti vartotojo atstumą iki tam tikrų stotelių.

Algoritmas yra paremtas Merklio galvosūkiais ir principu, kad atakuotojui reikėtų labai daug laiko rakto nulaužimui, kad atakuojantysis, nebesistengtų nulaužti rakto.

Sakykim A ir B nori susitarti dėl rakto. A sugeneruoja k galvosūkių ir nusiunčia juos B. Galvosūkiai yra sudaryti iš ID, atsitiktinio kriptografinio rakto ir atpažįstamo teksto, pvz.: teisingai suformuotas sakinys. Gavęs šiuos galvosūkius, B pasirenka vieną iš jų ir nulaužia(visi galvosūkiai yra šifruojami

silpnu kriptografiniu raktu). Kuomet B iššifruoja gautą užduotį, nusiunčia surastą ID atgal į A. dabar A žinos kurį raktą naudoti bendravimui su B.

Jei atakuotojas norėtų perimti raktą, šiame žingsnyje nežinotų kurią užduotį spręsti. Minimaliai turėtų išspręsti k/2, kad rastų tinkamą.

Norint labiau sustiprinti šią sistemą, galvosūkių šifravimui naudojamas sudėtingas kriptografinis raktas. Užduoties iššifravimui reikalinga žinoti bendrą paslaptį tarp A ir B.

Bendra paslaptis gaunama panaudojant atstumą tarp A ir B. Abu A ir B sugeneruoja raktų rinkinius pasinaudodami apskaičiuotu atstumu. A užšifruoja sugeneruotas užduotis gautu raktų rinkiniu ir nusiunčia galvosūkius į B. B turėdamas raktų rinkinį sugeneruotą iš panašaus apskaičiuoto atstumo, gauna kelis raktus, kurie atitinka tuos, kurie buvo sugeneruoti A ir jam pavyksta iššifruoti kelis galvosūkius. Gauti raktai iš šių galvosūkių yra apjungiami į vieną kriptografinį raktą, o jų ID nusiunčiami atgal į A. A gavęs raktų ID, iš jų taip pat sugeneruoja vieną kriptografinį raktą. Vėliau gauti raktai gali būti naudojami šifravimui arba stipresnio kriptografinio rakto sukūrimui ir apsikeitimui.

Puolantysis norėdamas perimti raktus, turėtų tiksliai atspėti atstumą tarp A ir B. Jei būtų naudojami keli B taškai, tuomet puolantysis turėtų atspėti A atstumus iki visų B taškų, nes visi atstumai būtų naudojami raktų apsikeitime.

1.2. Biometrijos panaudojimo šifravimo raktų generavimui ir valdymui galimybių analizė.

1.2.1. Biometrinių duomenų ir kriptografijos apjungimo galimybės

Kriptografija nuo seno naudojama paslėpti duomenis ar padaryti tekstą neperskaitomą kitiems asmenims. Jei informacija skirta tik vienam asmeniui, idealu būtų panaudoti jo biometrinius duomenis tam tekstui ar duomenims užšifruoti. Tokiu būdu tik pats biometrinių duomenų savininkas galėtų iššifruoti jam skirtus duomenis. Dėl šios priežasties pradėta galvoti, kaip apjungti kriptografiją su biometrija, taip gauta biometrinė kriptografija ir jos pagrindiniai tipai[15]:

- kriptografinio rakto ir biometrinių duomenų apjungimo biometrinės sistemos – šis metodas apjungia kriptografinį raktą su biometrinių duomenų šablonu, kad tik rakto savininkas galėtų juo naudotis;
- kriptografinio rakto generavimo biometrinės sistemos – šis metodas generuoja kriptografinį raktą iš vartotojo biometrinių ir pagalbinių duomenų.

Abi šios sistemos turi savų stiprybių ir silpnybių. Kriptografinio rakto ir biometrinių duomenų apjungimo biometrinėse sistemose, užpuolikas sužinojęs privatųjį raktą, gali gauti originalius biometrinius duomenis. Kriptografinio rakto generavimo biometrinėse sistemose raktas nėra saugomas pačioje sistemoje, todėl sudėtinga išgauti biometrinius duomenis iš rakto. Tačiau tokiose sistemose negalima naudoti rakto atstatymo algoritmų, o rakto apjungimo sistemos yra pažeidžiamos registrų atakomis.

Yra keturios rakto ir biometrinių duomenų apjungimo biometrinių sistemų technikos:

1. biometrinis šifravimas (angl. Biometric encryption) – naudojama standartinė kriptografija, norint sukurti saugų biometrinį šabloną;

2. klaidinančių įsipareigojimų schema (angl. fuzzy comitment scheme) – naudoja kriptografiją ir klaidų taisymo algoritmus. Generuoja duomenis iš biometrijos ir privataus rakto, raktas yra apsaugomas naudojantis maišos funkcijomis;
3. klaidinanti saugykla (angl. fuzzy vault) – naudoja netvarkingą biometrinių duomenų rinkinį siekiant paslėpti privatųjį raktą saugykloje. Saugykla nėra užkoduota tol, kol biometriniai duomenys sutampa;
4. skydo funkcija (angl. shielding function) – sugeneruoja saugų šabloną iš atsitiktinių biometrinių duomenų. Negalima gauti biometrinių duomenų, nežinant privačiojo rakto.

Taip pat yra du rakto generavimo biometrinių sistemų tipai:

1. privačių šablonų schema (angl. private template scheme);
2. kvantavimo schema (angl. quantization schemes).

Biometrinis duomenų atpažinimas turi problemą, kad vartotojas gali būti klaidingai identifikuojamas. Visos ankščiau išvardintos technikos yra vertinamos pagal kelis kriterijus:

- FAR arba FMR – rodo santykį, kokiu gautiems neteisingiems biometriniams duomenims yra randamas atitikimas duomenų saugykloje;
- FRR arba FNMR – rodo santykį, kokiu gautiems teisingiems biometriniams duomenims yra nerandamas atitikimas duomenų saugykloje.

Kelių metodų palyginimas pagal šiuos kriterijus aprašytas 2 lentelėje.

2 lentelė. Metodų palyginimas[28]

Autorius	**Technika**	**Biometrinių duomenų tipas**	**Rakto ilgis (bitais)**	**FAR/FMR (%)**	**FRR/FNMR (%)**
Jain *et al*., (2008) [18]	„Fuzzy Vault“	Piršto atspaudas	112	0.13	9
			128	0.01	9
			160	0	14
Li *et al*., (2010) [19]	„Fuzzy Vault“	Piršto atspaudas	14	0	12
Marino *et al*., (2012) [20]	„Fuzzy Extractor“	Akies rainelė	192	4.42	9.67
Eskander *et al.*,(2014) [21]	„Fuzzy Vault“	Parašas neprisijungus	192	1.41	20.68
Amirthalingam and Radhamani (2016) [22]	„Fuzzy Vault with PSO“	Multimodalinis – Veidas ir ausys	10	10	10
Adamovic *et al*., (2017) [23]	„Fuzzy Commitment“	Akies rainelė	400	0	3.75
			300	0	2.21
			200	0	1.26
Yang *et al.*, (2018) [24]	„Cancelable biometric system“(Fuzzy Commitment)	Veidas	15	0	13
			29	0	16
			36	0	35
			64	0	59
			8	0.72	11
			9	0.51	16

Chitraand Sujitha (2018) [25]	„Fuzzy Vault“	Piršto atspaudas	10	0.04	17
			11	0.01	22
			12	0.01	25
Elrefaei and Al-Mohammadi (2019) [26]	„Fuzzy Commitment“	Eisena	50-bits	0 (greitas ėjimas)	0 (greitas ėjimas)
			64-bits	0 (lėtas ėjimas)	4 (lėtas ėjimas)
			45-bits	0 (45-laipsniai)	0 (45- laipsniai)
Ponce-Hernandez *et al.,* (2020) [27]	„Fuzzy Vault“	Parašas	128-bits	MCYT duomenų bazė	
				6.91	7.85
				Nuosava duomenų bazė	
				6.21	4.86
				BioSecure duomenų bazė	
				6.16	13.6

1.2.2. Galimi biometrinių duomenų tipai

Biometrinius duomenis galima suskirstyti į tipus, pagal tai kokia asmens fizinė ar elgesio charakteristika buvo nuskaityta. Labiausiai paplitusios charakteristikos[7]:

- piršto atspaudas – egzistuoja kelių tipų skaitytuvai skirti piršto atspaudo atvaizdui padaryti, tačiau yra du pagrindiniai metodai duomenims iš paveikslų išgauti: susikirtimo taškų (angl. Minutiae) paieškos ir koreliacijos (angl. Correlation) metodai;
- veido atpažinimas – naudojantis kamera padaromas veido atvaizdas. Veido identifikavimui naudojami taip pat 2 metodai: veido sričių atpažinimo ir Eigeno veidų(angl. Eigen faces) metodai. Pirmasis metodas nustato atstumus tarp veido sričių tokių kaip akys, burna ar nosis, o antrasis lygina gautą nuotrauką su iš anksto sudarytu veido sričių paveikslų rinkiniu ir suranda didžiausią atitikimą kiekvienai sričiai;
- akies rainelė – padaroma juodai balta akies nuotrauka, kurioje surandama akies rainelė. Akies rainelės paveiksle surandami tam tikri taškai, kurie yra skirtingose vietose, kiekvieno žmogaus akyje. Sudaromas šių taškų žemėlapis kuris vėliau panaudojamas palyginimui;
- delno geometrija – nuskenuojamas delno atvaizdas. Gautame atvaizde išskaičiuojami delno, pirštų pločiai ir atstumai. Žmonių delnų forma skiriasi, tačiau šis metodas yra vienas iš netiksliausių;
- akies tinklainės geometrija – specialiu skenavimo prietaisu skenavimas užtrunka apie 10-15 sekundžių, kol skenuojama reikia akį išlaikyti stabiliai viename taške. Iš esmės yra nuskenuojamas akies kraujagyslių žemėlapis. Vienas iš saugiausių būdų, klaidos tikimybė 1 iš 10000000;
- balso atpažinimas – ištarus tam tikrą iš anksto žinomą tekstą, įrašomas žmogaus balsas. Iš įrašo galima išgauti duomenis ne tik apie fizinius balso stygų ir burnos duomenis, bet ir žmogaus manierų duomenis. Senstant žmogui jo balsas keičiasi, todėl reikia vis atnaujinti sistemos duomenis;
- parašo patvirtinimas – nuskaitomas ir lyginamas ne pats parašo atvaizdas, o kokią jėga, kokiu greičiu buvo pasirašyta. Kokiuose taškuose buvo sustota ir su kokia inercija judėjo rašiklis. Sudėtinga atkartoti, nes nepakanka atkartoti parašo atvaizdą;

– galimi ir kiti būdai, tačiau jie yra naudojami retai: delno atvaizdas, rankos kraujagyslės, DNR, terminė nuotrauka, ausies forma, kūno kvapai, odos po nagu struktūra ir kiti.

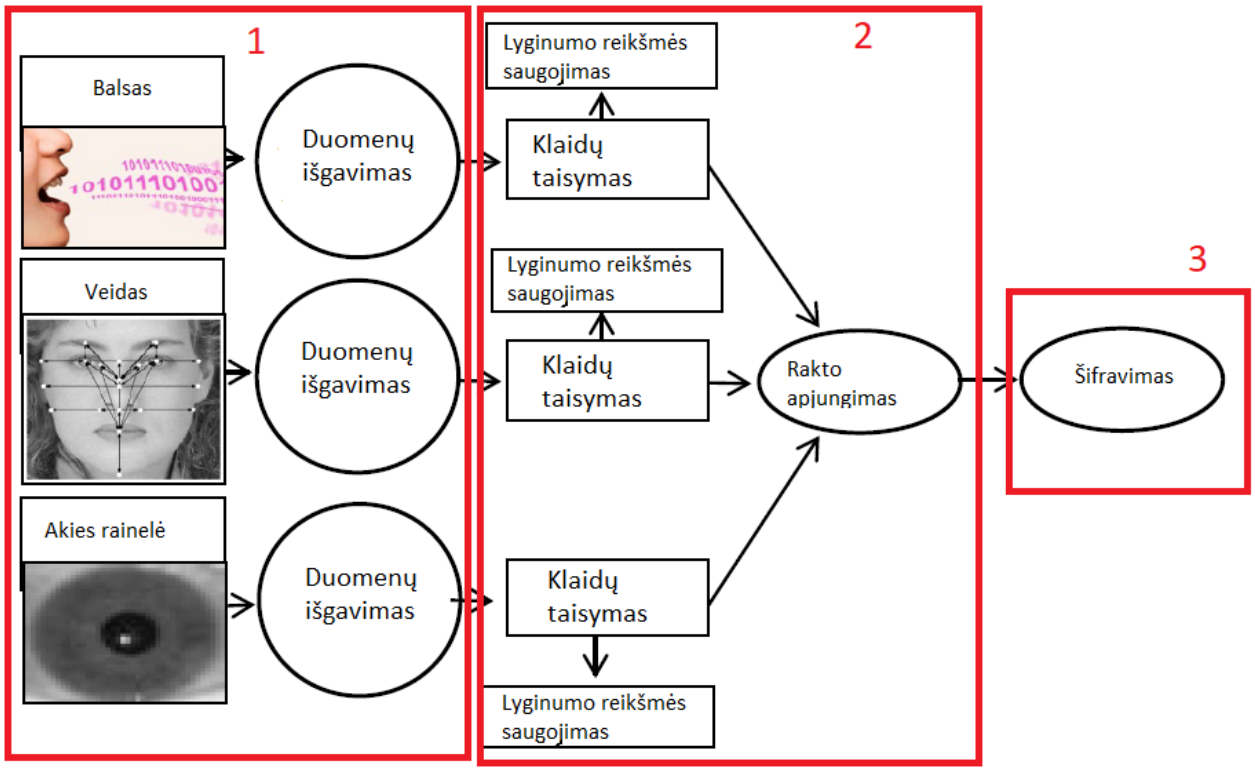
Daugelis šių biometrinių duomenų tipų gali būti naudojami atskirai arba grupuojami.

1.2.3. Kelių skirtingų biometrinių duomenų tipų apjungimas ir panaudojimas šifravimui

Naudojant kelių tipų biometrinius duomenis, galime išgauti saugesnius šifravimo raktus. Panašiai, kaip standartiniuose raktų generavimo algoritmuose didinant pradinių duomenų ilgį, išgaunami ilgesni ir saugesni šifravimo raktai, taip ir biometrijoje, naudojant kelis skirtingus biometrinių duomenų tipus – išgaunamas sunkiau atkartojamas pradinių duomenų rinkinys.

Panaudojant balso, veido ir akies rainelės duomenis, sugeneruojama saugi skaitmenų seka, kurią galima panaudoti rakto generavimui [5]. Norint apjungti skirtingų tipų biometrinius duomenis reikia pereiti 3 etapus (2 paveikslas):

1. biometrinių duomenų išgavimas ir pavertimas į dvejetainius duomenis;
2. klaidų taisymas ir duomenų apjungimas;
3. šifravimas.



2 pav. 3 tipų biometrinių duomenų apjungimas rakto generavimui[5]

Nuskaičius balso įrašą, reikia išvalyti galimus triukšmus ir sustiprinti pagrindinius balso dažnius. Panaudojus specialius filtrus išgaunamas reikiamas balso įrašas. Tuomet įrašas yra skaitmenizuojamas ir paverčiamas į dvejetainių simbolių seką.

Gavus veido nuotrauką, jos raiška sumažinama ir panaudojant diskrečiąją kosinuso transformaciją (angl. discrete cosine transform) išgaunamas juodai baltas paveikslas. Gauta pikselių matrica yra paverčiama į dvejetainių simbolių seką.

Gavus akies rainelės nuotrauką, ji apkarpoma, kad neliktų antakių ir kitų analizei nereikalingų dalių. Apdorota rainelės nuotrauka yra dar kartą apdorojama taip, kad jos raiška būtų sumažinta tokiu pat būdu kaip ir su veido nuotrauka. Tuomet gauta pikselių matrica yra paverčiama į dvejetainių simbolių seką.

Sekančiam žingsnyje reikiami duomenys išsaugojami ir vėliau bus panaudoti iššifravimui. Panaudojami maišymo ir AES šifravimo metodai, kad trijų skirtingų biometrinių tipų duomenys būtų apjungiami.

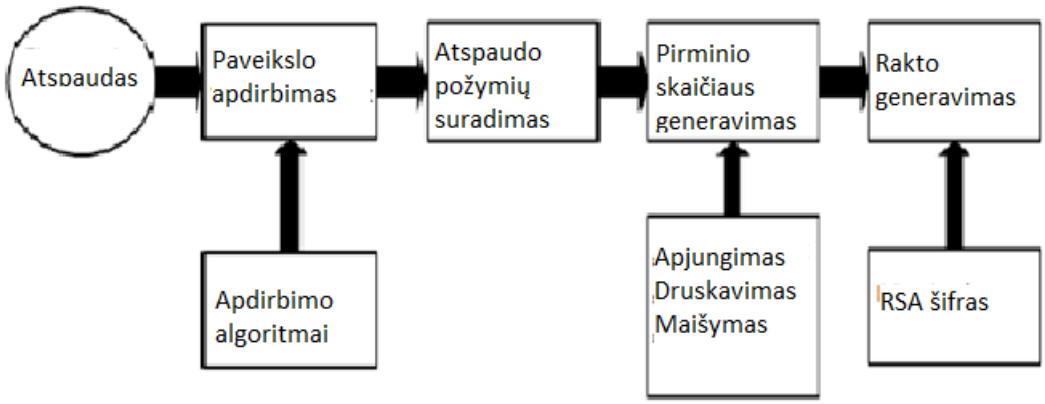
Paskutinis žingsnis, suskaido apjungtus duomenis į 3 dalis ir panaudoja jas kaip 3 64 bitų raktus šifravimui naudojant 3DES metodą.

Iššifravimo žingsnyje, vėl nuskaitomi biometriniai duomenys ir apjungiami su prieš tai išsaugotais duomenimis. Jei asmuo, kurio biometriniai duomenys buvo nuskaityti sutampa, tuomet yra gaunamas teisingas raktas duomenų iššifravimui.

1.3. Biometrinių šifravimo raktų generavimo ir valdymo metodų analizė ir įvertinimas.

1.3.1. RSA rakto generavimas iš piršto atspaudo[9]

Vietoj standartinių RSA rakto kūrimui reikalingų pirminių skaičių, tam panaudojami piršo atspaudo duomenys(procesas pavaizduotas 3 paveiksle):



3 pav. RSA rakto generavimas iš piršto atspaudo[9]

1. pakoreguojamas piršto atspaudo atvaizdas. Šis žingsnis skirtas padidinti atvaizdo kontrastą, prafiltruoti gautus triukšmus, pakoreguoti ryškumą;
2. suskaidžius atvaizdą į blokus, visi blokai paverčiami į binarinę formą, t. y. pirma turimas pilkų atspalvių vaizdas yra konvertuojamas į juodus arba baltus pikselius, be tarpinių reikšmių. Tai padaroma pasitelkus tikslius algoritmus, kurie suskaičiuoja atskaitinę reikšmę tame bloke. Tuomet kiekvienas bloko pikselis lyginamas su gauta reikšme. Jei pikselio reikšmė mažesnė, nustatoma balta spalva, jei reikšmė didesnė – juoda;
3. norint padaryti atpažinimą tikslesnį, panaudojamas atspaudo pakoregavimas. Naudojamos dvi operacijos:

   – priauginimas – pridedami papildomi pikseliai;
   – erozija – pikseliai pašalinami iš atvaizdo;

4. siaurinimas. Šio žingsnio metu, visos piršto atspaudo linijos susiaurinamos iki vieno pikselio storio;
5. bruožų išskyrimas. Iš kiekvieno 3x3 pikselių bloko paimamas vidurinis skaitmuo. Jei vidurinis skaitmuo yra 1 ir turi tris kaimyninius 1, tuomet laikoma, kad šis taškas yra išsišakojimo taškas. Jei centrinis skaitmuo yra 1 ir turi tik vieną kaimyninį 1-etą, tuomet laikoma, kad tai yra atspaudo rievės galas.;
6. duomenų grupavimas. Visi gauti duomenys yra apibūdinami kaip x ir y koordinatės. Šios koordinatės yra sudedamos į 2 vektorius:

   – $F_1 = [x_1, x_2, \dots, x_n]$; $|F_1| = n$;
   – $F_2 = [y_1, y_2, \dots, y_n]$; $|F_2| = n$;

7. pirminių skaičių generavimas. Gauti x ir y koordinačių vektoriai yra sukoncentruojami į vieną vektorių, taip kad pirma eitų x taškai, o tada visi y taškai:
   $S = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]$; $|S| = 2n$
   Tuomet vektorius S yra paverčiamas į 1024 bitų vektorių. Tai padaroma pridedant papildomų atsitiktinių arba pašalinant perteklinius aukščiausius bitus.

   Sekančiu žingsniu panaudojamas “Mongean” duomenų maišymo algoritmas. Šio algoritmo principas paremtas dviem elementų krūvomis. Elementas imamas iš pirmos krūvos ir dedamas į antros pradžią, imamas sekantis elementas ir dedamas į antros krūvos pabaigą ir taip po vieną elementą tai į pradžią tai į pabaigą, kol visi elementai atsiduria antroje krūvoje.

   Paskutinis žingsnis yra gautų skaičių koncentravimas. Tikslas yra gauti lyginį pirminį skaičių. Jei turimas skaičius yra lyginis bet ne pirminis, bus pridedamas 2, kol skaičius pasidarys pirminis. Jei skaičius yra nelyginis ir ne pirminis, pirmu žingsniu bus pridėtas 1, jei skaičius vis dar netinkamas, toliau bus pridedamas 2;
8. gautas 1024 bitų pirminis skaičius panaudojamas RSA rakto generavimui, kaip ir įprastai.

1.3.2. Kriptografinio rakto generavimas panaudojant smegenų signalus[14]

Milijardai neuronų esančių žmogaus smegenyse, siuntinėdami signalus sukuria elektros signalus. Šie signalai sukuria didžiulį elektrinį aktyvumą smegenyse, kuris gali būti apibūdinamas kaip bangos. Šios bangos gali būti aptinkamos panaudojant medicininius įrenginius. Nuskaičius smegenų generuojamus signalus ir išnagrinėjus juos pasitelkiant dirbtinį intelektą, galima atskirti žmogaus būseną, t .y. jausmus, kaip žmogus jaučiasi, apie ką galvoja, nes yra nustatyta, kad skirtingos būsenos žmogaus smegenys generuoja skirtingus signalus, skirtingose smegenų vietose.

Duomenų šifravimas smegenų signalais atliekamas taip:

1. žmogui duodama užduotis, kurią jam atliekant yra analizuojamas smegenų aktyvumas;
2. išanalizavus smegenų aktyvumą, atskiriamos pagrindinės bangos ir paverčiamos į skaitinę reikšmę panaudojant signalo -> binarinį keitiklį, panašiai, kaip analoginio signalo į skaitmeninį keitimas;
3. iš gauto signalo sudaromas šifravimo raktas.

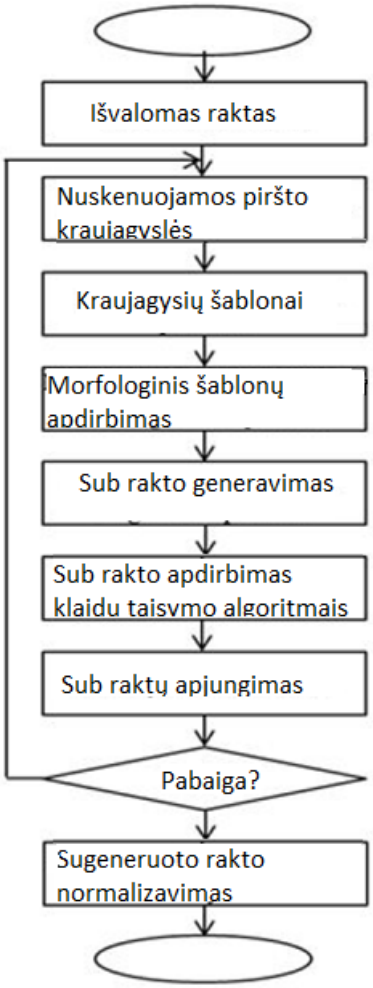
Gautu raktu užšifravus duomenis, juos iššifruoti galės tik tas pats asmuo atlikdamas tas pačias užduotis.

1.3.3. Rakto generavimas iš piršte esančių kraujagyslių[13]

Šis metodas leidžia generuoti kriptografinius raktus pasitelkiant piršto kraujagyslių „žemėlapio“ nuotrauką. Tai yra pseudo – kelių dalių (Pseudo-Multimodal) metodas. Biometriniai duomenys yra kombinuojami su slaptažodžiu, kuris yra sudaromas paduodant skirtingą piršto kraujagyslių nuotraukų seką, t. y. galima nuskenuoti iki 10 to paties asmens pirštų ir padavus juos skirtinga seka, galima sudaryti labai daug skirtingų slaptažodžių.

Rakto generavimui vykdomi tokie žingsniai (pavaizduoti 4 paveiksle):

1. padaromas piršto kraujagyslių paveikslas;
2. paveikslas apdorojamas Miura “Repeated Line Tracking”, Miura “Maximum Curvature”, Huang “Wide Line Tracking” kelio sekimo algoritmais ir papildomomis matematinėmis morfologinėmis funkcijomis;
3. gautas kraujagyslių žemėlapis apdorojamas ribų paieškos algoritmu, kuris suranda žemėlapio „kraštus“, taip gaunamas tarpinis kriptografinis raktas;
4. klaidų taisymo algoritmas panaudojamas pašalinti galimus netikslumus;
5. pakartojus procesą visiems turimiems kraujagyslių paveikslams, gauti tarpiniai raktai sukombinuojami į bendrą galutinį kriptografinį raktą;
6. gaunamas raktas yra kintamo ilgio, todėl panaudojamas raktų suderinimo algoritmas, kad rakto ilgis būtų pakeičiamas į reikiamą ilgį.



4 pav. Rakto generavimas iš piršto kraujagyslių[13]

Paveikslo apdorojimo metu, algoritmai ieško kraujagyslių pradžios, pabaigos ir susikirtimo taškų. Radus šiuos taškus, įsimenamos jų koordinatės paveiksle.

Visi pradžios ir pabaigos taškai yra sunumeruojami. Pasirinkus vieną iš pradžios taškų, kelių paieškos algoritmai ieško kelių iki kitų taškų. Radus visus pabaigos ir pradžios taškus, sudaryta rastų galūnių seka sukombinuojama į vieną tarpinį raktą, tokia tvarka, kokia galūnės buvo rastos.

1.3.4. Kriptografinio rakto generavimas iš piršto atspaudo

Šiame metode [17] biometriniai piršto atspaudo duomenys yra užfiksuojami naudojant blokinį metodą ir iš gautų blokų sudaromas funkcinis vektorius. Šis vektorius vėliau panaudojamas sukurti šifravimo raktui ir raktiniam žodžiui. Gautu kriptografiniu raktu duomenys yra šifruojami. Naudojant piršto atspaudo linijų, jungiančių taškus, ilgius ir pasvirimo kampus, bei Reed-Solomon kodavimą – išgaunamas raktinis žodis. Šis gautas žodis yra apjungiamas su užšifruota šifrograma ir taip gaunama galutinė šifrograma.

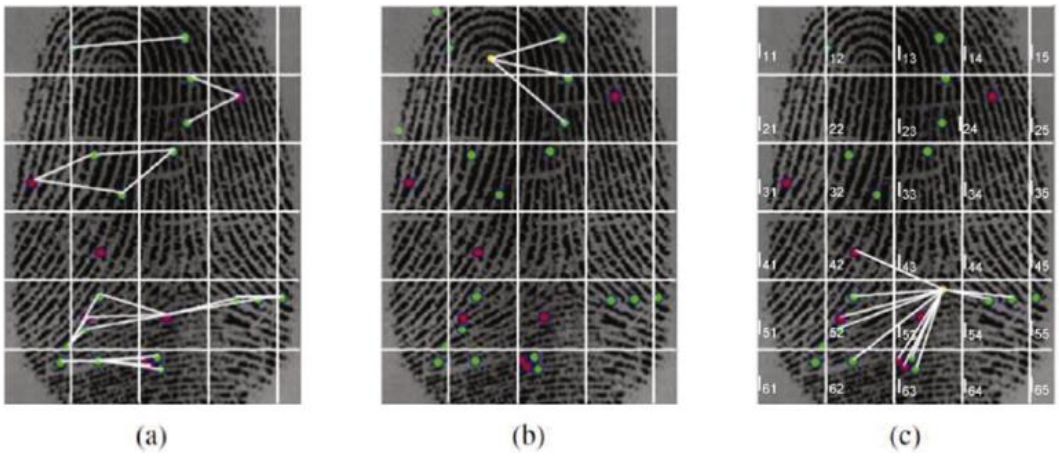
Iššifravimo procese vėl naudojama naujai padaryta piršto atspaudo nuotrauka iš kurios išgaunamas funkcinis vektorius. Gautas vektorius palyginamas su ankščiau gautu raktiniu žodžiu. Jei vartotojas atitinka – raktas sugeneruojamas iš iššifruoto raktinio žodžio. Procesas pavaizduotas 6 paveiksle.

Užšifravimas:

1. piršto atspaudo atvaizde surandami visi linijų susikirtimo (angl. minutiae) taškai, vėliau randami pagrindiniai (angl. core) ir delta taškai. Linujų susikirtimo taškų koordinatės sudedamos į rinkinį $P = \{p_1 (x_1 , y_1) , p_2 (x_2 , y_2), \cdots, p_k (x_k , y_k)\}$, pagrindinis taškas pažymimas $C_p(x_c , y_c)$ ir delta taškas žymimas $D_p(x_d , y_d)$;
2. norint suskaičiuoti atstumus ir kampus tarp p taškų, paveikslas yra suskaidomas į mažesnius blokus. Jei paveikslas yra I, jis suskaidomas į mažesnius blokus, kurių dydis m x m pikselių. m << M, N, kur M x N yra originalus paveikslo dydis. Iš esmės I galima aprašyti kaip p x q blokų matricą:

$$I = \begin{pmatrix} I_{11} & \cdots & I_{1q} \\ \vdots & \ddots & \vdots \\ I_{p1} & \cdots & I_{pq} \end{pmatrix}$$

   – linijų duomenų skaičiavimas naudojantis visais blokais – po paveikslo padalijimo į blokus, apskaičiuojamos visos linijos nuo parinkto pradinio bloko I_{ij} į visus gretimuose blokuose esančius linijų susikirtimo taškus p. Taip pat linijos skaičiuojamos ir į taškus esančius tame pačiame pradiniame bloke.
   Gavus visas linijas – skaičiuojamas jų ilgis ir posvyrio kampas. Jei p_b ir p_c yra du linijų susikirtimo taškai, tai naudojantis Euklido atstumu suskaičiuojamas linijos ilgis ir randamas posvyrio kampas nuo x ašies. Surandamos linijos tarp visų nesikartojančių taškų porų esančių tinkamuose blokuose. Linijų ilgiai l ir kampai a sudedami į rinkinį $F_B = \{(l_1 , a_1) , (l_2 , a_2) , \cdots, (l_{z^b} , a_{z^b})\}$, kur F_b ilgis yra z^b;
   – linijų duomenų skaičiavimas naudojantis delta ir pagrindiniu taškais – randame bloką I_{lm} kuriame yra pagrindinis taškas C_p. Tuomet randamos visos linijos esančios aplinkiniuose 8 blokuose I_{rs}. Tuomet randami visų linijų ilgiai ir kampai. Jei $C_p (x_c , y_c)$ ir $p'_c (x'_c , y'_c)$, tai gaunamas linijų ilgių ir kampų sąrašas $F_C = \{ (l'_1 , a'_1) , (l'_2 , a'_2) , \cdots, (l'_{z^c} , a'_{z^c}) \}$, kurio dydis z^c. Panašiai skaičiuojami ir delta taškai. Gaunamas linijų ilgių ir kampų sąrašas $F_D = \{ (l''_1 , a''_1) , (l''_2 , a''_2) , \cdots, (l''_{z^d}, a''_{z^d}) \}$, kurio dydis z^D.
   Tada visi rinkiniai yra apjungiami į vieną bendrą rinkinį $F= \{ F_B \| F_C \| F_D \}$, kurio ilgis yra $z = z^b + z^c + z^d$;



5 pav. (a) linijos pagal blokus, (b) linijos pagal pagrindinį tašką, (c) linijos pagal delta tašką[17]

3. linijų duomenų paslėpimas – kad būtų sunkiau atkurti duomenis, visų rinkinių linijų ilgiai ir kampai yra paverčiami į dvejetainę formą ir sujungiami naudojant XOR operaciją. Taip gaunami

rinkiniai sudaryti iš linijų, kurių duomenys jau yra apjungti. Apjungus visus rinkinius, gaunama ilga dvejetainė skaičių seka, kuri panaudojama gauti bio-kriptografiniam raktui;

4. rakto generavimas:
   - po rinkinių apjungimo, duomenų seka paduodama į duomenų pakeitimo modulį, kuris pagamina bio-kriptografinį raktą. Tikimasi gauti 1024 bitų ilgio raktą;
   - jei rakto ilgis mažesnis nei 1024 bitų – panaudojami du rinkiniai A_1 ir A_2, kurie sudaryti iš atsitiktinių duomenų. Jų duomenimis papildomas gautas raktas, kol jo ilgis pasiekia 1024 bitų;
5. kodinio žodžio generavimas – naudojantis Reed-Solomon klaidų tvarkymo kodus, sugeneruojamas $2^f - 1 - 2^t$ ilgio raktinis žodis, kur f yra bet koks teigiamas sveikas skaičius, o t reikiamų taisyti klaidų kiekis;
6. šifravimas - užšifravimui gali būti naudojamas bet koks šifravimo algoritmas turintis CBC rėžimą. Užšifravus su 4 žingsnyje gautu raktu, šifrograma dar apjungiama su kodiniu žodžiu – taip gaunama galutinė šifrograma.

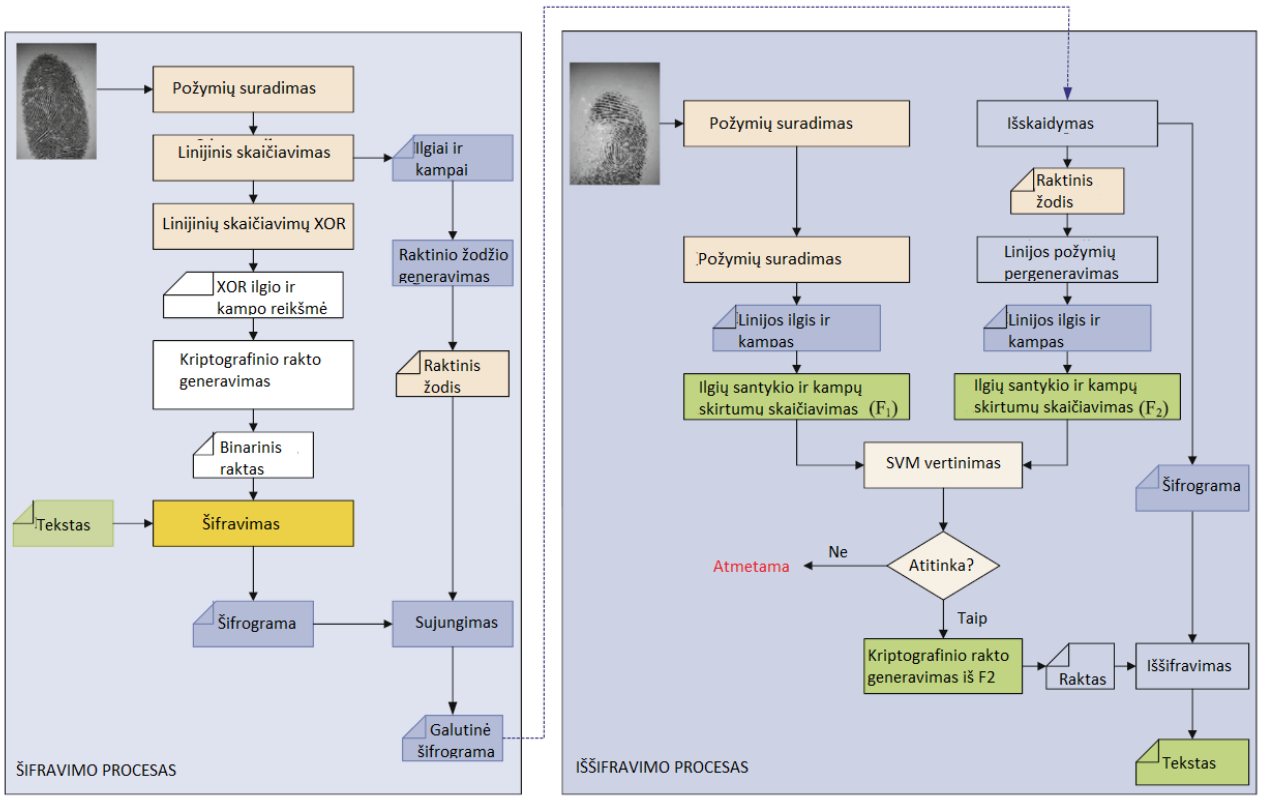
Iššifravimas:

1. iš naujai paimto piršto atspaudo išgaunamos visos linijos, kaip ir šifravimo etape. Gaunamas rinkinys $F_1 = \{ FV'_B , FV'_C , FV'_D \}$. Siekiant patvirtinti, kad tai yra tas pats vartotojas, turi būti palyginami linijų ilgių santykiai ir kampų skirtumai;
2. iš kodinio žodžio saugomų duomenų atgavimas – šifrograma padalinama į dvi dalis: kodinį žodį ir šifrogramą. Norint išgauti kodinį žodį, paimami paskutiniai $4 \times m \times z_{max}$ simboliai, kur z_{max} yra išgautas šifravimo proceso metu ir paverčiama į dešimtainę formą. Gaunama: $C_{dec} = \{ d_1 , d_2 , d_3 , \cdots, d_{z_{max}} \times 2 \}$. Likusi priekinė dalis yra originali šifrograma.

   Šiame algoritme, kodinis žodis yra linijų rinkinio F reprezentacija. Norint atgauti rinkinį F, kodinio žodžio elementai yra atstatomi į jų originalias išraiškas. Tam papildomai panaudojama klaidų paieškos lentelė (angl. Error Lookup Table) ir Reed-Solomon dekoderis.

   Apskaičiuojant ilgių santykius $Lr_{ij} = \frac{d_i}{d_j}$ ir kampų skirtumus $Ad_{ij} = d_{i+1} - d_{j+1}$ gaunamas rinkinys FV''_B. Panašiai gaunami ir rinkiniai FV''_C ir FV''_D. Iš jų sudaromas rinkinys $F_2 = \{ FV''_B , FV''_C , FV''_D \}$;
3. linijų rinkinių palyginimas – sakykime, kad $(lr_i , ad_i) \in FV'_B$ ir $(lr_j , ad_j) \in FV''_B$ yra bet kokios dvi poros, sakoma kad pora sutampa, jei $lr_i = lr_j$ ir $ad_i = ad_j$. Jei yra S tokių sutapimų, tada suskaičiuojama S_1 iš FV'_B ir FV''_B, taip pat ir S_2 ir S_3 atitinkamai iš FV'_C, FV''_C ir FV'_D ir FV''_D. Taip gaunamas vartotojo u įvertinimo vektorius $SV_u = \langle S_1 S_2 S_3 \rangle$

   Apskaičiuotas vektorius paduodamas vertinimo funkcijai F(.), jei jos gražinamas rezultatas yra daugiau už 0, tuomet vartotojas patvirtinamas, kitu atveju atmetamas;
4. naudojantis rinkiniais F''_B , F''_C ir F''_D sugeneruojamas bio-kriptografinis raktas, šifrogramos iššifravimui. Rakto generavimas toks pats kaip šifravimo etape;
5. iššifruojama tokiu pat algoritmu su CBC rėžimu kaip ir šifravimo etape.



6 pav. Šifravimo procesas. Pritaikyta pagal [17] šaltinį

1.3.5. Saugaus biometrinių duomenų šablono sukūrimas ir saugojimas

Vartotojo autentifikacijai ar duomenų iššifravimui gali būti naudojami biometrinių duomenų šablonai, kurie buvo sukurti vartotojo registracijos ar duomenų užšifravimo metu. Šiuos šablonus taip pat reikia tinkamai apsaugoti, kad sistemą atakuojantis asmuo negalėtų pasinaudoti šiais duomenimis.

Norint saugiai išsaugoti akies rainelės paveikslą atliekami tokie veiksmai:

1. užkraunamas paveiksliukas;
2. paveiksliukas normalizuojamas ir iš jo sukuriamas rainelės duomenų šablonas;
3. šablonas perskiriamas į dvi dalis, A ir B;
4. sugeneruojamas „Twofish“ šifravimo raktas;
5. a dalis yra užšifruojama naudojantis „Twofish“ algoritmu ir prieš tai sugeneruotu raktu;
6. sugeneruojamas „3DES“ šifravimo raktas;
7. b dalis yra užšifruojama naudojantis „3DES“ algoritmu ir prieš tai sugeneruotu raktu;
8. parenkamas paveikslėlis duomenų talpinimui ir paverčiamas į dvejetainį formatą;
9. kiekvieno pikselio pirmas žemiausias bitas yra pakeičiamas užšifruoto šablono bitu;
10. taip gautas paveikslas, kurio pikselių dvejetainio formato kodo, žemiausi bitai yra užšifruoto šablono duomenys;
11. paveikslas išsaugomas ir jame negali įžvelgti jokių akies rainelės komponentų.

Naudojantis šiuo metodu, akies rainelės duomenys gali būti patalpinami į bet kokį paveikslą, praktiškai nepakeičiant jo išvaizdos, taip nesukeliant įtarimo, kad paveiksle yra užšifruoti duomenys.

1.4. Bio kriptografijos raktų generavimo metodų analizės išvada

Analizės metu buvo aprašyti galimi biometriniai duomenų tipai, biometrinių duomenų apjungimo galimybės ir palyginti keli raktų generavimo metodai. Vieni metodai siekia sukurti saugų raktą, kiti apsaugoti raktų apsikeitimo procesą. Pastebėta, kad visi algoritmai yra aprašyti moksliniame lygyje ir nėra pakankamai patikimi komerciniam naudojimui. Sukombinavus kelis aprašytus metodus, būtų galima sukurti pakankamai saugų raktų generavimo algoritmą.

Sekančiuose skyriuose, bus siekiama aprašyti ir pateikti biometrinės kriptografijos metodą, kuris skirtas duomenims užšifruoti ir iššifruoti naudojant to paties asmens biometrinius duomenis. Atliekant eksperimentus bus siekiama patikrinti ar pateikiamas metodas tikrai išnaudoja visas galimybes, kurios leidžia sugeneruoti saugius raktus iš biometrinių duomenų. Taigi tolimesni uždaviniai yra:

1. aprašyti biometrinį šifravimo raktų generavimo metodą;
2. įgyvendinti aprašytą biometrinį šifravimo raktų generavimo metodo prototipą;
3. ištirti sukurtą biometrinį šifravimo raktų generavimo metodo prototipą ir įvertinti jo patikimumą.

2. Bio kriptografijos raktų generavimo metodo realizacija

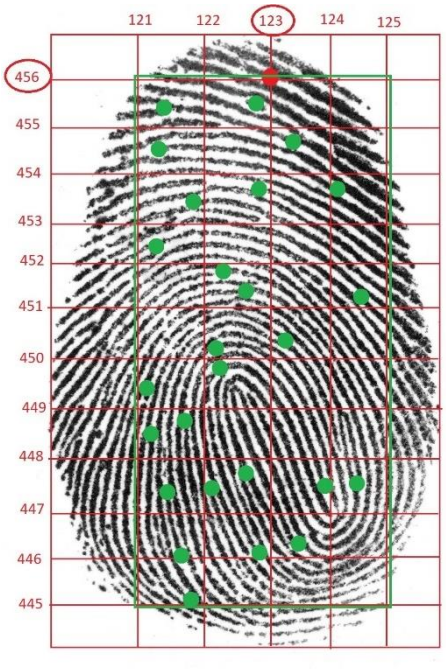
2.1. Idėja

Bus naudojami piršto atspaudo atvaizdai, dėl jų įvairovės, neinvazinio ir greito duomenų paėmimo būdo. Algoritmas išnaudoja du iš trijų saugumo trejybės faktorių:

1. tai kas aš esu;
2. tai ką žinau;
3. tai ką turiu.

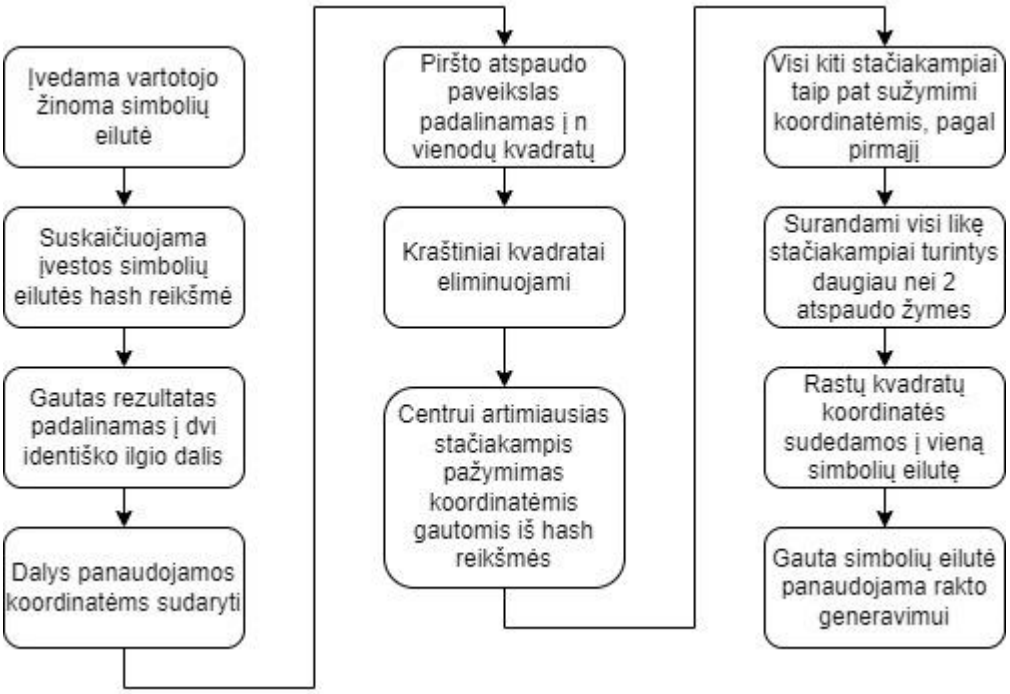
Išnaudojami bus pirmieji du faktoriai. Tai kas aš esu – asmens piršto atspaudo vaizdas. Tai ką žinau – asmens žinoma simbolių eilutė. Algoritmas veiktų tokiais etapais(7 ir 8 paveikslai):

1. paimamas piršto atspaudo vaizdas ir paruošiamas darbui;
2. asmuo įveda jam žinomą simbolių eilutę(skaičių, žodį ar frazę);
3. įvestai frazei suskaičiuojama maišos funkcijos reikšmė pagal SHA-256;
4. piršto atspaudo paveikslas sudalinamas į daug vienodo dydžio stačiakampių;
5. surandamas centrui artimiausias stačiakampis, kuriame yra piršto atspaudo žymė(linijų susikirtimas, pabaiga);
6. stačiakampis pažymimas koordinatėmis, kurios gaunamos padalinus maišos reikšmę į dvi vienodas dalis;



7 pav. Atspaudo koordinačių sistema

7. likę stačiakampiai taip pat gauna koordinates atitinkamai atsižvelgiant į pirmojo kvadrato koordinates;
8. surandami visi likę kvadratai, kuriuose yra kokių nors atspaudo žymių;
9. visų kvadratų kuriuose yra daugiau nei 2 požymiai koordinatės surašomos į simbolių eilutę;
10. gauta simbolių eilutė panaudojama kriptografinio rakto generavimui.



8 pav. Algoritmo veiksmų eiga

2.2. Piršto atspaudo paruošimas

Pirmiausia reikia paruošti piršto atspaudo paveikslą darbui. Dažniausiai tik padarius atspaudo nuotrauką, joje gali atsirasti papildomų trukdžių. Tai gali būti dulkės ant skaitytuvo, arba neryškus atspaudo atvaizdas. Norint sutvarkyti atspaudo atvaizdą, kad jį būtų galima naudoti, reikia atlikti kelis atvaizdo pakoregavimo veiksmus.

Pirmas žingsnis yra padidinti paveikslo kontrastą, kad atsirastų didesnis spalvų skirtumas tarp balto fono ir ryškesnių atspaudo linijų. Tokiu būdu sutvarkome neryškaus atvaizdo problemą. Jei kontrasto pakoregavimas nepadeda, tuomet reikėtų atvaizdą daryti per naują.

Tuomet atspaudas turėtų būti pasukamas taip, kad kelis kartus paėmus atspaudo nuotrauką ir ją pasukus, gautume vienodą rezultatą. Norint gauti vienodus duomenis iš atspaudo, atspaudas turi atrodyti kiek įmanoma vienodai, todėl kelių atvaizdų kampas taip pat turi sutapti.

Toliau pritaikomi algoritmai aprašyti tolimesniuose poskyriuose.

2.2.1. Gaussian Blur ir Gabor filtras

Gauso suliejimas (angl. Gaussian blur) leidžia sulieti paveikslą, sumažinant triukšmų kiekį jame. Suliejimas atliekamas pritaikant Gauso žemo dažnio filtrą. Filtras gali būti išreiškiamas formule:

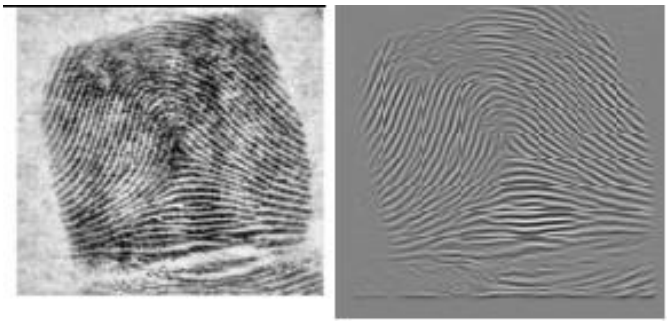
$$G_{2D}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}; \quad (1)$$

čia – x, y yra pikselio koordinatės;

σ – reikšmių pasiskirstymo standartinis nuokrypis.

Gabor filtras yra linijinis filtras. Leidžia prafiltruoti objektus pakreiptus tam tikru kampu, kaip pvz.: vertikalios arba horizontalios linijos. Šis filtras pritaikytas paveikslui suskirstytam mažomis sekcijomis – leidžia pašalinti pašalinius objektus gadinančius piršto atspaudo vaizdą.

Pritaikius šiuos filtrus visam paveikslui po kelis kartus, gaunamą rezultatą galima matyti 9 paveiksle.

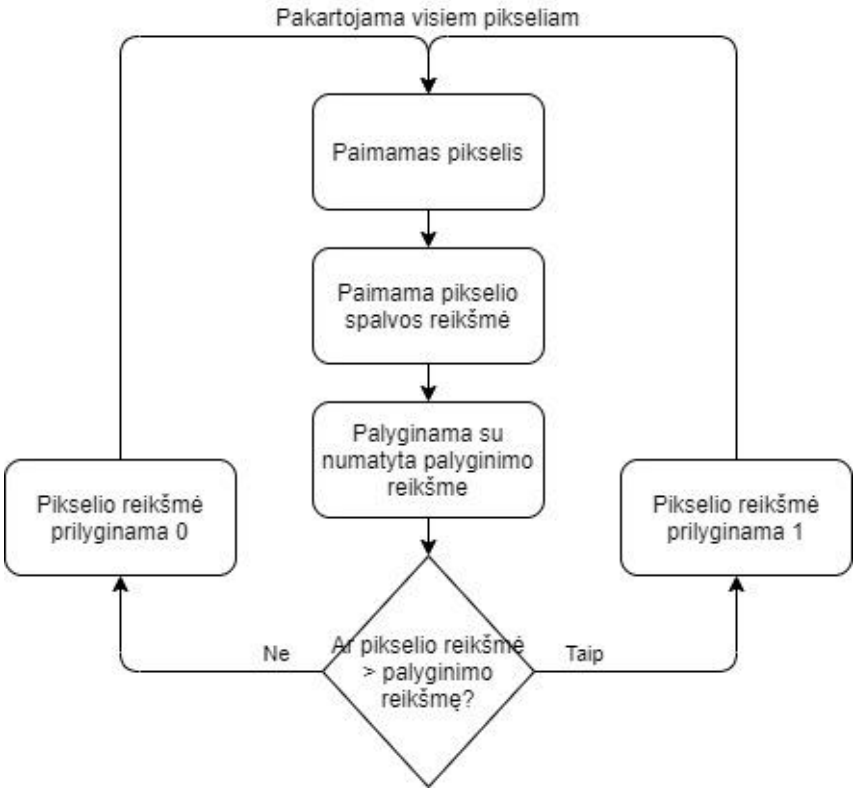


9 pav. Originalus paveikslas kairėje, pritaikius filtrus – dešinėje. Pagal šaltinį[29].

2.2.2. Binarizavimas

Binarizavimo metu paveikslas praranda visas spalvų reikšmes ir vietoj jų įgyja binarinę spalvos išraišką. 1 arba 0 reikšmės priskiriamos pagal tai ar tenkinama sąlyga pikselio spalva > z, kur z yra palyginimo svertinė reikšmė. Jei sąlyga yra tenkinama, pikseliui priskiriama reikšmė yra 1, jei sąlyga netenkinama – tuomet 0.

Pakeitus visų pikselių spalvas gauname paveikslą, kuriame yra tik juoda ir balta spalvos, atitinkamai ten kur pikselių reikšmės lygios 1 arba 0. Algoritmo veikimo žingsniai matomi 10 paveiksle. Gavus paveikslą, kuriame matomos tik juoda ir balta spalvos, galime lengvai atskirti kur yra fonas, o kur atspaudo linijos.



10 pav. Binarizavimo procesas

2.2.3. Ploninimas

Piršto atspaudo ploninimas – tai linijų susiaurinimas iki 1 pikselio pločio linijų. Tam bus naudojamas Zhang-Suen algoritmas. Šis žingsnis reikalingas tam, kad būtų galima lengvai surasti piršto atspaudo linijų išsišakojimus, susikirtimus, pradžias ar pabaigas.

Algoritmas veikia tikrindamas ar esamas 3x3 pikselių kvadratas tenkina tam tikras sąlygas. Šios sąlygos leidžia nuspręsti ar tikrinamo kvadrato centrinis pikselis turi likti juodas ar reikia pakeisti jo spalvą į baltą. Tikrinamas kvadratas sunumeruojamas atitinkama tvarka parodyta 3 lentelėje.

3 lentelė. Pikselių numeravimas

P9	**P2**	**P3**
P8	P1	P4
P7	P6	P5

Sunumeravus pikselius suskaičiuojamos dvi sumos:

- A(P1) = perėjimų nuo 0 į 1 skaičius reikšmių eilėje: P2, P3, P4, P5, P6, P7, P8, P9, P2
- B(P1) = juodų (reikšmė 1) pikselių, kurie yra kaimyniniai suma = P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9 (sudedami tik juodi pikseliai)

Tuomet tikrinama ar esamas 3x3 kvadratas atitinka tokias sąlygas, suskirstytas 5 du žingsnius:

1. pirmas žingsnis:

   - 2 <= B(P1) <= 6

- A(P1) = 1
- P2 * P4 * P6 = 0
- P4 * P6 * P8 = 0

2. antras žingsnis:

   - 2 <= B(P1) <= 6
   - A(P1) = 1
   - P2 * P4 * P8 = 0
   - P2 * P6 * P8 = 0

Jei visos sąlygos yra tenkinamos, P1 pikselis yra nustatomas kaip baltas. Jei yra netenkinamų sąlygų, pikselis paliekamas juodas. Abu žingsniai yra tarpusavyje nesusieti, tai reiškia, kad abiejų žingsnių sąlygos tikrinamos atskirai ir vieno žingsnio rezultatas, neįtakoja kito žingsnio rezultato.

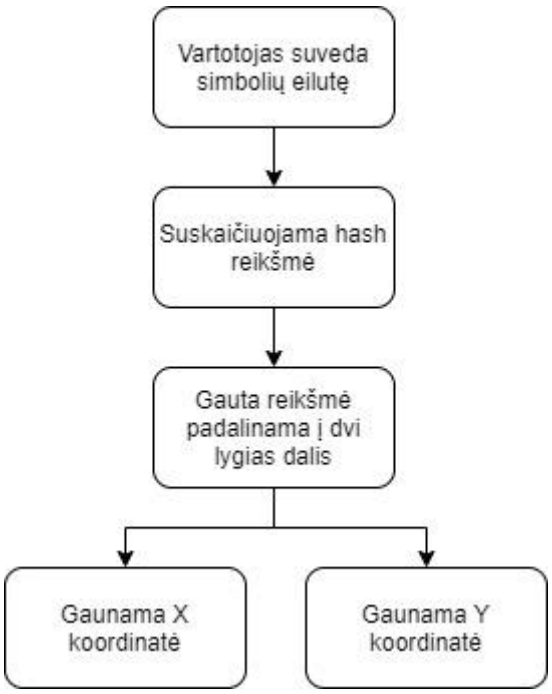
2.3. Koordinačių taško paruošimas

Pirmojo atskaitos taško koordinatės, bus sudaromos iš vartotojo įvestos simbolių eilutės. Svarbu, kad vartotojas įsimintų ką suvedė, nes tik įvedus identišką simbolių eilutę, bus galima išgauti raktą duomenų iššifravimui.

Vartotojui įvedus simbolių eilutę, bus skaičiuojama jos maišos reikšmė. Maišos funkcijos panaudojimas, leidžia iš bet kokio ilgio ar sudėtingumo simbolių eilutės išgauti vienodo ilgio reikšmę. Maišos funkcijos rezultato ilgis nekinta nei pasikeitus įvesties ilgiui, nei pasikeitus įvesties reikšmei.

Turint maišos rezultatą, jis bus padalintas į dvi lygias dalis. Tos dvi dalys sudarys taško koordinates. Pirmoji dalis bus traktuojama kaip X koordinatė, o antroji dalis – kaip Y koordinatė. Taip gausime unikalią koordinačių sistemą, kurios koordinačių atspėti beveik neįmanoma.

Koordinačių sudarymo procesas taip pat pavaizduotas 11 paveiksle.



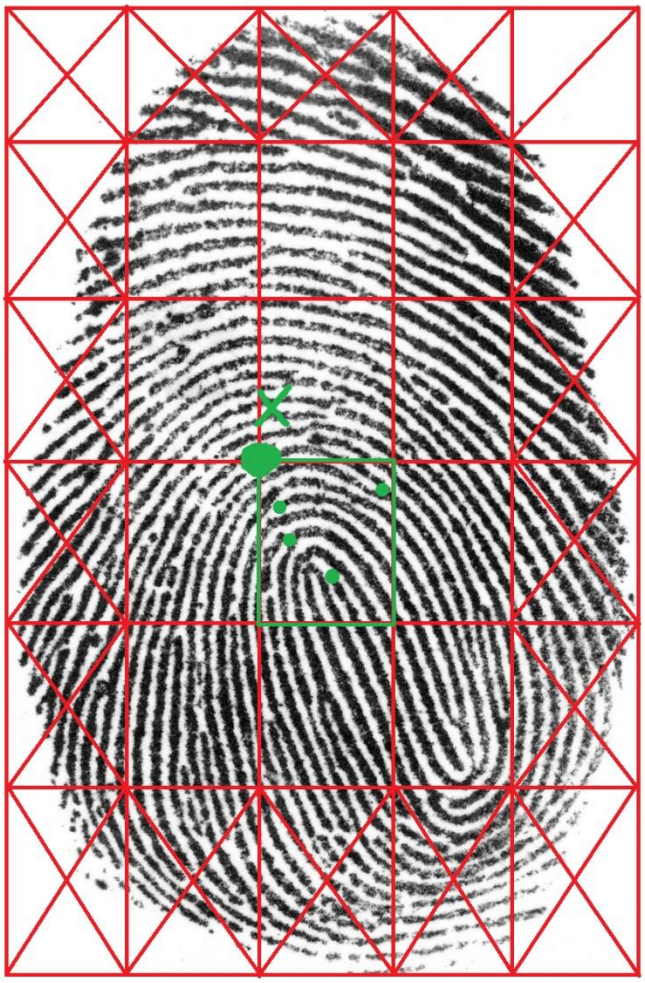
11 pav. Koordinačių sudarymas

2.4. Atspaudo reikšmės apskaičiavimas

Paruošus piršto atspaudo paveikslą ir vartotojo įvestą reikšmę, reikia sudaryti koordinačių tinklelį, kaip pavaizduota 12 paveiksle.

Suradus atspaudo kraštus, atspaudas apibraukiamas stačiakampiu. X ašis padalinama į 5 lygias dalis, o Y ašis padalinama į 6 lygias dalis. Kraštiniai gaunami stačiakampiai išmetami, nes atspaudo kraštai visada bus gauti skirtingi. Toliau:

1. gaunama 5x6 stačiakampių matrica;
2. vidurinis trečioje eilėje nuo viršaus esantis stačiakampis pažymimas, kaip atskaitinis;
3. atskaitinio stačiakampio kampo koordinatės nustatomos, pagal vartotojo įvestą reikšmę;
4. sudedamos koordinatės likusiems stačiakampiams;
5. surandami visi piršto atspaudo požymiai;
6. suskaičiuojama, kiek kuriame stačiakampyje yra požymių;
7. suskaičiuojamos visų eilučių ir stulpelių reikšmių sandaugos ir surašomos į vieną vektorių eilės tvarka;



12 pav. Tinklelio paruošimas

8. jei iššifruojama – tinklelis stumdomas po vieną pikselį, lyginant gautą sandaugų vektorių su originaliu. Jei gretimai esančių skaičių poroje vienas padidėjo, o kitas pamažėjo lyginat su

pradinėmis reikšmėmis – vadinasi, tinklelį reikia paslinkti sumažėjusio skaičiaus kryptimi. Kiekvieno žingsnio metu, suskaičiuojama vidutinė visų langelių sandaugos reikšmė;
9. jei iššifruojama – tinklelis koreguojamas pagal 8 žingsnį tol, kol sandaugų vektoriai sutampa, arba kol sutampa vidutinė sandaugos reikšmė;
10. iš stačiakampių, kuriuose yra daugiau nei du atspaudo požymiai, koordinačių – sudaroma simbolių eilutė, surašant stačiakampių koordinates eilės tvarka;
11. apskaičiuojama gautos eilutės maišos reikšmė.

2.5. Šifravimas

Iš piršto atspaudo gauta maišos rezultato reikšmė yra 256 bitų ilgio, todėl ją galima naudoti kaip šifravimo raktą AES-256 algoritmui. Algoritmas yra simetrinis, todėl ir šifravimui ir iššifravimui reikalingas toks pats raktas. Teisingai apdorojus piršto atspaudą, visuomet bus gaunamas tas pats raktas, todėl simetrinis algoritmas puikiai tinka.

2.6. Išvada

Atlikus analizę ir apgalvojus galimą algoritmo eigą, pastebėta, kad vienas iš sunkiausių uždavinių bus tokio pat pradinio piršto atspaudo vaizdo gavimas. Tikimasi, kad pozicionavimas aprašytas 2.4 skyriuje, padės išspręsti šią problemą.

3. Bio kriptografijos raktų generavimo metodo realizacija ir eksperimentas

3.1. Eksperimento aplinka

Prototipo kūrimui ir eksperimento atlikimui pasirinkta „Python“ programavimo kalba. Dėl savo paprastumo ir gero suderinamumo su „OpenCV“ programine įranga, ši kalba puikiai tinka darbui su vaizdais – šiuo atveju piršto atspaudų paveikslais.

3.2. Prototipo realizacija

3.2.1. Atspaudo paveikslo paruošimas

Prieš pradedant rakto generavimą, reikia turėti paruoštą piršto atspaudo paveikslą.

Paveikslo paruošimui reikalingi žingsniai buvo aprašyti 2 skyriuje.

Pirmiausia bus panaudojama „fingerprint-enchancer“ biblioteka, kuri pritaiko „Gabor“ filtrus tam tikra eilės tvarka ir išgauna šiek tiek švaresnį atspaudo vaizdą.

Pasinaudojant „OpenCV“ funkcijomis, pritaikyti „Gaussian Blur“ ir „Otsu threshold“ algoritmai. Pirmasis leidžia minimaliai sulieti paveikslą, kad būtų sušvelninami kai kurie aštrūs kampai ir išsišakojimai, kurie galimai atsirado dėl šiukšlių ar triukšmo darant piršto atspaudo nuotrauką. Antrasis algoritmas perskaičiuoja visų pikselių reikšmes, atsižvelgdamas į nurodytą atskaitos reikšmę ir paverčia visus pikselius juodais arba baltais, taip binarizuodamas visą paveikslą. Pritaikius „Gabor“ filtrus, paveikslas yra invertuojamas, todėl reikia atstatyti jo spalvas. Tam panaudota „OpenCV“ funkcija, leidžianti invertuoti bitų reikšmes.

Atlikus šiuos žingsnius gaunamas paveikslas labai panašus į originalų atspaudo paveikslą (palyginti paveikslai matomi 13 paveiksle), tačiau šis gautas paveikslas jau yra tinkamas ploninimo operacijai, nes yra pravalytas nuo triukšmų ir binarizuotas.

13 pav. Piršto atspaudas, po apdorojimo

3.2.2. Linijų ploninimas

Šio etapo metu, linijos suploninamos iki 1 pikselio storio. Tai reikalinga, norint supaprastinti atspaudo požymių suradimo etapą. Ploninimas atliekamas tos pačios „enchancer“ bibliotekos pagalba, kuri panaudoja „skimage“ bibliotekos išteklius, ir sukuria 14 paveiksle matomą vaizdą. Gautas vaizdas šiek tiek skiriasi nuo originalaus paveikslo, gali atsirasti įvairių išsišakojimų, dėl netolygaus originalaus paveikslo vaizdo – todėl panaudojamas prafiltravimas, kuris palygina gautą paveikslą su originaliu ir pravalo rastus nukrypimus.

14 pav. Suplonintas vaizdas

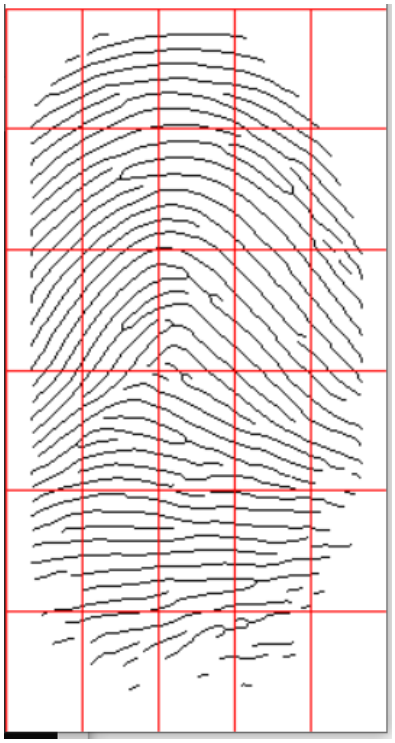
3.2.3. Koordinačių paruošimas

Turint paruoštą piršto atspaudo paveikslą, reikia sudaryti koordinačių sistemą skirtą šiam paveikslui. Tam naudojama vartotojo įvesta skaičių eilutė. Pasinaudojant „hashlib“ biblioteka, suskaičiuota vartotojo įvestos simbolių eilutės hash256 reikšmė.

```
sha256 values for X and Y:
42a3ef3e47ee8de6cc67a8fa9451e29d
83730b28b127ab9d52095afec20bcb54
sha256 values for X and Y (int32 form):
185993026694804564613730055038656945869880232237
369973559301934633238803485012050267549637946532
```

15 pav. Hash reikšmių pavyzdys

Gautas šio algoritmo rezultatas, padalinamas į dvi lygias dalis (matoma 15 paveiksle). Pirmoji dalis bus X koordinatė, o antroji – Y koordinatė. Ant paveikslo nupiešiamas 5x6 stačiakampių tinklelis (pavaizduotas 16 paveiksle). Šios koordinatės pritaikomos centriniam koordinačių sistemos stačiakampiui, o aplinkiniams stačiakampiams koordinatės suteikiamos atsižvelgiant į jų buvimo vietą centrinio stačiakampio atžvilgiu.



16 pav. Koordinačių tinklelis ant atspaudo nuotraukos

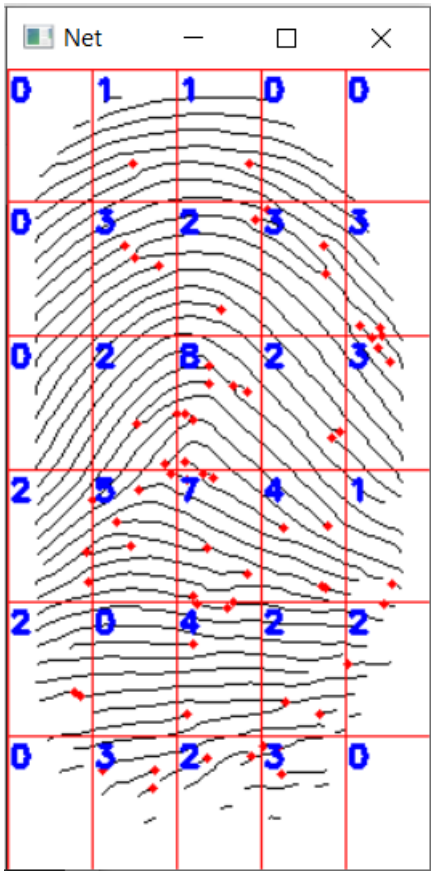
3.2.4. Požymių paieška

Atlikus visus pasiruošimo etapus, galima pradėti požymių paiešką.

Atspaudo požymių paieška atliekama naudojantis algoritmu, kuris pagrįstas „susikertančių skaičių technika“ (angl. Crossing number technique). Paprasčiau tariant, algoritmas tikrina kiekvieną paveikslo pikselį ir skaičiuoja kiek aplink jį, vieno pikselio atstumu, yra juodų pikselių. Pagal tai galima nuspręsti ar tai yra linijos vidurys, linijos pabaiga ar išsišakojimas.

Radus išskirtinius taškus, jų koordinatės surašomos į masyvą.

Atlikus paiešką, prafiltruojami gauti rezultatai ir išmetami taškai esantys paveikslo kraštuose. Gaunamas galutinis rezultatas grafiškai pavaizduojamas ekrane (17 paveikslas).



17 pav. Rasti atspaudo požymiai

Stačiakampių kampuose yra pavaizduoti skaičiai, nurodantys, kiek požymių aptikta tame stačiakampyje. Šie skaičiai reikalingi tinklelio pozicionavimui iššifravimo etape. Pasinaudojant šiais skaičiais sudaromos eilučių ir stulpelių skaičių sandaugos, gauti rezultatai surašomi į vieną masyvą ir atspausdinami ekrane (18 paveikslas). Šie skaičiai vėliau gali būti panaudojami tinklelio pozicijos koregavimui, kadangi vienam taškui perėjus į kitą langelį, pasikeičia bent dvi eilučių ir stulpelių sandaugos reikšmės. Pagal tai galima nuspręsti ar reikia slinkti aukštyn/žemyn ar kairėn/dešinėn.

```
Features found:  65
MatrixValue:  [4, 90, 896, 144, 18, 1, 54, 96, 280, 32, 18]
```

18 pav. Sandaugos

3.2.5. Sandaugų reikšmių vidutinė sandauga

Tam, kad būtų galima išvengti begalinio algoritmo veikimo, buvo sugalvota naudoti vidutinė sandaugos reikšmę. Ši reikšmė leidžia sustabdyti algoritmo veikimą tuomet, kai reikšmė pasidaro artima pradinio paveikslo reikšmei su tam tikra paklaida. Paklaidos reikšmė, bus nustatoma eksperimento metu. Reikšmė apskaičiuojama pagal formulę:

$$Rezultatas = \frac{\prod x}{\frac{\sum x}{12}}; \quad (2)$$

čia x – yra sandaugų eilutės reikšmės.

Skaičius 12 parinktas artimiausias lyginis skaičius nuo tinklelio dydžių sumos, t.y. 5 + 6 = 11, artimiausia didesnė lyginė reikšmė – 12.

Šis ankstesnis algoritmo sustabdymas sumažina tikimybę, kad algoritmas pateks į begalinį ciklą. Ciklas gali prasidėti, kuomet vieninteliai galimi tinklelio pastūmimo veiksmai pradeda kartotis ir anuliuoja vienas kitą, kaip pvz.: pirmas pastūmimas 1 pikselis x ašimi, antras pastūmimas -1 pikselis x ašimi. Šiems veiksmams pradėjus kartotis, algoritmas „pakimba“ amžiname cikle. Jei vidurkis pasidaro priimtinas ankščiau nei prasideda ciklas – algoritmas sustabdomas.

Taip pat ši funkcija padeda teisingiau surasti problemos sprendimą. Kadangi yra sudėtinga atkartoti identiškas reikšmes kiekviename tinklelio stačiakampyje – gavus panašią vidutinę reikšmę daroma prielaida, kad algoritmas jau pasiekė reikiamą postūmį.

3.2.6. Rakto generavimas

Šifravimo raktas sugeneruojamas susumuojant stačiakampių koordinates. Sumuojamos, tik tų langelių koordinatės, kuriuose yra daugiau nei 2 atspaudo požymiai. Gautą reikšmę galima interpretuoti kaip raktą, arba kaip pradinę reikšmę rakto generavimo algoritmui.

Sugeneravus algoritmo rezultatą, gautas skaičius parodomas ekrane (19 paveikslas).

```
sha256 values for X and Y:
42a3ef3e47ee8de6cc67a8fa9451e29d
83730b28b127ab9d52095afec20bcb54
sha256 values for X and Y (int32 form):
185993026694804564613730055038656945869880232237
369973559301934633238803485012050267549637946532
Features found:  65
MatrixValue:  [4, 90, 896, 144, 18, 1, 54, 96, 280, 32, 18]
Generated key:  611563244596413117637786894055779347614699966366
```

19 pav. Algoritmo rezultatas

3.3. Eksperimentas

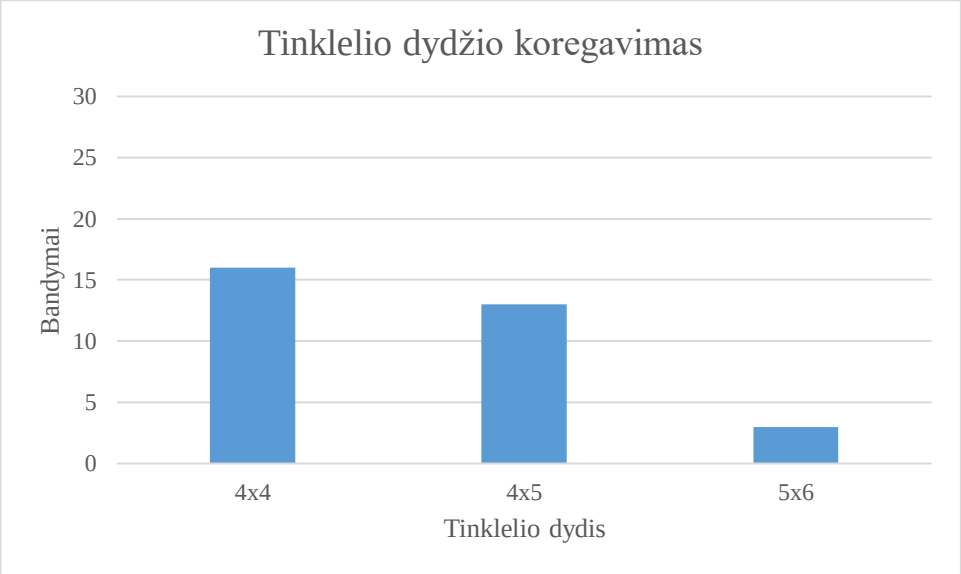
3.3.1. Eksperimento planas

1. Parametrų koregavimas:
   - įvertinti kokią įtaką daro tinklelio dydžio koregavimas;
   - įvertinti kokią įtaką daro minimalaus požymių kiekio langelyje koregavimas;
   - įvertinti kokią įtaką daro reikšmių vidurkio atskaitos koregavimas;
   - įvertinti kombinuotą šių parametrų keitimo įtaką.
2. Suradus optimaliausią variantą, atlikti patikimumo vertinimą:
   - neteisingi rezultatai (angl. False positive) įvertinimas;
   - negautas joks rezultatas;
   - teisingas rezultatas.
3. Įvertinti kokią įtaką daro maišos funkcijos ilgio keitimas.

3.3.2. Eksperimento rezultatai

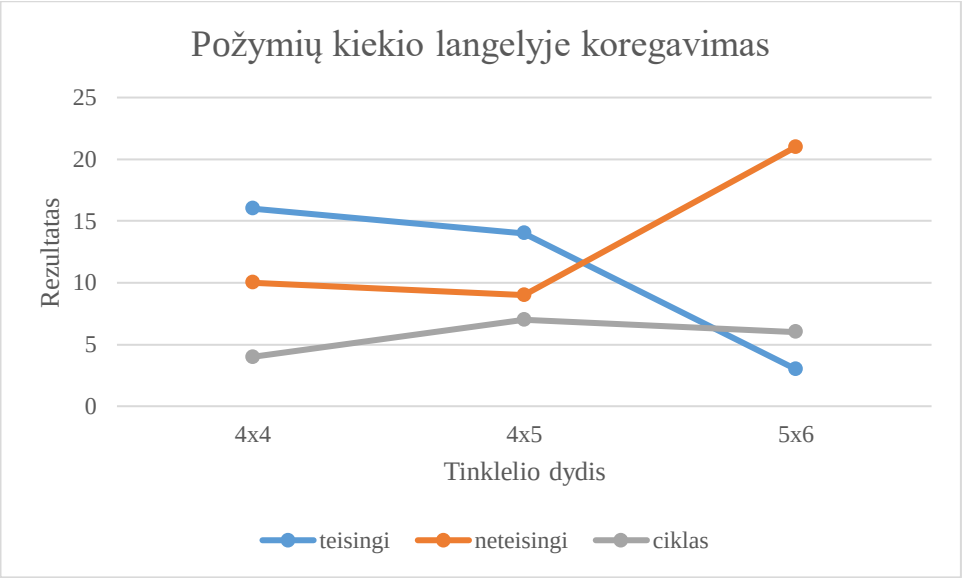
Eksperimentas buvo atliktas naudojantis ir koreguojant aukščiau aprašytą prototipą. Eksperimento etapai atlikti tokia eilės tvarka kokia jie aprašyti. 2 etapas atliekamas sukombinavus visų bandymų geriausius rezultatus. Visi bandymai buvo atlikti naudojant sudarytą 30 paveikslų rinkinį, kurį sudaro įvairios kokybės paveikslai (remiantis 30 šaltiniu). Visi paveikslai buvo paruošti taip, kad posūkio kampas būtų vienodas. Paveiksluose skiriasi atspaudo padėtis plokštumoje.

1. Tinklelio dydžio koregavimo bandymas.
   Tinklelio dydžio koregavimo bandymas buvo atliktas su 3 dydžių tinkleliais: 4x4, 4x5 ir 5x6 langelių. Gautas rezultatas matomas 20 paveiksle. Atlikus bandymą pastebėta, kad mažėjant tinkleliui, didėja tikimybė teisingai atkurti tą pati rezultatą. Kadangi reikšmės pasiskirsto per mažesnį kiekį langelių, padidėja požymių skaičius visuose langeliuose. Dėl to, leidžiama didesnė paklaida, nes yra didesnė tikimybė, kad langeliai turės didesnį kiekį požymių, nei reikalaujamas minimalus kiekis. Pasiekus minimalią požymių ribą, algoritmas neatsižvelgia kiek ji yra viršijama ir langelis užskaitomas kaip tinkantis rakto generavimui. Gautas rezultatas 4x5 tinkleliui – 13/30 pavykusių bandymų.



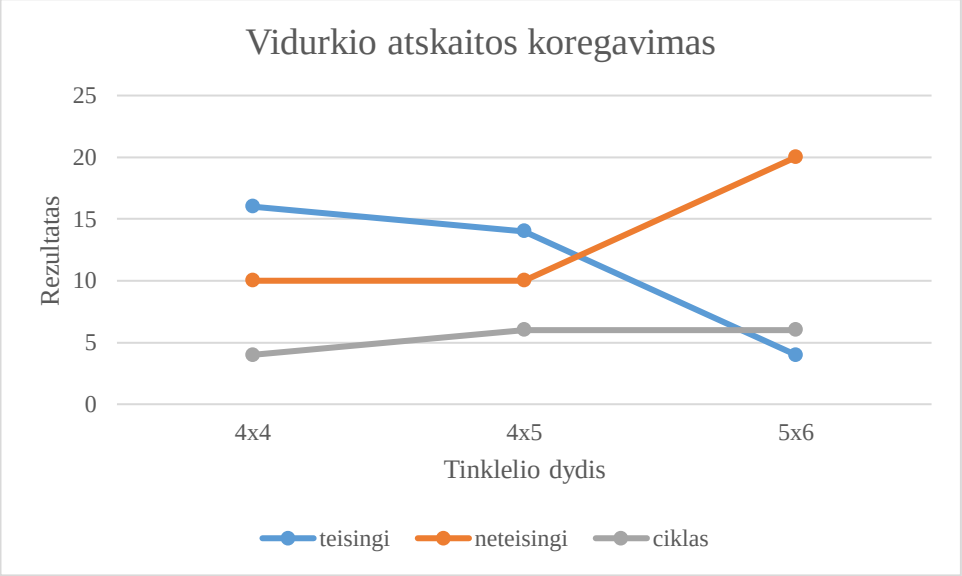
20 pav. Tinklelio dydžio koregavimo eksperimento rezultatas

2. Minimalaus požymių kiekio langelyje koregavimo bandymas.
   Atliekant bandymą, buvo pastebėta, kad tiesiog keičiant minimalų požymių kiekį, paklaida tik didėja, bet jei minimalus kiekis nustatomas, pagal apskaičiuotą vidutinę sandaugų reikšmę – gaunami šiek tiek tikslesni rezultatai. Kuomet bendra sandaugos reikšmė yra maža – yra mažiau požymių piršto atspaude – tuomet minimali požymių langelyje riba yra mažesnė. Kuomet sandaugos reikšmė yra didesnė – minimali požymių langelyje riba yra didesnė.
   Pasirinkta skaičiuoti su 2 ir 3 požymių minimaliomis ribomis. Gautas rezultatas matomas 21 paveiksle. Atlikus bandymą su paveikslų rinkiniu – rezultatas labai artimas kaip ir pirmuoju atveju. Gautas rezultatas 4x5 tinkleliui – 14/30 pavykusių bandymų.



21 pav. Požymių kiekio langelyje koregavimo rezultatas

3. Reikšmių vidurkio atskaitos koregavimo bandymas.
   Reikšmių vidurkio atskaitos reikšmės koregavimas taip pat buvo atliktas remiantis apskaičiuota reikšmių vidurkio reikšme. Sudaryti 3 reikšmių intervalai, kurie keičia galimą antrojo paveikslo skaičiavimo paklaidą: 1.1 , 1.175 ir 1.25. Kuo mažesnė pradinė vidurkio reikšmė, tuo didesnė paklaida jam leidžiama skaičiuojant antrą kartą, pvz.: jei vidurkis gautas 100 – tai jo galima paklaida yra +/- 10. Jei gautas vidurkis yra 10000 – tai jo galima paklaida yra +/- 2500. Toks reikšmių suskirstymas leido šiek tiek atlaidžiau nuspręsti ar galima baigti paveikslo koregavimą ir apskaičiuoti rezultatą. Kuo didesnės leidžiamos paklaidos – tuo didesnė tikimybė gauti neteisingą raktą – nes algoritmas per anksti nuspręs sustoti. Jei paklaida per maža arba ji neleidžiama, didėja tikimybė patekti į begalinį ciklą, kur algoritmas bandys koreguoti tinklelio poziciją ir rasti geriausią variantą. Atliekant bandymus nuspręsta pasirinkti būtent aukščiau aprašytas reikšmes, nes pastebėta, kad jos duoda šiek tiek geresnį rezultatą, nei originalus algoritmo variantas be leidžiamos paklaidos. Bandymo rezultatas matomas 22 paveiksle. Gautas rezultatas 4x5 tinkleliui – 14/30 pavykusių bandymų.



22 pav. Vidurkio atskaitos koregavimo rezultatas

4. Kombinuotas koregavimo įvertinimas.
   Sukombinavus prieš tai atliktus pakeitimus į bendrą visumą ir atlikus testavimą gautas rezultatas identiškas pradiniam variantui. Todėl galima teigti, kad atlikti pakeitimai turi mažą įtaką bendroje visumoje. Gautas galutinis rezultatas matomas 4 lentelėje. Viena vieta, kur matomas pagerėjimas nuo pradinio varianto – yra sumažėjusi begalinių ciklų rizika.

4 lentelė. Bandymų rezultatai

Tinklelio dydis	**4x4**	**4x5**	**5x6**
Teisingi rezultatai	16/30	13/30	4/30
Begalinis ciklas	5/30	8/30	6/30
Neteisingi rezultatai	9/30	9/30	20/30

4 lentelę galima interpretuoti ir kaip teisingo teigiamo (angl. true positive) ir neteisingo teigiamo (angl. false positive) rezultatų vertinimą. Teisingi rezultatai lentelėje atitinka teisingo teigiamo vertinimą, o neteisingi rezultatai lentelėje atitinka neteisingo teigiamo rezultato vertinimą. Kadangi begalinio ciklo metu algoritmas neduoda jokio rezultato – nėra standartizuoti vertinimo kriterijaus tokiai būsenai.

5. Maišos funkcijos ilgio įtakos vertinimas.
   Skirtingų ilgių maišos funkcijos panaudojimas galutinio rakto generavimui, gražina skirtingų ilgių reikšmes. Buvo išbandytos SHA1, SHA224, SHA256, SHA384 ir SHA512 funkcijos. Kuo ilgesnis funkcijos rezultatas, tuo sudėtingesnis raktas gaunamas. Gautinio rakto sudėtingumas iš esmės remiasi į maišos funkcijos ilgį – kuo ilgis didesnis, tuo mažesnė tikimybė atsitiktinai atkartoti reikšmę – tuo saugesnis raktas.

3.3.3. Eksperimento išvados

1. Atlikus koregavimą prototipo kūrimo ir eksperimento metu, gautas rezultatas yra apytiksliai 1 iš 3 teisingų sprendimų. Algoritmas nesuveikė pakankamai gerai, kad būtų galima jį naudoti, tačiau matomas galimas potencialas. Jei pavyktų gauti daugiau tinkamų testavimui paveikslų, suskirsčius reikšmių vidurkius į daugiau intervalų, pakoregavus jų reikšmes turėtų pavykti išgauti geresnį algoritmo variantą.
2. Pritaikius didesnio ilgio maišos algoritmą, rakto saugumas didėja, tačiau piršto atspaudo atkartojimo tikimybė nesikeičia. Taip pat būtų galima bandyti pritaikyti klaidų taisymo algoritmus, tačiau jų panaudojimui reikėtų pergalvoti algoritmo veikimą, nes reikėtų išsaugoti daugiau papildomų duomenų.
3. Palyginimui su algoritmu, kuris išsaugo daugiau duomenų ir naudoja juos iššifravimo metu, šiame darbe aprašytas algoritmas atsilieka. 17 šaltinyje aprašytas algoritmas panaudojantis šifravimo metu sugeneruotus duomenis išgauna netoli 99% tikslumą su specialiai ruoštu duomenų rinkiniu, tačiau tam naudoja didesnį išsaugomų duomenų kiekį, ko buvo stengiamasi išvengti šiame darbe aprašyto algoritmo kūrimo metu.

Išvados ir rezultatai

1. Atlikus esamų algoritmų analizę, padaryta išvada, kad visi esami aprašyti biometriniai rakto generavimo metodai yra tik mokslinio lygmens. Daugelis algoritmų aprašyti ir įgyvendinti bei išbandyti atliekant tam tikrus eksperimentus. Pagrindinė to priežastis – sudėtinga išgauti vienodus biometrinis duomenis darant kelias imtis.
2. Atlikus prototipo realizavimą ir eksperimentą, galima daryti išvadas, kad algoritmas nėra labai tikslus. Gaunamas rakto ilgis gali būti redaguojamas pagal maišos funkcijos rezultato ilgį, bei didinant vartotojo įvedamos eilutės ilgį. Nors egzistuoja aprašytų algoritmų išgaunančių aukštesnį patikimumą, iš esmės pats algoritmas galėtų būti naudojamas sistemoje, kur nėra reikalingas aukštas saugumo ir patikimumo lygmuo.
3. Algoritmo teisingų suveikimų skaičių galima didinti mažinant tinklelio dydį, tačiau tada mažėja algoritmo patikimumas, nes yra didesnė tikimybė atkartoti tas pačias reikšmes su kitu atspaudo paveikslu.
4. Algoritmą galima naudoti sistemoje su sąlyga, kad vartotojų pirštų atspaudų paveikslai algoritmui bus paduodami pasukti tuo pačiu kampu. Skenuojant vartotojo piršto atspaudą turėtų būti apibrėžta zona, kurioje vartotojas turi tiksliai padėti pirštą, kad būtų gaunamas kuo panašesnis į originalą paveikslas. Tokiu atveju didėja tikimybė teisingai atkurti šifravimo raktą.

Literatūros sąrašas

1. Ali, S. H. “Novel Approach for Generating the Key of Stream Cipher System Using Random Forest Data Mining Algorithm.” In *2013 Sixth International Conference on Developments in ESystems Engineering*, 259–69, 2013.

   Staraipsnis apie raktų generavimą panaudojant „Random Forest Data Mining“ algoritmą
2. Piyush Naik, Karthikeyan Ravichandran, Krishna M. Sivalingam. “Cryptographic key exchange based on locating information,” *Pervasive and Mobile Computing Volume 3, Issue 1*. January 2007

   Straipsnis apie raktų apsikeitimą naudojant vietos informaciją patalpose.
3. Mohsin, R. M., R. I. Ahmed, R. Yaqub, and S. Ethar. “A New Technique for Diffie-Hillman Key Exchange Protocol Security Using Random Image Generation.” In *2019 First International Conference of Computer and Applied Sciences (CAS)*, 262–67, 2019.

   Metodas aprašantis Diffie-Hellman raktų generavimą iš paveikliukų.
4. Biswas, Barun, Krishnendu Basuli, and Samar Sarma. “ON A KEY EXCHANGE TECHNIQUE, AVOIDING MAN-IN-THE-MIDDLE- ATTACK,” September 16, 2010.

   Straipsnis apie raktų apsikitimą išvengiant „Man-In-The-Middle“ atakos
5. Eng, Haitham Wahdan, Abdel-moneim Wahdan Prof Dr, Aliaa A. A. Youssif, and Prof Dr. “*Cryptosystem from Multiple Biometric Modalities“*, n.d.

   Straipsnis pasakoja apie kelių biometrijos sričių kombinavimą kriptografijoje
6. Oluwakemi Christiana Abikoye, Umar Abdulraheem Ojo, Joseph Bamidele Awotunde, Roseline Oliwaseun Ogundokun. “A safe and secured iris template using steganography and cryptography” 10 June 2020

   Straipsnis aprašo saugaus šablono sukūrimo ir saugojimo techniką.
7. Bhattacharyya, Debnath, and Rahul Ranjan. “Biometric Authentication: A Review.” *Science and Technology* 2, no. 3 (2009): 16.

   Straipsnis apžvelgia biometrinių šifravimo raktų panaudojimą
8. Robertas Damaševičius, Rytis Maskeliūnas, Egidijus Kazanavičius, Marcin Wozniak “*Combining Cryptography with EEG Biometrics*”, Hindawi Computational Intelligence and Neuroscience Volume 2018, Article ID 1867548, 11 pages

   Straipsnis aprašantis smegenų signal panaudojimą kriptografijoje
9. Rashid Mofeed, and Huda Zaki. “RSA Cryptographic Key Generation Using Fingerprint Minutiae.” *Iraqi Journal for Computers and Informatics* 41, no. 1 (December 31, 2014): 66–69.

   Straipsnyje aprašomas raktų generavimo metodas naudojantis piršto atspaudo, gijų susikirtimo vietų žemėlapį
10. Xi, Kai, Tohari Ahmad, Fengling Han, and Jiankun Hu. “A Fingerprint Based Bio-Cryptographic Security Protocol Designed for Client/Server Authentication in Mobile Computing Environment.” *Security and Communication Networks* 4, no. 5 (2011): 487–99.

    Straipsnis aprašo piršto atspaudo panaudojimą autentifikacijai kliento/serverio architektūros mobiliose sistemose.
11. Yao-Jen Chang, Wende Zhang, and Tsuhan Chen. “Biometrics-Based Cryptographic Key Generation.” In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 3:2203-2206 Vol.3, 2004.

Biometrinių raktų generavimo metodų apžvalga.

12. Zainulina, E. T., and I. A. Matveev. “Binding Cryptographic Keys into Biometric Data: Optimization.” *Journal of Computer and Systems Sciences International* 59, no. 5 (September 1, 2020): 699–711.

    Straipsnis aprašantis susiejimą tarp biometrinių duomenų ir kriptografijos

13. Dr. Algimantas Venčkauskas, Povilas Nanevičius “*Cryptographic Key Generation from Finger Vein*”, April 2013

    Straipsnyje aprašomas raktų generavimo metodas panaudojant piršto kraujagysles.

14. Akhila V. A., Arunvinodh C., Reshmi K. C., Sakthiprasad K. M. “*A New Cryptographic Key generation Scheme Using Psychological Signals*”, Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016)

    Straipsnyje aprašomas kriptografinių raktų generavimo metodas panaudojant smegenų signalus.

15. Zumrut Muftuoglu, Tulay Yildirim “*Comparative Analysis of Crypto Systems Using Biometric Key*”, 8th International Congress of Information and Communication Technology, ICICT 2019

    Straipsnyje palyginamos biometrinės kriptografinės sistemos

16. Mondira D., Mohamed A., Tanay “*Biometric cryptography using micromachined ultrasound transducers*“, December 12, 2017

    Straipsnyje aprašomas kriptografinis metodas panaudojant ultragarso jutiklius piršto nuskaitymui.

17. Gaurang Panchal, Debasis Samanta “*A Novel Approach to Fingerprint Biometric-Based Cryptographic Key Generation and its Applications to Storage Security*”, February 1, 2018

18. Jain, A. K., Nandakumar, K., & Nagar, A. „*Biometric Template Security“.* Hindawi Publishing Corporation, EURASIP Journal on Advances in SignalProcessing, 1-17, 2008
    Straipsnis apie įvairius piršto atspaudo šablono saugumo aspektus.

19. Li, P., Yang, X., Cao, K., Tao, X., Wang, R., & Tian, J. „*An alignment-free fingerprint cryptosystem based on fuzzy vault scheme*“. Journal of Network and Computer Applications, 33(3), 207-220, 2010
    Straipsnis apie šablono ir atspaudo vaizdo palyginimą išvengiant keleto pagrindinių tikrinimo pažeidžiamumų.

20. Marino, R. A., Alvarez, F. H., & Encinas, L. H. „*A crypto-biometric scheme based on iristemplates with fuzzy extractors*“. Elsevier, Information Sciences, 195, 91-102, 2012
    Straipsnis apie akies rainelės šablono kūrimą ir pnaudojimą biometrinės kriptografijos raktams.

21. Eskander, G. S., Sabourin, R.,& Granger, E. „*A bio-cryptographic system based on offline signature images*“. Elsevier, Information Sciences, 259, 170-191, 2014
    Straipsnis apie šifravimą panaudojant asmens parašo vaizdą.

22. Amirthalingam, G., & Radhamani, G. „*New chaff point based fuzzy vault for multimodal biometric cryptosystem using particle swarm optimization*“. Elsevier Journal of King Saud University-Computer and Information Sciences, 28, 381-394, 2016
    Straipnsis apie kelių skirtingų biometrinių duomenų apjungimą ir panaudojimą.

23. Adamovic, S., Milosavljevic, M., Veinovic, M., Sarac, M., & Jevremovic, A. „*Fuzzy commitment scheme for generation of cryptographic keys based on iris biometrics*“. IET Biometrics, 6(2), 89-96, 2017
    Straipsnis apie kriptografinio rakto generavimą iš akies rainelės.

24. Yang, W., Wang, S., Zheng, G., Chaudhry, J., & Valli, C. „*ECB4CI: an enhanced cancelable biometric system for securing critical infrastructures*“. Springer Science and Business Media, LLC, 74, 4893– 4909, 2018
Straipnis apie biometrinių duomenų panaudojimą asmens identifikavimui kritinio saugumo sistemose.
25. Chitra, D., & Sujitha, V. „*Security analysis of prealigned fingerprint template using fuzzy vault scheme*“. Cluster Comput, 22, 12817–12825, 2018
Straipsnis apie pirsto atspaudo panaudojimą kriptografinio rakto generavimui ir jo stiprumo įvertinimą.
26. Elrefaei, L. A., & Al-Mohammadi, A. M. „*Machine vision gait-based biometric cryptosystem using a fuzzy commitment scheme“.* Journal of King Saud University, Computer and Information Sciences, 1-14., 2019
Straipsnis apie mašininio mokymo algoritmo panaudojimą šifravimo rakto generavimui.
27. Ponce-Hernandez, W., Blanco-Gonzalo, R., Liu- Jimenez, J., & Sanchez-Reillo, R. „*Fuzzy Vault Scheme Based on Fixed-Length Templates Applied to Dynamic Signature Verification*“. IEEE Access, 8, 11152-11164, 2020
Straipsnis apie biometrinio rakto naudojimą su papildomu verifikavimo žingsniu.
28. Neeraj Tantubay. „*A Survey of Biometric Key-Binding Biocrypto-System using different Techniques*”. December 2019
Straipsnis palyginantis kelis skirtingus bio kriptografijos metodus.
29. Erwin et al „The Enhancement of Fingerprint Images using Gabor Filter” 2019 J. Phys.: Conf. Ser. 1196 012045
Straipsnis apie Gabor filtro panaudojimą apdorojant piršto atspaudus.

Informacijos šaltiniai

30. NIST Special Database 302. https://www.nist.gov/itl/iad/image-group/nist-special-database-302 [žiūrėta 2022-05-10]