

**UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I RAČUNARSTVO**

**ANALIZA TRANSFORMACIJE BIOMETRIJSKIH PODATAKA
I NJIHOVA PRIMENA U KRIPTOGRAFSKIM OPERACIJAMA**

DOKTORSKA DISERTACIJA

Mentor:

Prof. dr Milan M. Milosavljević

Kandidat:

mr Goran Kunjadić

Beograd, 2014.

ANALIZA TRANSFORMACIJE BIOMETRIJSKIH PODATAKA I NJIHOVA PRIMENA U KRIPTOGRAFSKIM OPERACIJAMA

Univerzitet Singidunum

Fakultet za informatiku i računarstvo

Beograd, Srbija



Doktorska Disertacija
Univerzitet Singidunum
Fakultet za informatiku i računarstvo
Beograd, Srbija, 2014.

Mentor:

Prof. dr Milan M. Milosavljević

Članovi komisije:

Prof. dr Branko Kovačević

Prof. dr Mladen Veinović

Goran Kunjadić

**ANALIZA TRANSFORMACIJE BIOMETRIJSKIH PODATAKA
I NJIHOVA PRIMENA U KRIPTOGRAFSKIM OPERACIJAMA**

Doktorska disertacija

Goran Kunjadić

**ANALYSIS OF TRANSFORMATION BIOMETRIC DATA
AND THEIR USE IN CRYPTOGRAPHIC OPERATIONS**

Doctoral dissertation

Mentor: Prof. dr Milan M. Milosavljević

2014

Zahvalnosti

Veliku zahvalnost dugujem svojoj porodici, Suzani, Filipu i Kristini, za podršku, strpljenje i razumevanje koje su pokazali za vreme koje sam im oduzeo i posvetio pisanju ovog rada.

Takođe se zahvaljujem svim ostalima koji su na različite načine doprineli izradi ovog rada.

U Beogradu,

Božić, 2014.

Predgovor

Genetski materijal koji roditelji prenose na svoje potomke u potpunosti određuje fizičke karakteristike nove jedinke. Prirodne karakteristike ljudskog tela su vidljive korišćenjem ultrazvučnog pregleda već u prenatalnom periodu. Pojedine fizičke karakteristike se značajno menjaju u toku života, dok druge ostaju iste, praktično, celog života. Iskustvom je zaključeno da se: otisak prsta, iris, retina i geometrija lica veoma malo menjaju u toku života pa se mogu smatrati konstantnim biometrijskim podacima. Ovi podaci se mogu koristiti za identifikaciju osobe sa visokim stepenom tačnosti. Mentalni sklop ličnosti koji će usloviti model ponašanja u velikoj meri je definisan kodom zapisanim u genetskom materijalu. Modeli ponašanja su, pored mentalnog sklopa ličnosti, uslovljeni i uticajima okoline, tako da specifični način ponašanja pojedinca može biti izazvan i spoljašnjim uticajima. Ipak, model ponašanja kao odgovor na spoljašnje uzroke može se uočiti u vidu obrasca za pojedine tipove ličnosti, a samim tim i za pojedinca koji pripada nekom od tipova ličnosti. Postoje postupci koji omogućavaju identifikaciju osobe na osnovu modela ponašanja.

Biometrija je interdisciplinarna nauka koja proučava i koristi merljive biološke karakteristike i koja objedinjava prvenstveno biologiju i matematiku. Akvizicija i obrada biometrijskih podataka, prema savremenim merilima, podrazumevaju korišćenje računara i računarskih programa, kao moćnog alata za sofisticiranu obradu velike količine podataka. Sam proces akvizicije podataka zahteva korišćenje specijalnih hardverskih komponenti kao što su skeneri različitih tipova koji se koriste za različite namene. Standardni algoritmi kao i priroda digitalnih računara sadrže ograničenja uzrokovana koncepcijom i načinom rada digitalnih računara. Uprkos ograničenjima razvijeni su specijalizovani sistemi za obradu signala dobijenih sa skenera i kreiranje reprezentata biometrijskog podatka. Razvijene su posebne vrste sistema koje imaju za cilj obradu podataka dobijenih na ovaj način.

Biometrijsko prepoznavanje je prirodni način razlikovanja jedinki iste vrste na osnovu njihovih fizioloških karakteristika kao i na osnovu načina njihovog ponašanja. Čovekov um je razvijao, i uspešno razvio, sposobnost prepoznavanja drugih osoba tokom dugog perioda evolucije, kao jednu od bitnih funkcija u borbi za opstanak. U ljudskoj svesti funkcionišu biometrijski mehanizmi koji veoma uspešno razlikuju individue jednu od druge. Algoritmi za identifikaciju osobe koje je razvio i koristi ljudski mozak služe kao polazna osnova i, za sada nedostižan, ideal, za razvoj sistema biometrijskog prepoznavanja. Način funkcionisanja ljudskog mozga, misaoni procesi koji se u njemu odvijaju kao i sama svest, i pored značajnih naučnih prodora, sadrže više pitanja nego odgovora.

Postojeći algoritmi imaju sposobnost izdvajanja karakteristika otiska prsta (minucija), dok njihovo upoređivanje i donošenje odluke o identičnosti dva otiska nije tako jednostavan problem, kako u matematičkom smislu tako ni u praktičnim primenama. Prepoznavanje otiska je složen problem i u tu svrhu su razvijene različite metode i različiti algoritmi, korišćenjem složenog matematičkog aparata. Prepoznavanje otiska prsta postaje još kompleksniji problem ako se radi o korisnicima koji su nekooperativni ili je prst zaprljan ili je u ožiljcima. U tom slučaju kao rezultat skeniranja dobijena slika, koja služi kao polazna osnova za dalju obradu, je lošeg kvaliteta. Potrebno je napomenuti da oko 5% svetske populacije, većinom poreklom iz Azijskih zemalja, ima veoma plitke papile - krive linije na jagodici prsta, tako da tom delu populacije nije moguće skenirati prst niti dobiti sliku otiska. Postoji široko rasprostranjeno mišljenje da je automatsko prepoznavanje otiska prsta nepogrešivo i da je rešilo problem identifikacije, što je daleko od istine jer je prepoznavanje otiska prsta još uvek složen proces, zavisao od mnogobrojnih faktora, iako se u njegovom razvoju i njegovoj primeni najdalje otišlo. Otisak prsta je prva biometrijska veličina koja se koristila u cilju identifikacije osobe, tako da je sasvim logično što je otisak prsta najviše korišćena biometrijska osobina. Postoji još jedan bitan momenat koji favorizuje korišćenje otiska prsta a to je jednostavnost akvizicije samog otiska, što nije zanemarivo kada je potrebno dobiti rezultat u realnom vremenu.

Biometrijski podaci imaju tu osobinu da se ne reprezentuju uvek na isti način zbog standardne greške merenja kao i zbog prirode samih biometrijskih veličina. Ukoliko se posmatra uzimanje otiska prsta očigledna je činjenica da se trodimenzioni oblik same površine - jagodica prsta, transformiše u dvodimenzionu sliku. Pomenuta transformacija sama po sebi izaziva greške i dovodi do izobličenja originalnog podatka. Izuzetak od ovog pravila čini DNK (Dezoksiribonukleinska kiselina), čija analiza obezbeđuje nedvosmislenu identifikaciju osobe. Nedostatak DNK analize je dužina trajanja samog procesa, te nije podesna za praktične upotrebe gde se zahteva identifikacija u realnom vremenu, kao što je identifikacija putnika na aerodromima. Čine se pokušaji izgradnje sistema za automatsko utvrđivanje identiteta na osnovu biometrijskih karakteristika. Najčešće se koriste karakteristike kao što su: otisak prsta, lice, iris, DNK retina, glas, dlan, geometrija vena nadlanice, rukopis, način ponašanja itd. Prepoznavanje i razlikovanje biometrijskih karakteristika se svodi na prepoznavanje oblika što čini jednu od osnovnih postavki veštačke inteligencije.

Biometrija, kao način identifikacije, nije u potpunosti podržana od strane nekih verskih i društvenih zajednica. Razne grupe za zaštitu ljudskih prava smatraju da je davanje reprezenata biometrijskih podataka na upotrebu i čuvanje istih, čin narušavanja privatnosti. Pomenute grupe su naročito jake u SAD gde se otisak prsta uzima samo registrovanim kriminalcima, i osobama koje dolaze u SAD iz drugih zemalja. U Srbiji je privatnost višestruko narušena, ne samo obaveznim uzimanjem otiska prsta i geometrije lica svakog građanina, već i javnim štampanjem otiska prsta i lika na obrascu polikarbonatne kartice sa mikroprocesorom ili bez mikroprocesora- lične karte. Na ovaj način se do otiska prsta bilo koje osobe može doći jednostavnim kopiranjem lične karte.

Pored motiva zaštite privatnosti postoje i religijski motivi, pogotovo ukoliko se radi o implementaciji biometrijskih podataka na kartici sa čipom. U Bibliji, svetoj knjizi svih Hrišćana, u knjizi Otkrovenja stoji: „*I učini (druga „zver“) da svi, mali i veliki, bogati i siromašni, slobodni i robovi, načine sebi žig na svojoj desnoj ruci ili na svome čelu, da niko ne može ni kupiti ni prodati sem onoga ko ima žig, ime zveri ili broj njenoga imena*“ (Otkrovenje 13:16-17).

„*Reč je data čoveku da bi sakrio svoju misao*“, kao što je rekao Jezuitski sveštenik Gabriel Malagrida (*Gabriel Malagrida, 1689 - 1761*). Verovatno je skrivanje sopstvenih misli i sopstvenih namera evolutivno nasleđe ljudskog roda kojim je vrsta obezbedila svoj opstanak. Kriptografija (od grčkih reči κρυπτος, što znači skriven ili tajan i γραφειν, što znači pisati) je nauka koja se bavi zaštitom podataka. Osnovni postulati kriptografije se mogu sažeti u četiri stavke: poverljivost podataka, integritet podataka, identifikacija i neporecivost. Cilj kriptografije je transformisanje izvornog podatka - otvorenog teksta, koji je čitljiv i prepoznatljiv u oblik koji je nerazumljiv neovlašćenom posmatraču. Ovaj proces se naziva šifrovanjem dok se rezultat šifrovanja naziva šifratom. Inverzni proces prevođenja šifrata u otvoreni tekst se naziva dešifrovanjem. Kao rezultat dešifrovanja dobija se početni otvoreni tekst. Kriptografski ključ predstavlja neophodan činilac u kriptografskim operacijama koji omogućava pomenute procese šifrovanja i dešifrovanja. Karakteristike kriptografskog ključa su definisane u kriptografiji. Bitno je napomenuti da se biometrijska veličina ne može direktno, u izvornom obliku, koristiti kao kriptografski ključ, već je neophodna njena prethodna obrada. Uloga kriptografije u savremenom životu postaje sve značajnija. Korišćenje kriptografije u procesu plaćanje platnim karticama propisuju Visa i MasterCard kao vodeći vendori u oblasti platnih karica u svetu, odnosno organizacija EMV (Europay MasterCard Visa) koja je neprofitna organizacija nastala pod pokroviteljstvom pomenuta dva vendora. Kriptografija je nezamenjiva u zaštiti korišćenja računarskih resursa, upotrebi i zaštiti mobilnih uređaja kao i u zaštiti podataka i komunikacija svih vrsta, pogotovo onih namenjenih u vojne i policijske svrhe.

Sprega biometrije i kriptografije se ukazuje kao „dobitna kombinacija“ koja omogućava pouzdanu identifikaciju osobe uz istovremenu zaštitu privatnosti. Moguće je koristiti biometrijske karakteristike kao komponente u standardnim kriptografskim operacijama.

Nakon što su uočena ograničenja klasičnih računarskih sistema u obradi biometrijskih podataka, došlo se na ideju primene fazi tehnologije kao nove metodologije. Fazi logika pokušava da približi način rada digitalnog računara odnosno način obrade informacije - načinu razmišljanja koji se odvija u ljudskom mozgu. Polazeći od činjenice da ulazni signal predstavlja podatak koji osim ciljnih parametara sadrži u sebi i dodatne, ometajuće signale, osnovni zadatak fazi sistema je da razdvoji osnovni podatak od ometajućih parametara - šuma. Ovakvi podaci se obično nazivaju podaci sa šumom. Na osnovu dobijenog čistog signala moguće je dobiti binarni reprezent posmatrane biometrijske veličine koji se u manjoj ili većoj meri može identično reprodukovati prilikom svakog sledećeg merenja. Godine 1965, Lotfi Zadeh (*Lotfi A. Zadeh, 1921*) na Berkli Univerzitetu u Kaliforniji objavio je rad „Fuzzy Sets“ u kome je obradio osnovne postavke fazi skupova a samim tim i fazi logike. Zadeh je

uočio da konvencionalna računarska logika ne može manipulirati podacima koji predstavljaju subjektivne ili nejasne pojmove. Upotreba fazi logike u obradi biometrijskih podataka je otvorila nove horizonte u posmatranoj oblasti. Velike doprinose fazi tehnologije u oblasti obrade i šifrovanja biometrijskih podataka tek treba očekivati.

Binarni reprezent se često naziva i templejtom i može se kreirati korišćenjem različitih metoda na više različitih načina. Jedan od mogućih načina kreiranja templejta biometrijske veličine je kreiranje korišćenjem fazi ekstraktorskih funkcija. Fazi ekstraktorske funkcije izdvajaju željene karakteristike biometrijske veličine, s tim što se ne preciziraju vrednosti izdvojenih karakteristika već se definišu oblasti u kojima se očekuje njihovo pojavljivanje.

Matematički modeli za korekciju greške predstavljaju tehnike za otkrivanje (error-detecting code) i uklanjanje šuma (error-correcting code) u podacima koji se obrađuju. Sistem se bazira na dodavanju redundantnih podataka u poruku, tako da ona može kvalitetno da se reprodukuje čak i ukoliko u nizu koji se prenosi postoje ometajući podaci. Ispravljanje grešaka se može vršiti na nivou bita ili na nivou bloka podataka. Primeri kodova za ispravljanje grešaka u podacima su: Hamingovi kodovi (*Richard Hamming, 1915 - 1998*) zatim Rid-Solomonovi kodovi (*Irving S. Reed, 1923 - 2012; Gustave Solomon, 1930 - 1996*) kao i Slepian Vulfovi kodovi (*David S. Slepian, 1923 – 2007; Jack K. Wolf, 1935 – 2011*) koji će biti primenjeni u ovom radu. Studija kodova za ispravljanje grešaka i odgovarajućih matematičkih postupaka je poznata kao teorija kodiranja.

Tematika obrađena u ovom radu je odraz savremenih težnji za bezbednošću u današnjem svetu punim raznovrsnih pretnji po bezbednost uz istovremeno obezbeđivanje slobode i zaštite integriteta pojedinca. Ako se uzme u obzir činjenica da bezbednosne pretnje kao i „krađe identiteta“ pokazuju tendenciju rasta, postaje očigledno da tema obrađena u ovom radu zaslužuje mnogo više pažnje.

Imajući u vidu iznete činjenice i razmišljanja nameće se zaključak da oblasti: biometrije, prepoznavanja oblika, korekcije greške, kriptografije i fazi tehnologije predstavljaju izazovne oblasti za mnoga istraživanja.

Sadržaj

Predgovor.....	iii
SIŽE	3
ABSTRACT	5
1. UVOD	7
1.1 DEFINISANJE PROBLEMA	8
1.2 CILJEVI ISTRAŽIVANJA.....	10
1.3 HIPOTEZE	11
1.4 METODE ISTRAŽIVANJA.....	15
1.4.1 Testiranje matematičkih modela	15
1.4.2. Razvoj softvera	16
1.5 OČEKIVANI REZULTATI ISTRAŽIVANJA I NAUČNI DOPRINOS	16
1.6 PREGLED SADRŽAJA PO POGLAVLJIMA.....	17
2. PREDMET ISTRAŽIVANJA	19
2.1 ELEMENTI BIOMETRIJE	19
2.1.1 Kratak uvod u biometriju.....	20
2.1.2 Pregled biometrijskih veličina.....	35
2.1.3 Proces biometrijske identifikacije.....	52
2.1.4 Zaštita privatnosti.....	58
2.1.4.1 Elementi Fazi logike.....	59
2.1.4.2 Zaštita privatnosti biometrijskih podataka.....	67
2.2 ELEMENTI KRIPTOGRAFIJE	73
2.2.1 Proces razmene šifrovanih podataka.....	74
2.2.2 Elementi matematičkog aparata koji se koristi u kriptografiji	75
2.2.3 Simetrični šifarski sistemi.....	85
2.2.4 Kriptografija sa javnim ključevima	88
2.2.5 Kriptoanaliza.....	92
3. KODOVI ZA KOREKCIJU GREŠKE.....	99
3.1 ŠENONOVA TEOREMA	100
3.2 OSNOVNI KONCEPTI LINERANIH KODOVA	101
3.3 DRUGE KLASSE KODOVA	104
3.4 SLEPIAN-VULFOVI KODOVI	107
4. KREIRANJE TEMPLEJTA I UPOREĐIVANJE DVA OTISKA PRSTA	111
4.1 UPOREĐIVANJE LATENTNIH OTISAKA.....	111
4.1.2 Izdvajanje minucija	113
4.1.3 Upoređivanje otisaka.....	115
4.2 DELONEOVA TRIANGULACIJA	118
5. PREDLOŽENO REŠENJE	121
5.1 IZBOR ALGORITAMA ZA KREIRANJE ŠABLONA OTISKA PRSTA	121

5.2 IZBOR METODE KODIRANJA	122
5.3 GENERISANJE KRIPTOGRAFSKOG KLJUČA	122
5.4 KOMBINOVANJE TEMPLEJTA OTISKA PRSTA SA KRIPTOGRAFSKIM KLJUČEM	122
5.5 UPOREĐIVANJE DVA OTISKA.....	123
5.6 KONCEPT DVA TEMPLEJTA.....	123
5.7 DEFINISANJE PREDLOŽENOG KONCEPTA	125
5.7.1 Proces kodiranja.....	125
5.7.2 Proces dekodiranja	126
6. ANALIZA EKSPERIMENTALNIH REZULTATA	131
6.1 KORIŠĆENI RESURSI.....	131
6.1.1 Korišćena baza podataka.....	131
6.1.2 Korišćeni hardver	132
6.1.3 Korišćeni softver	132
6.2 ANALIZA REZULTATA	136
7 ZAKLJUČAK	141
LITERATURA.....	145
SPISAK SLIKA	151
SPISAK TABELA.....	155
DODATAK 1	157
DODATAK 2	161
DODATAK 3	163

Siže

Polazeći od činjenice da na našoj planeti živi preko 7 milijardi ljudi, i da se do 2050 očekuje porast broja stanovnika na oko 9 milijardi, identifikacija osobe uz istovremenu zaštitu privatnosti predstavlja veliki izazov. Sa stanovišta bezbednosti društva, potvrda identiteta zasnovana na biometrijskoj identifikaciji postala je dominantan zahtev od strane mnogih vlada i agencija koje se bave bezbednosnim poslovima. Pored toga, određeni broj privatnih kompanija se takođe uključio u razvoj tehnologije biometrijske identifikacije čime im se otvara mogućnost pristupa visokoprofitabilnom tržištu koje je do nedavno bilo namenjeno isključivo vladinim sektoru. SAD kao ekonomski i tehnološki najrazvijenija zemlja, uprkos ekonomskim problemima u kojima se nalazi, posvećuju posebnu pažnju i ulažu ogromna finansijska sredstva za unapređenje biometrijske identifikacije, naročito posle terorističkih napada koji su se odigrali 11. septembra 2001. godine. Pored SAD, Kina, čija privreda beleži izuzetan rast u vreme svetske ekonomske krize, takođe ulaže velike napore u razvoju biometrijskih mehanizama za identifikaciju pojedinca. Imajući u vidu da je Kina najmnogoljudnija zemlja izazov biometrijske identifikacije dobija novu dimenziju. U Indiji su motivi uvođenja tehnologije biometrijske identifikacije dijametralno različiti od motiva SAD i većine ostalih zemalja i nisu vezana za pitanja bezbednosti već za daleko životnije probleme - raspodelu hrane. Evropska Unija takođe ulaže velike napore u razvoju i primeni biometrijske identifikacije, u čemu naročito prednjači Holandija, koja je ovom problem pristupila sistemski. Privatnost prikupljenih biometrijskih podataka zavisi, pre svega, od stava i politike organizacija koje se bave akvizicijom i čuvanjem biometrijskih podataka, kao i metoda zaštite prikupljenih podataka. Čak i ako pretpostavimo da to čine državne institucije - institucije „od poverenja“ zloupotrebe su, ipak, uvek prisutna opasnost.

Ravnotežu između bezbednosti i privatnosti potražili su mnogi istraživači širom sveta. Poslednjih decenija, izneto je više različitih pristupa ovom problemu. Razvojem računarske tehnologije, dolazi i do razvoja novih pristupa problemu obzirom na otvaranje novih mogućnosti. Potrebno je imati u vidu da su biometrijski podaci - podaci sa „šumom“, što znači da se ne reprezentuju uvek na isti način. Posledica ove činjenice je ta da se pojedini postulati zaključivanja iz klasične logike ne mogu direktno primeniti u ovoj oblasti. Takođe, primena kriptografskih procedura nad biometrijskim podacima nije moguća iz razloga što biometrijski podaci ne zadovoljavaju kriterijume koje zahteva kriptografija. Fazi logika je ponudila novi pristup koji je blizak načinu ljudskog razmišljanja i iskorišćena je u različitim razmatranjima i predloženim rešenjima biometrijske identifikacije. Pojedina rešenja predlažu formiranje kriptografskog ključa neposredno iz biometrijskih podataka, pri čemu se za izdvajanje biometrijskih karakteristika koriste fazi ekstraktorske funkcije. Fazi ekstraktorske funkcije uzimaju u obzir neodređenost merenja biometrijskih podataka. Nakon uvođenja randomizacije i korišćenjem specifičnih procedura, fazi ekstraktorske funkcije kreiraju jedinstveni „ključ“ - fazi skeč. Sa druge strane postoje rešenja koja predlažu nezavisno formiranje kriptografskog ključa, koji zadovoljava sve kriptografske zahteve a zatim njegovo kombinovanje sa biometrijskom veličinom. U okviru ove grupe rešenja predlažu se različite

metode za korekciju greške koje kompenzuju neodređenost biometrijskog podatka. Pojedini autori predlažu korišćenje više uzastopnih metoda za korekciju grešaka. Medijumi za arhiviranje mogu biti različiti dok se polikarbonatna kartica sa mikrokontrolerom predlaže kao uobičajeno rešenje.

Prethodna rešenja se mogu posmatrati iz više uglova pa prema tome mogu se uočiti njihove dobre i loše strane. Osnovni nedostatak rešenja sa fazi skečom je pojavljivanje biometrijske karakteristike u otvorenom obliku prilikom dešifrovanja. Pored toga proces dešifrovanja skeča nije pouzdan u dovoljnoj meri kako bi se koristio u praktičnim primenama. Rešenja koja predlažu nezavisno formiranje kriptografskog ključa a zatim njegovo kombinovanje sa biometrijskom veličinom daju bolje rezultate nego rešenja sa skečom. Postoje i mišljenja da bi se rešenja sa skečom mogla unaprediti poboljšanjem matematičkog modela, mada i u tom slučaju ostaju osnovni nedostaci - vidljivost nezaštićene biometrijske veličine u nekom trenutku procesa. Korišćenje različitih ili sukcesivno kombinovanih metoda za korekciju greške daju manje ili više dobre rezultate. Pojedina rešenja koja tretiraju iris kao biometrijsku veličinu daju sasvim korektne rezultate. Ta rešenja se ne mogu efikasno primeniti za obradu otiska prsta zbog različite prirode binarnih reprezentacija irisa i otiska prsta.

Rešenje predloženo u ovom radu kao biometrijsku veličinu tretira otisak prsta. Nakon uočavanja i izdvajanja minucija - karakterističnih tačaka otiska prsta, kreiraju se dva nezavisna templejta. Prvi templejt se koristi za klasičnu identifikaciju, dok se drugi templejt kombinuje sa unapred kreiranim kriptografskim ključem nad kojim je prethodno primenjeno kodiranje za korekciju greške. U predloženom rešenju radi se o Slepian Vulfom kodiranju. Drugi templejt predstavlja šifrovani biometrijski podatak sa inkorporiranim kriptografskim ključem. Oba templejta se zatim smeštaju na prenosivi token kao što je kartica sa mikroprocesorom. Prilikom rekonstrukcije ključa neophodno je prisustvo originalne biometrijske veličine - otiska prsta. Prvi korak predstavlja klasičnu identifikaciju, kojom se utvrđuje da li se radi o istom otisku prsta, odnosno o istoj osobi. Nakon pozitivne identifikacije, vrši se rekonstrukcija ključa razdvajanjem otiska prsta od kriptografskog ključa i ponovnom primenom Slepian Vulfvog kodiranja kojim se koriguju greške. Dalje se kriptografski ključ koristi u standardnim kriptografskim operacijama.

Predloženo rešenje značajno smanjuje grešku prilikom identifikacije osobe, odnosno rekonstrukcije kriptografskog ključa. Zbog toga je efikasno i pogodno za praktičnu primenu. Istovremeno se poboljšava bezbednost originalnog otiska, koji nije vidljiv u otvorenom obliku ni u jednoj fazi procesa. Rešenje izneto u ovom radu je otporno na široki spektar mogućih napada u cilju kompromitovanja biometrijske veličine a samim tim i identiteta osobe. Ukoliko bi došlo do kompromitovanja bilo biometrijskog bilo kriptografskog ključa ili šifrovanog biometrijskog podatka, moguće je povući prethodni reprezent otiska i kreirati novi korišćenjem novog kriptografskog ključa bez posledica po integritet identiteta osobe.

Ključne reči: Biometrija, Kriptografija, Fazi logika, Kodovi za korekciju greške, Tokeni sa mikroprocesorom

Abstract

Based on the fact that our planet is inhabited by more than 7 billion people, and by 2050 the population growth expected at around 9 billion, the identification of person while protecting privacy is a major challenge. From the perspective of collectively security, authentication based on biometric identification has become a dominant requirement by many governments and agencies involved in security affairs. In addition, a number of private owning companies are also involved in the development of biometric identification technology taking an opportunity to access a highly profitable market that was exclusively dedicated for the government sector. United States as an economically and technologically most developed country, despite the economic problems, pay special attention and invests huge funds to improve biometric identification, especially after the terrorist attacks that took place 11th September 2001. year. In addition to the U.S., China, whose economy recorded impressive growth during the global economic crisis, also making great efforts to develop mechanisms for biometric identification of individuals. Bearing in mind that China is the most populous country challenge of biometric identification gets a new dimension. In India, the motive of introducing biometric identification technology is diametrically different from the motives of the United States and most other countries, and it is not related to security issues, but for a far more vital issues - the distribution of food. The European Union is also making great efforts in the development and application of biometric identification, leads by the Netherlands, who approached the problem systematically. Privacy of collected biometric data depends, above all, of the attitude and policies of organizations dealing with the acquisition and storage of biometric data, and methods for protecting collected data. Even if we assume that they are state institutions - "trusted" institutions, abuse, however, are ever-present danger.

Balance between security and privacy are considered by many researchers around the world. In recent decades, a number of different approaches to this problem is presented. Development of information technology, contributed the development of new approaches to the problem due the new opportunities. It is necessary to bear in mind that the biometric data are data with "noise", which means that they do not always represent at the same manner. The consequence of this fact is that certain principles of reasoning in classical logic can not be applied directly in this field. Also, the usage of cryptographic procedures using biometric data is not possible because the biometric data does not meet the criteria required by cryptography. Fuzzy Logic offers a new approach which is close to the human way of thinking and it is used in a variety of considerations and proposed solutions for biometric identification. Some of the solutions proposed extracting cryptographic keys directly from the biometric data, using fuzzy extractor functions. Fuzzy extractor functions take care about the uncertainty of measurements of biometric data. After introducing the randomization and using specific procedures, fuzzy extractor function creates an unique "key" - a fuzzy sketch. On the other side there are solutions that propose the independent creation of an cryptographic key, which meets all the requirements of cryptographic and then combine it with the biometric data.

Within this group of proposed solutions there are a various error correcting methods are used to compensate the fuzziness of the biometric data. Some authors suggest the usage of sequence of error correcting methods. Media for archiving can be varied while the polycarbonate card with microcontroller is proposed as a common solution.

Previous solutions can be considered from multiple angles and therefore can be distinguished for their good and bad characteristic. The main disadvantage for solutions with the fuzzy sketch is the appearance of biometric features in an open format during decoding process. In addition, the process of fuzzy sketch decryption is not sufficiently reliable to be used in practical applications. The solutions with independent creation of an cryptographic key and its combination with the biometric data gives better results than the solution with the sketch. There are also the opinions that the solutions with the sketch could be enhanced by improving their mathematical models, but in this case the main shortcomings remain - the visibility of unprotected biometric data at some point in the process. The usage of different or successively combined error correcting methods more or less give good results. Some of those solutions that are treated iris as a biometric data gives correct results. These solutions can not be efficiently used for the processing of fingerprints because of the different nature of the binary representation of iris and fingerprint.

The solution proposed in this paper, treats the fingerprint as biometric data. After identifying and distinguishing minutiae - the characteristic points of the fingerprint, the two independent templates are created. The first template is used for identification process, while the other template combines biometrics with pre-built cryptographic key. The cryptographic key is previously processed with the error correction encoding. The proposed solution uses the Slepian Wolf encoding. The second template is encrypted biometric information with incorporated cryptographic key. Both templates are then stored on a token such as a card with a microprocessor. During the reconstruction of the cryptographic key the original biometric - fingerprint must be present. The first step is the identification, which determines whether it is the same fingerprint, or the same person. After positive identification, key reconstruction is processing by separating the cryptographic key from fingerprint and re-applying Slepian Wolf encoding on cryptographic key to correct the errors. Further, the cryptographic key can be used in the standard cryptographic operations.

The proposed solution significantly reduces the errors in person identification, and in the process of cryptographic key reconstruction. Therefore, it is efficient and suitable for practical use. At the same time it improves the security of the original fingerprint image, which is not visible in the open form at any stage of the process. The solution outlined in this paper is resistant to a wide range of possible attacks when attackers want to compromise biometric data and therefore the person identity itself.

Keywords: Biometrics, Cryptography, Fuzzy logic, Error correcting codes, Tokens with microprocessor

1. Uvod

Gde se nalazi granica između bezbednosti i privatnosti? Da li možemo istovremeno biti bezbedni u društvu sa rastućim brojem stanovnika i pri tome sačuvati svoj integritet i slobodu delovanja? Da li je aktivnost koja ugrožava drugu osobu sloboda i da li je sprečavanje takve aktivnosti narušavanje slobode?

Pitanja na ovu temu se nižu u nedogled. Mnogi pokušavaju da odgovore na ova pitanja polazeći sa različitih stanovišta i dajući odgovore koji se međusobno razlikuju. Najveći broj dilema koje postoje u posmatranoj oblasti ostaje bez pravog odgovora.

Povodom terorističkih napada putničkim avionima na SAD, 11. Septembra 2001, Džon MekKonel (*John Michael McConnell*), direktor Nacionalne obaveštajne službe Sjedinjenih Američkih Država, izjavio je sledeće:

„... Internet aktivnosti će morati pažljivo da se prate. To znači davanje ovlašćenja vladi da ispita sadržaj svake elektronske pošte, svakog prenosa podataka i svake aktivnosti Web pretraživača ... privatnosti i bezbednost vode igru bez pobednika“.

Podsetimo se knjige Erika Blera (*Eric Arthur Blair, 1903 – 1950*) poznatijeg pod svojim pseudonimom Džordž Orvel (*George Orwell*), pod nazivom „Hiljadu devetsto osamdeset četvrta“ (*„Nineteen Eighty-Four“*). Da li zaista idemo ka društvu u kome nas „Veliki brat“ neprestano posmatra i kontroliše?

Sa druge strane, mnogobrojne organizacije koje se bore za ljudska prava se oštro suprotstavljaju pokušajima kontrole kakve je u svojoj „Orvelovskoj“ izjavi dao gospodin MekKonel. Jedan od najvećih protivnika ovakve ideje je organizacija „Human Rights Watch“, koja svim civilizovanim sredstvima nastoji da spreči totalitarnu kontrolu i nastanak „policijskih država“.

Bezbednost nasuprot privatnosti je bitka dve međusobno suprotstavljene koncepcije. Postoji mnoštvo argumenata za i protiv svakog od ovih pristupa problemu. Ako se pogleda „Deklaracija o ljudskim pravima“ koja je usvojena u Ujedinjenim Nacijama, u članu 3. stoji: „Svako ima pravo na život, slobodu i ličnu bezbednost“.

Može se uvideti da se čak i u ovom dokumentu sloboda i lična bezbednost stavljaju u istu ravan, bez davanja prednosti bilo kojoj od opcija ali i bez pominjanja dihotomije koja je očigledno, prisutna.

Privatnost nije isto što i anonimnost, što podrazumeva da se svako ljudsko biće mora ponašati odgovorno prema drugim ljudima.

Daljom razradom ovog postulata opisane suprotnosti bi se mogle značajno približiti. Međutim, ta tema je predmet proučavanja nekih drugih oblasti.

Tehnološka rešenja, ako ne mogu da reše, sasvim sigurno mogu da ublaže ovaj sukob. Mnogi istraživači širom sveta pokušavaju da pronađu način na koji bi biometrijski podaci, kao reprezentanti ljudske jedinice bili zaštićeni u dovoljnoj meri da ne zavise od organizacija koje te podatke čuvaju i obrađuju.

Pronalaženjem i primenom novih rešenja u sferi korišćenja biometrijskih podataka može da se izbegne nastajanje totalitarne vlasti uz istovremenu zaštitu privatnosti pojedinca. Traganje za novim rešenjima predstavlja osnovni motiv za izradu ovog rada sa nadom da će postignuti rezultati dati doprinos unapređenju kriptografskih operacija radi postizanja opšteg dobra.

1.1 Definisanje problema

Klasična kriptografija se zasniva na algoritmu i kriptografskom ključu. Kriptografski ključ predstavlja slučajni niz bitova sa ravnomernom raspodelom, koji se može precizno reprodukovati i koji čini tajni podatak u kriptografskim operacijama. Kada je reč o ravnomernoj raspodeli, u kontekstu kriptografskog ključa, ima se u vidu diskretna ravnomerna raspodela. Posmatrajući skup od n elemenata: $k_1, k_2, \dots, k_n$ od kojih se formira slučajan niz, verovatnoća pojavljivanja svakog od navedenih elemenata u formiranom slučajnom nizu je jednaka i iznosi $\frac{1}{n}$. U binarnom sistemu inicijalni skup sadrži dva elementa $S = \{0, 1\}$, dok bi verovatnoća pojavljivanja svakog od ova dva elementa u slučajnom nizu trebalo da iznosi 0,5 ili barem približno 0,5 ukoliko se radi o nizu sa ravnomernom raspodelom. Odnosno, broj jedinica i nula u slučajno formiranom nizu bi trebao biti jednak, ili barem približno jednak.

U kriptografiji se, između ostalog, koriste funkcije koje se primenjuju nad otvorenim tekstom i koje kreiraju posebnu vrstu šifrata iz koje nije moguće ponovo dobiti otvoreni tekst. Zapravo ne postoje matematički dokazi da nije moguće, ali je to sigurno veoma teško. Ili matematičkim rečnikom: postoje matematičke funkcije za koje nije moguće pronaći inverzne funkcije.

Ukoliko se želi da se biometrijski podatak iskoristi kao tajni podatak u kriptografskim operacijama, osnovni izazovi sa kojima se susreće je neravnomerna raspodela binarnog reprezentata biometrijske veličine kao i nemogućnost tačnog reprodukovanja biometrijske veličine prilikom svake akvizicije. Pored toga, nije dobro da se iz šifrata može rekonstruisati sama biometrijska veličina. U tom slučaju bi napad na šifrat odnosno biometrijsku veličinu bio rutinski postupak. Kompromitovanje kriptografskog ključa bi dovelo i do kompromitovanja biometrije, što je nepoželjan ishod.

Moguće je razmotriti primer autentifikacije putem lozinke. Korisnik poseduje lozinku w i želi da pristupi svom nalogu. Server čuva informaciju koja je rezultat funkcije $y = f(w)$ a ne samu lozinku w . Kada korisnik unese lozinku w , server omogućava pristup korisniku samo ukoliko je $f(w) = y$. U ovoj jednostavnoj aplikaciji, pretpostavka je da je bezbedno za korisnika da unese lozinku za verifikaciju. Međutim, dugoročno skladištenje podataka na serveru, pogotovo osetljivih podataka, po pretpostavci nije bezbedno. Cilj je da se realizuje funkcija f kojoj je teško pronaći inverznu funkciju (odnosno, za dato y teško je pronaći w' tako da je $f(w') = y$), na način da niko ne može otkriti korisničku lozinku poznavajući y . Takva funkcija f se naziva jednosmernom funkcijom.

Ukoliko bi se kao lozinka, u prethodnom primeru, koristio biometrijski podatak, primena rešenje bi imala sledeće nedostatke:

1. Po definiciji jednosmerne funkcije pretpostavlja se da je w uniformno distribuirani niz, a ukoliko to nije slučaj ne garantuje se jednosmernost funkcije. Uočimo da biometrijske lozinke nisu ni približno uniformne, iako se mogu do određene mere smatrati nepredvidivim pa nije isključeno da se može dogoditi i slučaj uniformne raspodele,
2. Korisnik mora identično da unese svoju lozinku svakog puta kada se autentifikuje. Treba imati na umu da će dva očitavanja biometrijskih podataka retko biti identična, iako će verovatno rezultati biti bliski.

Iz opisanih razloga, biometrijske veličine se ne mogu direktno koristiti kao ključ u kriptografskim operacijama. Zato je potrebno izvršiti određene transformacije biometrijskog podatka kako bi se mogao koristiti u kriptografiji pri čemu će zadržati osnovne informacije o biometrijskoj veličini.

Entropija se može posmatrati na više načina, a pre svega kao mera neuređenosti sistema. Prema termodinamičkoj definiciji entropije, sistem spontano uvek teži da pređe iz više uređenog u manje uređeno stanje, odnosno iz stanja manje verovatnoće u stanje veće verovatnoće, što znači da je entropija uvek veća od nule. Prilikom razmatranja problema u teoriji informacija, na umu imamo Šenonovu entropiju (*Claude Elwood Shannon, 1916 - 2001*). Entropija u informatičkoj nauci predstavlja prosečan stepen nepredvidivosti slučajne promenljive koje je u korelaciji sa količinom informacija koju ta promenljiva sadrži. O Šenonovoj entropiji biće kasnije reči u tekstu. Očigledno je da biometrijski podaci imaju daleko veću entropiju nego lozinke koje čovek može da zapamti.

Problem uočavanja i ispravljanja grešaka u biometrijskim podacima, pri čemu se zadržava dovoljan nivo pouzdanosti informacija, predstavlja jedno od ključnih razmatranja u ovom radu. Mera tolerancije i određivanje „prihvatljivog“ broja grešaka su takođe veliki izazovi. Ukoliko se dozvoli veći broj grešaka ili njihova veća kvantitativna vrednost time se pouzdanost umanjuje i obrnuto, ukoliko se dozvoljene granice greške ograniče na isuviše niskom nivou, pouzdanost će, prema prethodnom biti visoka, ali će takav sistem biti praktično neupotrebljiv. Potrebno je rešiti problem optimalne vrednosti dozvoljene greške.

Šifrovanje biometrijskih podataka se može ostvariti na više različitih načina. Neki od postulata koji se koriste u klasičnim šifarskim sistemima nisu primenljivi na biometrijske podatke. Razmatranja zaštite biometrijskih podataka primenom fazi tehnologija i kodova za korekciju greške nude nova rešenja čije ostvarenje ne bi imalo samo tehnološke već i šire civilizacijske konsekvence.

1.2 Ciljevi istraživanja

Opšti cilj istraživanja je unapređenje mogućnosti korišćenja biometrijskih karakteristika u kriptografskim operacijama. Pored toga, postoji potreba podizanja nivoa primenljivosti biometrijskih podataka - otiska prsta u kriptografskim operacijama. Postavljeni opšti ciljevi podrazumevaju rešavanje mnogobrojnih problema koji su vezani za definisanje načina, procedura i postupaka kojima je moguće transformisati biometrijske podatke u oblik pogodan za spregu sa kriptografijom. U tom kontekstu se ciljevi proširuju na povećanje efikasnosti procesa prepoznavanja odnosno identifikacije osobe na osnovu otiska prsta kao biometrijske veličine. Jedan od opštih ciljeva je i povećanje stepena bezbednosti biometrijske karakteristike - otiska prsta, čime se pruža doprinos zaštiti privatnosti ličnosti i smanjenju mogućnosti krađe identiteta koja postaje sve rasprostranjeniji vid kriminala. Posmatrajući cilj istraživanja iz ugla društvenih potreba, njegovim ostvarenjem bi se postigla potrebna ravnoteža između slobode i privatnosti ličnosti. Postizanje pomenutog balansa i pronalaženje sveobuhvatnog rešenja može se smatrati izazovom savremene kriptologije. Ovaj rad predstavlja pokušaj iskoraka ka otklanjanju fundamentalnih kontradiktornosti između slobode sa jedne i privatnosti sa druge strane.

Naučni cilj istraživanja bi se najbolje mogao opisati kao potreba za pronalaženjem originalnog načina obrade i korišćenja biometrijske veličine u kriptografskim operacijama. Mnogi istraživači su pokušavali, i dalje pokušavaju, da pronađu najbolje moguće rešenje za identifikaciju osobe korišćenjem biometrijske karakteristike, u ovom razmatranju otiska prsta, sa najmanjom mogućom greškom. Potrebno je smanjiti stepen greške prihvatanja identiteta osobe kada se ne radi o istoj osobi, kao i stepen greške odbacivanja identiteta osobe ukoliko se radi o istoj osobi. Da bi naučni cilj bio dostignut korišćeni su složeni matematički metodi i modeli kao glavne tačke oslonca. Transformacija biometrijskih podataka i njihova upotreba u sprezi sa kriptografijom je veoma izazovan cilj, zasnovan na različitim matematičkim metodama. Postoje mnoga rešenja, dok je u ovom radu razmatrano pronalaženje novih matematičkih modela. Pri tome su uzeta u obzir prethodna istraživanja i modeli koje su mnogi istraživači spoznali i objavili stručnoj javnosti, putem različitih međunarodni časopisa ili pojedinačnih članaka na mnogobrojnim konferencijama. Osnovna želja je da se uoče i unaprede teorijske postavke koje bi dovele do spoznaje načina unapređenja opisanih procesa.

Praktični cilj istraživanja koje će detaljno biti opisano u ovom radu predstavlja iznalaženje praktičnih i u tehnološki upotrebljivih rešenja. Pri tome se podrazumeva izrada originalnih algoritama zasnovanih na rezultatima postignutim prilikom ostvarivanja opštih i naučnih

ciljeva. Praktični cilj je i razvijanje procedura, na osnovu izrađenih algoritama koje se dalje mogu koristiti u procesu kodiranja softvera ili primene u nekom od mnogobrojnih softverskih alata. Izrada softvera, odnosno pojedinačnih komponenti ili modula softvera koji se mogu koristiti kao delovi sveobuhvatnijih softverskih rešenja. Softverske komponente su predviđene za integraciju i korišćenje kako u različitim savremenim programskim jezicima, tako i u modernim softverskim alatima koji omogućavaju praktičnu primenu rešenja kao i integraciju u širem kontekstu. Osnovni praktični cilje je upotrebljivo i primenjivo rešenje koje se može koristiti u praktičnim implementacijama a na osnovu rezultata koji slede iz ostvarivanja opšteg i naučnog cilja.

1.3 Hipoteze

Hipoteze na kojima se zasniva razrada teme u ovom radu su:

1. Biometrijski podaci su jedinstveni i predstavljaju dobru polaznu osnovu za formiranje jedinstvenog kriptografskog ključa.

Genetsko ukrštanje objašnjavaju Mendelovi zakoni nasleđivanja, (Gregor Johann Mendel, 1822 - 1884):

Partikularno nasleđivanje podrazumeva da su osobine određene naslednim faktorima koji se prilikom prenošenja sa roditelja na potomstvo ne mešaju, već se pri nastanku gameta najpre razdvajaju, a zatim nezavisno kombinuju u procesu oplodjenja. Nezavisno kombinovanje genetskog materijala obezbeđuje različitost svake jedinke i jedinstvenost biometrijskih podataka.

Mendel je eksperimentisao sa graškom. Odgajao je čiste sorte biljaka: jednu koja u svakoj generaciji daje okruglo seme i drugu koja daje naborano seme. Biljke iz takve dve sorte je ukrštao međusobno i nazivao ih roditeljskom ili parentalnom generacijom. U potomstvu je dobio sve biljke sa okruglim zrnom i tu generaciju je nazvao prva filijalna generacija ili F_1 , od latinskog *filius* - sin, dete. Međusobnim ukrštanjem biljaka F_1 generacije dobio je narednu, F_2 generaciju u kojoj se javljaju biljke i sa okruglim i sa naboranim semenom u brojnom odnosu 3 : 1 (tri puta ima više biljaka sa okruglim nego sa naboranim semenom).

Na osnovu opisanog eksperimenta, Mendel je zaključio da osobine kontroliše određeni nasledni faktor. Simbolom velikog slova A obeležio je faktor koji određuje okruglo zrno, a malim slovom a onaj koji izaziva naborano zrno. Kada se pomenuta dva faktora nađu zajedno (Aa) u jednoj biljci tada se ispoljava samo dejstvo faktora A, pa ga je Mendel nazvao dominantnim faktorom. Drugi faktor (a) koji nije došao do izražaja u F_1 generaciji označava se kao recesivan. Pri nastanku F_2 generacije faktori se rastavljaju jedan od drugog, prilikom obrazovanju gameta, a zatim se slobodno kombinuju sa faktorima iz druge jedinke, odnosno njenim gametima. Gameti su polne ćelije koje se stvaraju putem mejoze te sadrže haploidan broj hromozoma (postoji samo jedan primerak svakog od hromozoma). Mejoza je

ćelijska deoba u kojoj se broj hromozoma u novonastalim ćelijama redukuje na polovinu u odnosu na matične ćelije. U procesu mejoze se od diploidnih (postoje dva primerka svakog od hromozoma), matičnih ćelija spermatogonija i oogonija stvaraju polne ćelije, tj. muške i ženske gamete koje sadrže haploidan broj hromozoma, odnosno za polovinu manje od somatskih. Tako se u F_2 generaciji mogu dobiti samo četiri kombinacije ovih faktora : AA, Aa, aA i aa. Prve tri kombinacije, zbog prisustva dominantnog faktora daju okruglo seme, a samo jedna kombinacija (aa) daje naborano seme čime je odnos 3 : 1 potvrđen.

2. Biometrijski podaci u otvorenom obliku su osetljivi. Njihovo kompromitovanje je nenadoknadivo te ih je iz tih razloga neophodno štititi.

Identitet nije moguće ukrasti, već je moguće lažno se predstavljati, odnosno moguće je uveriti sagovornika da komunicira sa osobom čije karakteristike simuliramo. U realnom svetu to nije tako jednostavan proces, dok u digitalnom svetu postoje mnogi načini za krađu identiteta. Žrtve krađe identiteta najčešće nisu ni svesne da se neko drugi predstavlja u njihovo ime i da obavlja radnje u njihovo ime koje su najčešće krivična dela koja nanose štetu osobi čiji je identitet ukraden i protivpravnu imovinsku ili moralnu korist osobi koja je identitet ukrala. Krađa identiteta je pojava u kojoj pojedinac koristi tuđe lične informacije za identifikaciju, kao što su ime, broj socijalnog osiguranja, ili broj kreditne kartice, bez dozvole vlasnika, sa namerom da poćini prevaru ili drugo krivićno delo. Prema podacima amerićke Federalne Komisije za Trgovinu (*Federal Trade Commission - FTC*) procenjuje se da se u SAD godišnje ukrade identitet za ćak 9 miliona ljudi. Kradljivci identiteta mogu dobiti broj socijalnog osiguranja, kreditnu karticu ili neki drugi lićni dokument. U najvećem broju slućaja poćinioci krivićnog dela krađe identiteta ostaju nepoznati. Krađa identiteta je ozbiljan problem koji moće dovesti do hapšenja nedućne osobe za zloćine koje je neko drugi poćinio.

Svaki podatak koji se moće zloupotrebiti neophodno je štititi: podaci vezani za platnu karticu, parametri za pristup odrećdenim lokacijama ili raćunarskim resursima itd. Za krađu identiteta putem Interneta, hakeri su razvili specijalizovane alate, tako da ćak i relativno neiskusani korisnik moće jednostavno doći do broja nećije platne kartice kao i ostalih podataka koji mogu dovesti do zloupotrebe, odnosno plaćanja dobara tućdom platnom karticom, odnosno tućdim novcem.

Biometrijski podaci predstavljaju posebno osetljivu grupu podataka iz jednostavnog razloga što se kompromitovani biometrijski podaci ne mogu nadoknaditi. Rizik od kompromitovanja podataka zavisi od institucije ili organizacije koja ćuva biometrijske podatke. Treba imati na umu da otisci prstiju, kada se nalaze u bazi podataka, najćešće predstavljaju samo binarne stringove - nizove nula i jedinica. U najgorem slućaju, neko kome je dostupna baza biometrijskih podataka moće pridrućiti svoj otisak prsta drugoj osobi, poćiniti krivićno delo, a zatim optućiti tu osobu za poćinjeno krivićno delo. Izgubljen ili ukraden lićni dokument se moće zameniti novim. Izgubljena ili ukradena platna kartica se moće zameniti novom. Kompromitovanje biometrijskih podataka je mnogo opasnije nego kompromitovanje PIN-a (*Personal identification number*), broja kreditne kartice ili broja kartice socijalnog

osiguranja. Otisak prsta nam je dat na rođenju i u slučaju kompromitovanja, ne može se zameniti novim!

3. Prilikom akvizicije biometrijskih podataka, isti podatak se ne reprezentuje uvek na isti način.

Merenje predstavlja osnovu naučnog istraživanja. Rezultati merenja su, zapravo, približne vrednosti a ne prave vrednosti, koje zavise od: opsega merenja mernog uređaja, uslova merenja, načina merenja kao i ljudske greške. Težnja je da se greška merenja svede na najmanju moguću meru.

Greške se mogu podeliti u dve osnovne kategorije:

- Sistematske greške
- Slučajne greške

Sistematske greške nastaju usled nekorektne procedure merenja. Greške iz ove kategorije karakteriše odstupanje rezultata uvek na isti način u odnosu na pravu vrednosti. Sistematska greška može nastati usled:

1. neispravnog instrumenta
2. nekorektnog procesa merenja
3. prisutne subjektivnosti prilikom merenja

Slučajna greška za razliku systemske greške ne daje uvek odstupanja na isti način. Neke od izmerenih vrednosti su veće od prave vrednosti dok su neke manje od prave vrednosti. Smanjenje slučajne greške je moguće smanjiti višestrukim ponavljanjem merenja kao i promenom statističkih metoda za izračunavanje približne prave vrednosti.

U prirodi čoveka je da ne može ponoviti određenu akciju na potpuno isti način. Stoga je vrlo verovatno da ista osoba očitava različite rezultate prilikom svakog merenja istim instrumentom koji meri ispravno.

Iz prethodno rečenog se može zaključiti da je svako merenje podložno grešci.

Merenje biometrijskih podataka ima svoje specifičnosti i posebno je osetljivo jer zavisi od više činilaca. Pored navedenih standardnih grešaka merenja, merenje biometrijskih podataka postavlja dodatne izazove. Skeniranje otiska prsta, između ostalog zavisi od pritiska koji se vrši na skener, čime se dobija manje ili veće izobličenje slike. Treba imati u vidu činjenicu da je ljudski prst sastavljen od više kostiju i mekog tkiva obavijenog kožom. Očigledno je da prislanjanje opisane strukture prsta na ravnu podlogu stvara izobličenje realnog oblika. Trodimenziono skeniranje lica može biti različito u zavisnosti od raspoloženja osobe, jer se mišići lica grče ili opuštaju na specifičan način, čime se dobijaju delimično različite koordinate mernih tačaka itd. Iz navedenih razloga je potrebno definisati dozvoljena odstupanja, o čemu će detaljnije biti reči u daljem tekstu ovog rada.

4. Biometrijski podaci se mogu efikasno štititi tako što se superponiraju sa binarnim nizom - kriptografskim ključem.

U dosadašnjoj praksi su razvijene metode obrade biometrijskog podatka koje omogućavaju akviziciju biometrijskog podatka, obradu slike reprezenta biometrijskih podatka i njeno prevođenje u binarni niz. Po svojoj prirodi, binarni niz dobijen na opisani način ne zadovoljava kriterijume koje je potrebno da zadovoljava kriptografski ključ koji se pored kriptografskog algoritma koristi kao element u kriptografskim operacijama. Binarni niz dobijen obradom biometrijske veličine ne poseduje jednu od osnovnih osobina, a to je uniformnost raspodele. Ili pojednostavljeno rečeno, broj nula i jedinica u posmatranom binarnom nizu najčešće nije ni približno jednak, niti su nule i jedinice ravnomerno raspoređene. Kriptografski ključ koji je generisan prema kriptografskim pravilima se generiše tako da pored ostalih zahteva, poseduje i osobinu uniformnosti raspodele.

5. Zaštićene biometrijske podatke moguće je efikasno čuvati i obrađivati na različitim nosiocima podataka kao što su kartice sa mikrokontrolerom - čip kartice, hibridne kartice sa čipom i optičkom memorijom ili tokeni sa mikrokontrolerom.

U praksi je pokazano je da je moguća primena kartice sa mikrokontrolerom - čip kartice, hibridne kartice sa čipom i optičkom memorijom ili tokena sa mikrokontrolerom, kao nosioca biometrijskih podataka. Optička memorijska kartica sa čipom predstavlja odličan medijum, koji svojom memorijom daleko prevazilazi mogućnosti standardnih čip kartica. Ugrađeni čip na kartici omogućava izvršavanje kriptografskih funkcija u cilju zaštite biometrijskih podataka koji se nalaze na optičkoj memoriji kartice. Ulazni podaci se mogu obrađivati u memoriji čipa, koristeći specifične algoritam za kreiranje šifrovane vrednosti biometrijskog podatka. Algoritam kao i pripadajući kriptografski ključevi se čuvaju u memoriji čipa. Ukoliko napadač uspe da dešifruje sigurnosni kod, jednostavno se kreira novi na osnovu istih biometrijskih podataka i implementira se na novoj ili istoj kartici. Na opisani način se sprečava kompromitovanje samih biometrijskih podataka.

Analogni postupci se mogu primeniti i korišćenjem savremenih USB tokena koji pored dovoljno memorijskog prostora poseduju i zaštićeni deo radne memorije u kome se odvijaju željene kriptografske operacije i matematička izračunavanja.

Bezbednost, mogućnost procesiranja podataka u zaštićenom okruženju i redundantnost podataka su glavne komparativne prednosti kartica sa optičkom memorijom i mikrokontrolerom sa praktičnom primenom u cilju identifikacije, provere identiteta i zaštite biometrijskih podataka.

1.4 Metode istraživanja

U toku rada na disertaciji korišćena je komparativna analiza dosadašnjih znanja iz oblasti obrade i zaštite biometrijskih podataka kao i metode šifrovanja biometrijskih podataka sa ciljem dobijanja ponovljivog uniformnog niza. Proces je posmatran od trenutka akvizicije pa sve do kreiranja binarnog niza koji je moguće koristiti na više različitih načina. Binarni niz u razmatranjima koja slede će se koristiti pre svega kao kriptografski ključ. U procesu rada na disertaciji korišćene su metode zasnovane na analizi dobijenih rezultata i sintezi elemenata i zaključaka dobijenih primenom različitih matematičkih modela.

Proučavana je literatura koja obrađuje rezultate koji su do sada postignuti u ovoj oblasti. Literatura je najvećim delom prikupljena putem Interneta sa više različitih lokacija i u više različitih oblika. U najvećoj meri je korišćena digitalna biblioteka asocijacije IEEE (*Institute of Electrical and Electronics Engineers*) ali i mnogi drugi raspoloživi elektronski i pisani materijali. U obzir su uzeta fundamentalna istraživanja iz ove oblasti kao i najnovija dostignuća koja se baziraju na osnovnim postavkama transformacije biometrijskih podataka u binarni niz kao i njihova daljnja obrada.

1.4.1 Testiranje matematičkih modela

Razmatrani matematički model je testiran na realnoj bazi biometrijskih podataka dobijenoj od Kineske Nacionalne Laboratorije za Prepoznavanje Oblika (*National Laboratory of Pattern Recognition - NLPR*) kao i Institut za Automatizaciju, Kineske Akademije Nauka (*Institute of Automation, Chinese Academy of Sciences - CASIA*). Baza podataka sadrži sledeće:

- Otisak prsta: 20000 otisaka uzetih od 500 osoba,
- Iris: 54607 skeniranih irisa, uzetih od više od 1800 osoba i 1000 računarski kreiranih osoba za potrebe testiranja,
- Lik - 2D: 2500 dvodimenzionih fotografija lika, uzetih od 500 različitih osoba
- Lik - 3D: 4264 trodimenziono skenirana lika, uzeta od 123 različite osobe
- Dlan - monospektralno skeniranje: 5502 skenirana dlana, uzeta od 312 različitih osoba
- Dlan - multispektralno skeniranje: 7200 skeniranih dlanova, uzetih od 100 različitih osoba
- Rukopis: uzet od 188 osoba sadrži 1074 pisana teksta pri čemu je svako od njih napisao po 50 kineskih odnosno engleski reči u dve sesije.

Iz opisane biometrijske baze, u ovom radu je korišćena baza podataka otiska prsta na kojoj su testirani predloženi matematički modeli za generisanje binarnih nizova iz biometrijskih podataka. Pored toga, ista baza podataka je korišćena i za testiranje predloženog modela sa dva templejta radi komparativne analiza dobijenih rezultata.

1.4.2. Razvoj softvera

Za potrebe testiranja modela razvijen je originalni softverski alat programiran za Windows/Linux okruženje. Alat je programiran u alatu Visual Studio 2010 ASP.NET dok je za programiranje korišćen C# programski jezik. Za korišćenje alata na Linux operativnom sistemu, neophodna je dodatna instalacija MONO softverske platforme na Linux-u. Za potrebe testiranja je korišćen skener za skeniranje otiska prsta DigitalPersona „U.are.U 400B“ kao i odgovarajući drajveri i API (Application Programming Interface) funkcije za pristup hardverskim elementima skenera.



Slika 1. Skener otiska prsta korišćen u istraživanju

Drajvere i API funkcije obezbeđuje sam proizvođač. Dalja logika upoređivanja prepoznavanja istovetnih otisaka kao i izdvajanje binarnog niza predstavlja originalno razvijeni kod. U C# programskom jeziku razvijene su klase koje svoju primenu mogu pronaći i u rešavanju drugih, sličnih problema. Prilikom razvoja koda klasa primenjeni su matematičke metode koje su opisane u teorijskom delu. Prvenstveno se misli na kodove za korekciju grešaka, a zatim na izdvajanje jedinstvenog niza u cilju generisanja kriptografskog ključa. Nakon generisanja binarnog niza moguće je vršiti njegovo upoređivanja sa originalnim binarnim nizom u cilju identifikacije otiska prsta.

1.5 Očekivani rezultati istraživanja i naučni doprinos

Istraživanje opisano u ovom radu ima za cilj da potvrdi iznete hipoteze. Potrebno je pokazati da se biometrijski podaci mogu kvalitetno transformisati u binarni niz i da se na njih uspešno mogu primeniti kodovi za korekciju greške. Očekuje se da istraživanje pokaže i mogućnost praktične primene dobijenih teorijskih rezultata u smislu korišćenja biometrijskih podataka kao kriptografskih ključeva u različitim kriptografskim operacijama.

U ovoj disertaciji očekivani su sledeći doprinosi:

- **Kritička analiza** dosadašnjih istraživanja u oblasti transformacije biometrijskih podataka u binarni niz i dostignuća u oblasti primene kodova za korekciju greške na binarnim nizovima dobijenim na osnovu biometrijskih podataka. Analizom će biti obuhvaćeni i postojeći softverski alati za zaštitu biometrijskih podataka;
- **Eksperimentalna potvrda** transformacije biometrijskih podataka i njihova primena u kriptografskim operacijama uključujući korišćenje standardnih metodologija i alata za softversko modelovanje radi proširivanja funkcionalnosti postojećih aplikacija za obradu biometrijskih podataka;
- **Projektovanje i razvoj softverskog rešenja** koje omogućava testiranje i analizu zadatih matematičkih modela;
- **Kreiranje sopstvenih klasa** u C# programskom jeziku u svrhu transformacije i zaštite biometrijskih podataka;
- **Sinteza rezultata** i predlog smernica za razvoj i primenu biometrijskih podataka u kriptografskim operacijama.

Tokom istraživanja na ovoj disertaciji došlo se do rezultata koji su delimično publikovani kako u domaćim časopisima i na domaćim konferencijama, tako i na međunarodnim konferencijama i u međunarodnim časopisima kategorisanih na SCI listi.

Teorijski i praktični rezultati rada iz ove doktorske teze treba da doprinesu rešavanju problema korišćenja neuniformnih tajnih podataka sa šumom (podataka koji se ne mogu uvek identično reprodukovati) - biometrijskih podataka za kriptografske primene kao i za primene autentifikacije.

1.6 Pregled sadržaja po poglavljima

Teza je organizovana na takav način da su teme obrađivane po poglavljima u skladu sa tematikom i redosledom izlaganja.

U uvodnom poglavlju 1. daje se kratak uvod u oblast kao i opis osnovnih pojmova i problema koji se istražuje, kao i prikaz mogućih rešenja. Takođe je opisana i motivacija za rad na disertaciji.

U poglavlju 2. se detaljno opisuje predmet istraživanja i to u vidu dve celine u kojima su predstavljeni elementi biometrije i elementi kriptografije. Pomenute dve oblasti čine polazne osnove rada na kojima će biti zasnovana dalja razmatranja. Razumevanjem osnovnih postavki biometrije i kriptografije stižu se uslovi za analiziranje postojećih modela šifrovanja biometrijskih podataka.

U poglavlju 3. su date osnove fazi logike kao originalnog načina rešavanja problema. Fazi logika predstavlja interpretaciju problema i način njegovog rešavanja na način sličan načinu

na koji se rešavaju problemi u ljudskom umu. U ovom poglavlju je prikazan i originalan pristup transformaciji biometrijskih podataka korišćenjem fazi logike.

U poglavlju 4. su dati elementi kodova za korekciju greške. Razmatrani su Rid-Solomonovi kodovi, Adamarovi kodovi kao i Slepian Vulfovi kodovi. Posebna pažnja je posvećena Slepian Vulfovom kodiranju koji predstavlja bitan deo predloženog rešenja.

U poglavlju 5. je opisano kreiranje templejta otiska prsta čiji je jedan od autor Anil Jain. Date su analize izdvajanja svojstava otiska prsta i načini upoređivanje otisaka. Nakon toga je opisana i metoda Deloneove triangulacije. Detaljno je opisano predloženo rešenje i to: generisanje kriptografskog ključa i obrada otiska prsta. Kao deo predloženog rešenja posebno je obrađeno kombinovanje kriptografskog ključa sa templejtom otiska kao i koncept dva templejta.

Poglavlje 6. predstavlja analizu dobijenih eksperimentalnih rezultata. Detaljno su opisani: korišćena baza podataka, opis eksperimenta kao i korišćeni softverski alati. Opisani su dobijeni rezultati i data je njihova analiza.

Poglavlje 7. je zaključak u kome su data završna razmatranja. Ukazuje se na doprinose rada kao i smernice za budući rad u oblasti obrađenoj u disertaciji.

2. Predmet istraživanja

U radu će biti istraživana transformacija biometrijskih podataka i njihova primena u kriptografskim operacijama posmatrano iz ugla nekoliko naučnih oblasti. Kompleksnost postavljenih ciljeva zahteva proučavanje i analizu različitih materijala koji, iako obrađuju različite teme i na kraju definišu rešenja koja se predlažu u ovoj disertaciji. U radu će, u većoj ili manjoj meri, biti razmatrano više naučnih oblasti kao što su:

- Biotehnološke nauke,
- Formalne nauke (matematika, logika, statistika),
- Prirodne nauke (biofizika, biologija, fizika),
- Informacione i komunikacione tehnologije,
- Tehničke nauke (elektronika, elektrotehnika),
- Nauke bezbednosti i odbrane.

U ovom poglavlju će posebno biti obrađeni elementi biometrije kao i elementi kriptografije.

2.1 Elementi biometrije

U svakom aspektu svog bića, ljudi pokazuju kombinaciju osobina i to: fizičkih, emocionalnih i bihejvioralnih, koje izdvajaju jedinku od svih ostalih na planeti. Jedinstveni kvalitet svakog ljudskog života zaslužuje poštovanje, dostojanstvo i jednakost svakog pojedinca. „Svi ljudi stvoreni jednaki...“ [1] rekao je Tomas Džeferson (*Thomas Jefferson, 1743 - 1826*), treći predsednik Sjedinjenih Američkih Država i tvorac „Deklaracije o nezavisnosti“ donetoj 4. jula 1776. Koreni pomenutog iskaza potiču iz Svetog Pisma: „I stvori Bog čoveka, po liku Božjem stvori ga, muško i žensko stvori ih“ (I knjiga Mojsijeva, I poglavlje, stih 27). Ovakav način razmišljanja treba posmatrati pre kao filofske misli iskazane u simbolima nego kao doslovnu realnost.

Sa druge strane, uzimajući u obzir Darvinovu teoriju evolucije (*Charles Robert Darwin, 1809 - 1882*), ljudska vrsta predstavlja produkt evolucije živog sveta na planeti Zemlji. Za potrebe proučavanja biometrijske različitosti Darvinova teorija predstavlja polaznu osnovu.

Reč biometrija je akronim grčkog porekla može doslovno prevesti kao „mera života“. Pri pominjanju reči „biometrija“ misli se na procese merenja pojedinih fizičkih, fizioloških i bihejvioralnih karakteristika čoveka. Dalje se izmerene karakteristike mogu upoređivati i na

onovu toga odlučivati da li se radi o karakteristikama iste ili različite jedinke. Pomenuti proces se može posmatrati i kao proces biometrijske identifikacije.

Izbor određene biometrijske karakteristike se vrši na osnovu nekoliko faktora:

Univerzalnost - pretpostavlja da svaka osoba poseduje izabranu osobinu.

Jedinstvenost - pretpostavlja da izabrana osobina treba da bude dovoljno drugačiji za različite jedinke tako da se jedinke iste vrste mogu razlikovati posmatranjem i upoređivanjem izabrane biometrijske veličine.

Merljivost - podrazumeva mogućnost merenja posmatrane biometrijske veličine nekim od razvijenih uređaja ili metoda.

Prihvatljivost - se odnosi na činjenicu kako pojedinci, u okviru svoje kulturne zajednice, prihvataju činjenicu da se njihove biometrijske karakteristike mere i čuvaju u odgovarajućim institucijama.

Pored navedenog je potrebno da se iz podataka mogu ekstrahovati odgovarajuće karakteristika koje reprezentuju sam podatka u obliku koji omogućava naknadnu obradu.

Iz navedenog je jasno da se biometrijska identifikacija oslanja na merljive fiziološke, odnosno antropološke veličine ili na obrasce ponašanja pojedinca.

Fiziološke biometrijske osobine su: otisak prsta, geometrija dlana, iris, retina, geometrija lica. Bihevioralne osobine odražavaju ponašanje pojedinaca u vezi obavljanja određenih radnji kao što su: glas, potpis, hod, način pritiska tastera na tastaturi i itd.

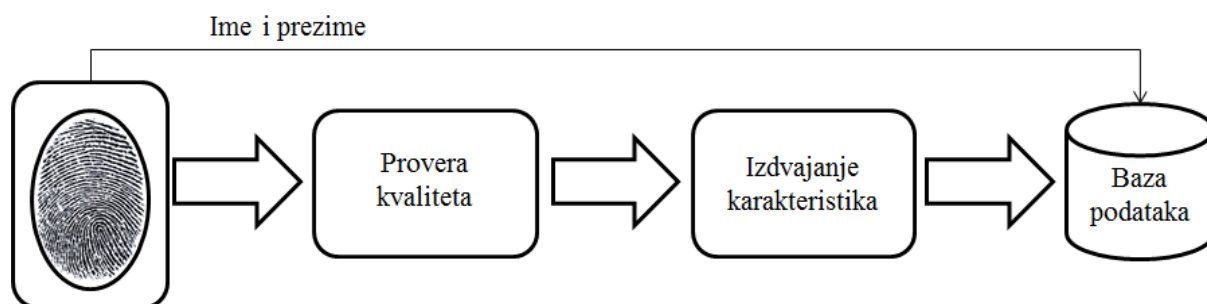
2.1.1 Kratak uvod u biometriju

Pitanje identiteta postaje sve značajnije a naročiti značaj dobija u „digitalnom svetu“ kada ne postoji lični kontakt sa osobom sa kojom se komunicira. To je pitanje koje se sve više zaokuplja pažnju: pojedinaca, vlada, bezbednosnih organizacija i poslovnih organizacija. Privredne organizacije nastoje da se bore protiv prevare, krađe intelektualnog vlasništva, neovlašćenog pristupa poslovnim informatičkim resursima. Na taj način se štiti položaj kompanije na tržištu. Vlada i bezbednosne organizacije se bore protiv organizovanog kriminala, terorizma, kao i za borbu protiv drugih ilegalnih aktivnosti. Biometrija je široko prihvaćena tehnologija koja koristi napredne računarske tehnike i koja stoji na prvoj liniji mera bezbednosti i za utvrđivanje identiteta, otkrivanje kriminala, a takođe obezbeđuje i efikasno odvracanje kriminalnih aktivnosti. Biometrija obuhvata niz tehnika merenja fizioloških i bihevioralnih veličina [2].

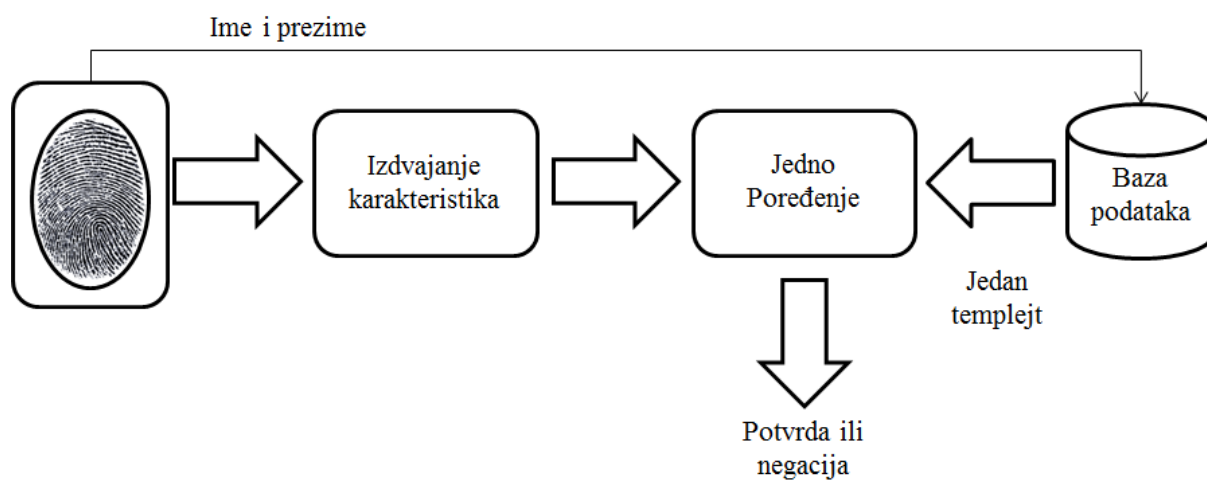
Biometrijski sistemi se svode na sisteme za prepoznavanje oblika ili sisteme za prepoznavanje obrazaca. Postupci u biometrijskim sistemima se mogu svesti na tri osnovne procedure:

- Čitanje biometrijskih podataka i upisivanje u bazu podataka
- Verifikacija
- Identifikacija

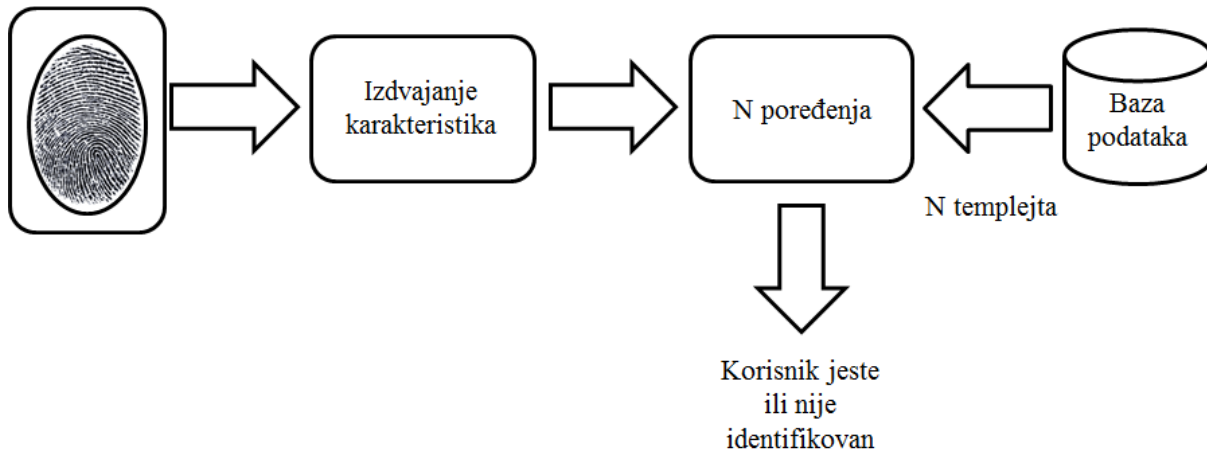
Šematski prikaz pomenutih procedura je prikazan na sledećim slikama:



Slika 2. Šematski prikaz čitanja biometrijskih podataka i njihovo upisivanje u bazu podataka



Slika 3. Šematski prikaz verifikacije identiteta korisnika



Slika 4. Šematski prikaz identifikacije korisnika

Postupak učitavanja biometrijski podataka u bazu podataka započinje skeniranjem biometrijske veličine specijalizovanim skenerom. U posmatranom slučaju razmatramo otisak prsta pošto će to biti podatak kojim ćemo se primarno baviti u ovom radu. Nakon skeniranja dobija se slika otiska prsta koja se dalje obrađuje. Daljom obradom izdvajaju se karakteristične tačke - minucije, o kojima će kasnije biti više reči. Nakon toga se podaci o minucijama upisuju u bazu podataka u predefinisanoj formatu. Istovremeno se u bazu podataka upisuje i neki od identifikacionih podataka, recimo ime i prezime. Podatak se pridružuje reprezentu biometrijske veličine.

U režimu provere, sistem potvrđuje identitet osobe upoređivanjem biometrijskih podataka dobijenih od prisutne biometrijske veličine sa podacima biometrijskog templejta koji se nalazi u bazi podataka sistema [3]. Osoba se predstavlja svojim imenom i prezimenom pri čemu prezentuje svoj biometrijski podatak.

U takvom sistemu, za osobu koja želi da potvrdi svoj identitet sprovodi se poređenje po sistemu poređanja jednog templejta iz baze podataka koji je izdvojen na osnovu imena i preizmena sa templejtom dobijenim od prisutne biometrijske veličine. Ukoliko se dva templejta podudaraju kao rezultat se dobija potvrda identiteta i obrnuto, ukoliko se dva templejta ne podudaraju, rezultat je negacija identiteta osobe.

Na opisani način se sprečava da više ljudi koristi isti identitet.

U režimu identifikacije, sistem prepoznaje pojedinca pretraživanjem templejta svih korisnika u bazu podataka za upoređivanje. Sistem vrši poređenje templejta prezentovane biometrijske veličine sa svakim od templejta koji se nalazi u bazi podataka. Ukoliko pronađe templejt koji se podudara sa posmatranim templejtom, sistem čita podatke o osobi i proglašava osobu identifikovanom. U suprotnom, ukoliko sistem ne pronađe templejt koji se poklapa, sistem proglašava identifikaciju neuspešnom. To znači da se templejt biometrijske veličine posmatrane osobe ne nalazi u datoj bazi podataka i da osobu nije moguće identifikovati. Na opisani način se sprečava da jedna osoba koristi više identiteta.

Biometrijske tehnologije se mogu posmatrati i kao automatizovane metode za prepoznavanje osobe na osnovu biološke i bihejvioralne karakteristike. Biometrijski tehnologija podrazumeva akviziciju i skladištenje biometrijskih veličina pojedinca. Templejti kao reprezentanti biometrijskih karakteristika se arhiviraju u obliku koji je merljiv i na osnovu koga se naknadno može utvrditi identitet osobe kako je to opisano u prethodnom tekstu. Pri tome se podrazumeva korišćenje informatičke opreme kako bi se prepoznavanje izvršili na automatizovan način, u realnom vremenu [4].

Biometrijski sistem se u suštini svodi na prepoznavanje oblika, što je jedna od definicija veštačke inteligencije. Oblici se u biometriji prezentuju u obliku templejta (obrazaca ili šablona). Templejti se u krajnjoj instanci predstavljaju u vidu binarnog niza nula i jedinica. Binarni niz se dobija iz početne slike biometrijske veličine korišćenjem specijalizovanih matematičkih algoritama. Poređenje templejta se zapravo svodi na upoređivanje binarnih nizova. Treba imati u vidu da kod biometrijskih podataka nije moguće izvršiti reverzni inženjering, tj. iz binarnog niza nije moguće rekonstruisati biometrijsku karakteristiku.

Biometrijski merljiva karakteristika može biti anatomska, što podrazumeva: oči, lice, prste, ruke ili glas. Pored toga, kao biometrijski merljiva karakteristika može se posmatrati i način ponašanja kao što su: način svojeručnog potpisivanja, ritam kucanja po tastaturi, način držanja tela, način hoda itd. Automatski sistemi prepoznavanja se najčešće baziraju na kombinovanju različitih biometrijskih karakteristika. Pri tome se može uočiti intencija da se izbegava intervencija čoveka, odnosno da se sve potrebne procedure izvršavaju automatski.

Proces identifikacije odnosno utvrđivanja identiteta se može predstaviti u tri nivoa:

- Na osnovnom nivou se vrši identifikacija na osnovu nečega što osoba poseduje. To može biti kartica sa fotografijom ili neki drugi identifikacioni dokument.
- Na sledećem nivou se osoba predstavlja nekim podatkom koji zna. Prilikom plaćanja platnom karticom radi se o PIN vrednosti, ili prilikom prijavljivanja na informacioni sistem podatak koji osoba zna je korisničko ime i lozinka.
- Na najvišem nivou identifikacije se utvrđuje identitet osobe merenjem njenih biometrijskih veličina, bilo anatomske bilo bihejvioralne.

Na ovom mestu bi bilo korisno napomenuti da korišćenje biometrijske tehnologije nije sveobuhvatno rešenje za sva pitanja vezana za bezbednost i identifikaciju. Radi postizanja višeg nivoa bezbednosti neophodno je da biometrija bude deo šireg i celovitog sistema upravljanja rizicima koji obuhvata više različitih tehnologija zaštite. Savremeni biometrijski sistemi koji se nalaze u operativnoj upotrebi dokazuju uspešnost koncepta biometrijske identifikacije. Upotrebom savremenih tehnologija ostvaruje se efikasna i brza identifikacija osobe i u velikoj meri se sprečava krađa identiteta.

Nije suvišno imati u vidu i činjenicu da je cena koštanja biometrijskih uređaja kao i odgovarajućeg softvera opala u poslednjih nekoliko godina, sa tendencijom daljeg pojeftinjenja. Stoga se biometrijska autentifikacija prilikom prijavljivanja na računar može naći u standardnoj ponudi laptop računara različitih proizvođača, što uključuje hardverske i

softverske komponente. Ubrzani razvoj biometrijske tehnologije, kako hardvera tako i softvera, može se primetiti nakon terorističkih napada koji su se odigrali 11.septembra 2001. u SAD.

Biometrija se može koristiti samostalno ili u sprezi sa drugim tehnologijama, kao što su kartice sa mikrokontrolerom, tokeni ili kao ključevi u kriptografskim operacijama odnosno elementi digitalnih potpisa i digitalnih sertifikata itd. Identifikacija korišćenjem biometrijskih mehanizama je daleko pouzdanija nego što je to korišćenje lozinke, PIN-a ili posedovanje kartice sa mikrokontrolerom ili tokena. Kako je to opisano u prethodnom tekstu biometrijska identifikacija predstavlja najviši nivo identifikacije. Sa ubrzanim rastom elektronske trgovine javlja se sve veća potreba za autentifikacijom osobe koja vrši finansijsku transakciju. Platne kartice sadrže u sebi mehanizme utvrđivanja identiteta kako bi se izbegle sve eventualne zloupotrebe koje za posledicu mogu imati gubitak novca.

Prilikom akvizicije biometrijskih podataka neophodno je obezbediti kvalitetan reprezent biometrijske veličine. Ukoliko je sama biometrijska karakteristika lošeg kvaliteta, na primer jagodica prsta može imati ožiljke koji onemogućavaju kvalitetno čitanje otiska, samim tim i reprezent biometrijske karakteristike je nekvalitetan. Treba imati na umu da oko 5% svetske populacije ima veom plitke papile na jagodici prsta, tako da im *de facto* nije moguće očitati otisak prsta. Najveći broj osoba sa plitkim papilama se nalazi u Aziji. U svetu kriminala nije nepoznata pojava spaljivanja papila hlorovodoničnom kiselinom, kako otisci osobe ne bi bili prepoznatljivi. Očigledno je da slaba čitljivost ili nečitljivost biometrijskih podataka može znatno otežati primenu biometrijske tehnologije a samim tim i biometrijske identifikacije [5].

Mesto čuvanja templejta je od velike važnosti za bezbednost biometrijskih podataka a samim tim i za zaštitu privatnosti pojedinaca. Naročito ukoliko se radi o velikom broju podataka aplikacije koje pristupaju tim podacima moraju biti strogo kontrolisane. Pored toga, aplikacije moraju biti efikasne prilikom pretraživanja i manipulacije podacima. Ipak, na prvom mestu treba voditi računa o bezbednosti samih podataka i onemogućavanju neovlašćenog pristupa podacima. Imajući u vidu navedeno, jasno je da je izbor medijuma na kome se čuvaju biometrijski podaci veoma bitan faktor bezbednosti samih podataka. Neka od originalnih rešenja su prikazana u radovima [6] i [7].

Medijumi za čuvanje biometrijskih podataka se mogu podeliti u tri osnovne kategorije:

1. Skladištenje podataka u samom čitaču biometrijske veličine ili na računaru na koji je čitač konektovan, odnosno računaru koji kontroliše rad biometrijskog čitača.
2. Skladištenje podataka u centralnoj bazi kojoj se pristupa korišćenjem mrežnih komunikacija.
3. Skladištenje podataka na prenosivom tokenu koji može biti i kartica sa mikrokontrolerom.

1. Skladištenje podataka u samom čitaču biometrijske veličine ili na računaru na koji je čitač konektovan, odnosno računaru koji kontroliše rad biometrijskog čitača ima svoje prednosti kao i i svoje nedostatke.

Prednost čini relativno brzo izvršavanje aplikacija koje manipulišu templejtima. Takođe, ne postoji oslanjanje na eksterne procese ili mrežnu infrastrukturu za pristup templejtima. U slučaju kvara uređaja, isti može biti zamenjen drugim. Postoji i mogućnost povezivanja više čitača na istom računaru, tako da se templejti mogu razmenjivati korišćenjem memorije čitača.

Nedostatak je relativno mali broj templejta koji se može čuvati na čitaču, kao i potencijalna izloženost napadu templejta dok se nalazi na čitaču. Ukoliko dođe do kvara uređaja ili računara na koji je uređaj konektovan, templejt se mora ponovo učitati.

2. Skladištenje podataka u centralnoj bazi kojoj se pristupa korišćenjem mrežnih komunikacija koja u velikoj meri dotiče aktivnosti administratora sistema. Potrebno je obezbediti bezbedno mrežno okruženje sa dovoljno velikom brzinom prenosa podataka. U ovakvom pristupu se može očekivati relativno velika baza podataka pa je potrebno projektovati posebne mehanizme koje će ubrzati rad sa podacima, pogotovo kada je u pitanju pretraživanje baze podataka. Na ovaj način se može kreirati efikasan sistem za proveru identiteta.

Veoma bitan momenat predstavlja prenos biometrijskih templejta kroz mrežu. Podaci moraju biti zaštićeni nekom od metoda šifrovanja dok se zaštita same mreže nekim od sigurnosnih protokola (kao što je na primer SSL - *Secure Socket Layer*) podrazumeva.

Mogući nedostatak se može odraziti u vidu zagušenja mreže ukoliko veliki broj korisnika pristupa bazi podataka istovremeno. Problem se rešava proširenjem kapaciteta mreže kao i projektovanjem efikasnijih metoda za upravljanje bazom podataka. Veličina samog templejta obično nema značajnog uticaja na brzinu pristupa bazi, imajući u vidu da templejt otiska prsta može da ima veličinu od 6 kB do 9kB, što je relativno mala količina podataka.

Problem se može javiti i ukoliko dođe do prekida mrežnog saobraćaja. U tom slučaju su moguća dva rešenja. Najbolje je imati rezervni link ka bazi koji se automatski aktivira ukoliko dođe do prekida saobraćaja po jednoj konekciji. Ili, drugo rešenje je da se baza podataka pamti na lokalnim resursima u keš memoriji. Istini za volju, drugo rešenje nije pozdano jer u međuvremenu može doći do promene nad podacima u centralnoj bazi.

3. Skladištenje podataka na prenosivom tokenu koji može biti i kartica sa mikrokontrolerom je poželjno iz tri osnovna razloga:

- Ne zahteva lokalnu ili centralnu bazu podataka.
- Korisnik nosi svoj templejt sa sobom i može ga koristiti prema potrebi na odgovarajućem uređaju - čitaču.
- Step en privatnosti je podignut na najviši mogući nivo.

Čak se i ne mora koristiti kartica sa mikrokontrolerom, već to može biti i obična plastična kartica na kojoj će biti odštampan templejt u vidu bar koda ili Q koda. Može se koristiti i kartica sa optičkom memorijom, ili hibridna kartica koja pored optičke memorije sadrži i mikrokontroler. U krajnjoj liniji to može biti token realizovan u bilo kojoj tehnologiji.

Potencijalnu opasnost predstavlja gubitak ili oštećenje prenosivog medijuma. Jedna od centralnih tema istraživanja u ovom radu će biti zaštita biometrijskih templejta motivisana upravo mogućnošću kompromitovanja biometrijskih podataka. O ovome će detaljno biti reči u delu teksta koji detaljno opisuje istraživanje i dobijene rezultate.

Upotreba određenog biometrijskog podatka nosi svoje specifičnosti. Za upravljanje određenom vrstom biometrijskih podataka se mora razviti specifičan softver. Ipak, iako se softver razvija za svaku vrstu biometrije koja se koristi, postoje i zajednički imenitelji koji se mogu uočiti za sve biometrijske aplikacije. Na toj osnovi se biometrijske aplikacije mogu podeliti u sledeće kategorije [8]:

- Sistem može biti otvoren ili prikriven. U otvorenom sistemu korisnik je svestan da se njegov identitet proverava dok se u prikrivenom sistemu vrši provera identiteta osobe bez njegovog znanja. Primer otvorenog sistema je skeniranje otiska prsta prilikom provere identiteta na graničnom prelazu. Primer korišćenja prikrivenog sistema se može videti u nekim prodavnicama u Amsterdamu u kojima kamera snima lice svakog ko ulazi u prodavnicu. Sistem automatski detektuje lik i pretražuje bazu podataka u kojoj se nalaze osobe koje su toj prodavnici vršili prekršaje. Ukoliko se pronađe osoba u bazi, alarmira se obezbeđenje koje udaljuje osobu iz prodavnice.
- Dobrovoljni ili prinudni sistemi se razlikuju po tome da li se zahteva aktivno učešće osobe kojoj se utvrđuje identitet ili se utvrđivanje identiteta vrši bez znanja i bez saglasnosti osobe čiji se identitet utvrđuje.
- Sistemi koji zahtevaju ili ne zahtevaju neposredno prisustvo osobe kojoj se utvrđuje identitet. Postavlja se pitanje da li će sistem za identifikaciju biti realizovan korišćenjem automatskih funkcija za identifikaciju osoba na udaljenoj lokaciji ili je neophodna intervencija tehničara i neposredno prisustvo osobe čiji identitet utvrđujemo. Prinudni i prikriveni sistemi obično zahtevaju nadgledanje od strane tehničara, dok otvoreni i dobrovoljni sistemi mogu funkcionisati bez nadzora tehničara.
- Uslovi rada sistema za identifikaciju mogu biti standardni ili nestandardni. Za rad čitača biometrijskih karakteristika u nestandardnim uslovima (ekstremne temperature, vlažnost, uticaj prašine ili peska itd.) neophodno je izvršiti određena podešavanja samih čitača. Pored toga, u nestandardnim uslovima potrebno je obezbediti kvalitetne komunikacione kanale za komunikaciju između čitača i baze podataka. Ukoliko se sistem koristi u zatvorenom prostoru uslovi rada se smatraju standardnim, dok ukoliko se sistem za identifikaciju koristi na otvorenom, uslovi rada se smatraju nestandardnim.

- Sistemi mogu biti javni ili privatni. Javni sistemi se koriste u sistemima javne bezbednosti dok se privatni sistemi koriste za obezbeđenja privatnih objekata ili kompanija. Koordinacija pomenute dve vrste sistema je poželjna i uobičajena.
- Sistemi za fizičko obezbeđenje i kontrolu pristupa identifikuje osobe koje žele da pristupe kontrolisanom fizičkom prostoru.
- Sistemi za kontrolu pristupa računarski resursima kao što su računar ili mrežne komunikacije i mrežni uređaji. U pomenutu kategoriju sistema spada i kontrola pristupa Internet sadržajima.
- Sistemi za identifikaciju koriste biometriju za različite namene: kontrolu pristupa informacijama, kontrolu prelaska državne granice, elektronska plaćanja itd.

Greška i veličina greške

Nijedan biometrijski sistem ne može da izvrši apsolutno tačno prepoznavanje osobe. Iako se konačni odgovor biometrijskog sistema svodi na potvrdu ili negaciju sam proces nije tako jednostavan. Sistem zapravo meri stepen sličnosti između prezentovanog biometrijskog templejta i templejta koji se nalazi u bazi podataka i sa kojim se vrši upoređivanje. Postoji nekoliko tipova grešaka koje se pojavljuju u biometrijskim aplikacijama, od koji se kao dva osnovna tipa greške mogu navesti greška prilikom upoređivanja dva templejta i greška prilikom donošenja odluke o istovetnosti dva templejta.

Greške koje će biti razmatrane u narednom delu teksta će biti sagledavane zajedno sa ogovarajućim stepenom svake greške koji kvantifikuje veličinu napravljene greške. Na taj način se prilikom pogrešne pozitivne identifikacije posmatra greška pogrešne pozitivne identifikacije (*FMR - False Match Rate*). Analogno, prilikom pogrešne negativne identifikacije posmatra se greška pogrešne negativne identifikacije (*FNMR - False Non-Match Rate*). Pomenute greške su uočene prilikom testiranja upoređivanja biometrijskih templejta i analiziranjem dobijenih rezultata [9].

Ili, matematičkim rečnikom, stopa predstavlja očekivanu verovatnoću da će se desiti greška u posmatranom biometrijskom sistemu. Pomenute stope daju merljive pokazatelje koji omogućavaju upoređivanje efikasnosti različitih tehnologija u biometriji.

Greške poređenja se mogu posmatrati analizom pogrešnih pozitivnih upoređivanja ili pogrešnih negativnih upoređivanja. Kao jedan od razloga pojavljivanja pomenute greške, može se razmatrati funkcionalnost harvera koji se koristi u svrhu identifikacije. Realitvno često je osnovni uzrok greške prilikom identifikacije upravo greška koja se dešava prilikom akvizicije biometrijskog podatka na hardverskom uređaju.

Pogrešna identifikacija predstavlja donošenje pogrešne odluke da su dva biometrijska templejta istovetna iako to nisu. Odnosno reč je o pogrešnom zaključku da je prezentovani templejt jednak templejtu u bazi podataka koji je vezan za drugu osobu a ne onu koja je prezentovala svoj templejt.

Pogrešna negativna identifikacija predstavlja donošenje pogrešne odluke da su dva biometrijska templejta različita iako to nisu. Odnosno reč je o pogrešnom zaključku da prezentovani templejt nije jednak templejtu u bazi podataka koji je vezan za istu osobu iako se radi o istoj osobi.

Greška prilikom odlučivanja se odnosi na donošenje pogrešnih zaključaka na osnovu grešaka prilikom upoređivanja dva templejta. Uzrok greške prilikom upoređivanja zavisi od same aplikacije koja se koristi kao i biometrijske veličine koju aplikacija obrađuje.

Pogrešno prihvatanje identiteta u aplikaciji, kao što je aplikacija za kontrolu pristupa, je donošenje pogrešnog zaključka pri čemu se prihvata identitet osobe onako kako se predstavila iako se radi o drugoj osobi. Pri tome se za prezentovani biometrijski templejt ustanovljava da je istovetan sa nekim drugim biometrijskim templejtom u bazi podataka. Stopa pogrešnog prihvatanja identiteta (*FAR – False Acceptance Rate*) je očekivana verovatnoća da će biti pogrešno prihvaćen identitet osobe iako se radi o potpuno drugoj osobi.

Pogrešno negiranje identiteta prilikom kontrole pristupa, je donošenje pogrešnog zaključka pri čemu se negira identitet osobe onako kako se predstavila iako se radi o toj osobi. Pri tome se za prezentovani biometrijski templejt ustanovljava da nije istovetan sa originalnim biometrijskim templejtom u bazi podataka. Stopa pogrešnog negiranje identiteta (*FRR – False Rejection Rate*) je očekivana verovatnoća da će biti pogrešno negiran identitet osobe iako se radi o istoj osobi.

Radi pojašnjenja, skraćenice FAR i FRR su sinonimi za FMR i FNMR respektivno. Tokom vremena je došlo do izmene terminologije tako da su skraćenice FAR i FRR postali široko prihvaćene i korišćene.

Posledica pogrešnog prihvatanja identiteta je dozvoljavanje pristupa sistemu osobi koja nije ta za koju tvrdi da jeste. Na taj način sistem je kompromitovan jer se neovlašćenoj osobi dozvoljavaju akcije u sistemu.

Posledica pogrešnog negiranja identiteta je sprečavanje pristupa sistemu osobi koja bi na to trebalo da ima pravo. Na taj način sistem onemogućava ovlašćenoj osobi legitimne akcije u sistemu.

Neuspeh prilikom akvizicije može da unese dodatne probleme prilikom identifikacije. Obično nastaje iz tehničkih razloga nemogućnosti prevođenja biometrijske karakteristike u digitalni oblik, odnosno templejt. Uzrok može biti neispravnost ili loš kvalitet samog biometrijskog čitača ili slabo izražena biometrijska karakteristika osobe kao što su plitke papile na jagodici prsta. Najčešća posledica neuspeha prilikom akvizicije je pogrešno negiranje identiteta.

Može se konstatovati da se proces poređenja svodi na proces upoređivanja jednog ili više prezentovanih biometrijskih templejta sa jednim ili više biometrijskih templejta koji se čuvaju u bazi podataka. Cilj poređenja je utvrđivanje da li se biometrijski templejti poklapaju ili ne i na taj način se utvrđuje identitet osobe.

Iz prethodno navedenog može se zaključiti da postoje tri osnovne klase grešaka prilikom biometrijske identifikacije:

1. Neuspeh prilikom akvizicije
2. Pogrešno prihvatanje identiteta (*FAR*)
3. Pogrešno negiranje identiteta (*FRR*)

Greške prilikom očitavanja biometrijske kao i greške prilikom kreiranja binarnog reprezentata biometrijske veličine za posledicu imaju nemogućnost kreiranja kvalitetnog templejta. Kao što je već pomenuto, određeni broj osoba ima plitke papile na jagodicama. Takođe postoje osobe kojima su otisci oštećeni usled zanimanja kojima se bave ili kriminalci koji su namerno uništili svoje otiske. Pomenutim grupama nije moguće kvalitetno skenirati otiske prstiju, te se kod tih osoba otisak prsta ne može posmatrati kao relevantna biometrijska karakteristika.

U većini biometrijskih sistema mogu se podesiti parametri kojima se definiše prag prihvatanja ili odbijanja identiteta. Pomenimo na ovom mestu pojam minucije, koji će u daljem delu teksta biti detaljnije objašnjen. Primer skupa karakterističnih tačaka - minucija je predstavljen na slici 5. Minucije predstavljaju karakteristične tačke otiska prsta definisane svojim koordinatama kao uglom definisanim između tangente na papilu i pozitivnog dela x ose. Posmatrajmo jednu uočenu karakterističnu tačku - minuciju u Dekartovom pravouglom koordinatnom sistemu na slici 6.

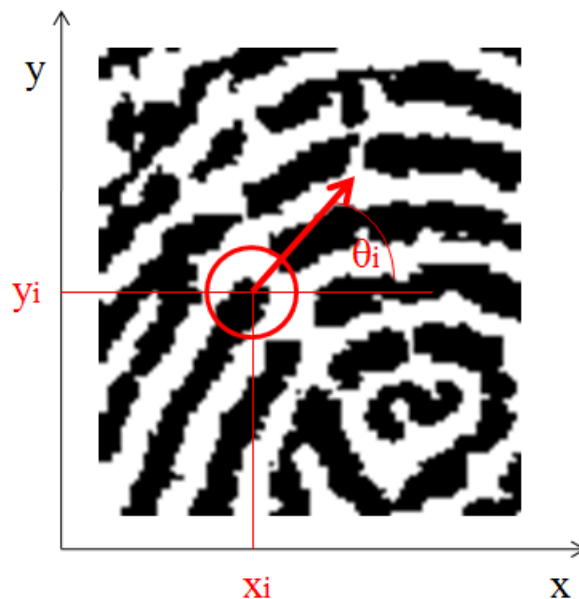
Prilikom posmatranja minucije, može se uočiti je definisana sa tri elementa od kojih su dve koordinate u Dekartovom pravouglom sistemu x_i, y_i , kao i uglom koji zaklapa tangenta na papilu u tački minucije sa pozitivnim delom x ose, θ_i . Indeks i označava i -tu minuciju [10]. Pretpostavka je da otisak prsta sadrži više od jedne minucije, u protivnom razmatranje minucija otiska prsta kao biometrijskog sistema za prepoznavanje ne bi imao smisla.

Pre daljeg razmatranja, potrebno je napomenuti da se prilikom ove analize posmatra pojednostavljeni model.



Slika 5. Primer skupa minucija jednog otiska prsta

O tipovima minucija kao i algoritmu za njihovo uočavanje biće reči u daljem tekstu.



Slika 6. Označavanje koordinata izabrane minucije

Ukoliko bi se prilikom upoređivanja koordinata zahtevalo da se tri koordinate apsolutno poklapaju prilikom svakog merenja, očigledno je da koordinata posmatrane minucije nikad ne bi bila potvrđena zbog greške merenja.

Kao što je poznato u teoriji merenja, greške merenja se mogu svrstati u tri osnovne kategorije:

Grube greške nastaju nepažnjom osobe koja vrši merenje, izborom neodgovarajuće metode (postupka) merenja ili zbog neuočavanja uzroka greške. Grube greške merenja izbegavaju se većom pažnjom prilikom merenja. Najčešće rezultati sa grubom greškom značajno odstupaju od srednje vrednosti većeg broja merenja, pa ih je lako uočiti tokom analize i odstraniti iz skupa mernih podataka.

Sistematske greške nastaju zbog nesavršenosti merne opreme (skenera biometrijske karakteristike), procedure merenja ili mernog postupka, mernog objekta, kao i zbog uticaja okoline i ličnih uticaja osobe koja vrši merenje. Ove greške imaju određenu vrednost i predznak, pa mogu da se uzmu u obzir prilikom korekcije. Postoje i sistematske greške čija vrednost nije poznata, pa nije moguća njihova korekcija s obzirom da imaju nepoznatu fiksnu odnosno stalnu vrednost i nepoznati, ali uvek isti, predznak.

Zbog navedenog, uvek je potrebno imati u vidu ograničenja merne opreme koja se koristi. Nikada ne treba posmatrati mernu opremu kao idealnu, već treba uvidom u dokumentaciju i određenim proverama odrediti karakteristike koje mogu da dovedu do greške u merenju.

Slučajne greške nastaju usled neobuhvatljivih promena koje nastaju u mernim instrumentima i mernom objektu i usled neobuhvatljivih uticaja okoline i osobe koja vrši merenje. Ove greške su rezultat velikog broja faktora koji različito deluju na svako pojedinačno merenje. Zbog toga se menja i apsolutni iznos i predznak slučajne greške. Ove greške dovode do rasipanja rezultata i do nesigurnosti u određivanju stvarne vrednosti merenja [11].

Pojedinačni merni rezultati sa slučajnim greškama su podjednako značajni za merenje, ni jedan od njih nije u prednosti. Gaus je 1795. godine (*Johann Carl Friedrich Gauss, 1777. - 1855.*) pokazao da je u ovom slučaju prema metodi najmanjih kvadrata srednja vrednost (aritmetička sredina) pojedinačnih rezultata najverovatnija vrednost merene veličine.

Slučajna greška u biometrijskim merenjima ima poseban značaj. Posmatrajmo skeniranje otiska prsta na uređaju za skeniranje. Od osobe kojoj se skenira otisak, zahteva se da prsloni jagodicu prsta na skener. Imajući u vidu da je jagodica prsta sastavljena od mekog tkiva, prilikom prislanjanja jagodice dolazi do deformacije mekog tkiva. Ukoliko je pritisak na uređaj za skeniranje veći, deformacija tkiva će biti veća. Zaključujemo da izgled samog otiska zavisi, između ostalog, od intenziteta pritiska jagodice prsta na uređaj za skeniranje. Potpuno je jasno da osoba ne može dva puta ponoviti potpuno identičnu vrednost pritiska, iz čega sledi da ne možemo dva puta dobiti istovetni otisak iako se radi o istoj osobi i istom prstu.

Na osnovu iznetog se može zaključiti da koordinate x_i, y_i, θ_i „nemaju uvek identičnu vrednost. Ukoliko se za identifikaciju osobe zahteva apsolutna tačnost navedenih koordinata gotovo je sigurno da nikada ne bismo pozitivno identifikovali osobu. Iz tih razloga se uvodi dozvoljeno odstupanje ε ili ε okolina u kojoj se može naći vrednost svake od posmatranih koordinata. Određivanjem vrednosti veličine ε podešava se stepen prihvatanja ili odbijanja identifikacije.

Ukoliko je ε relativno malo, sistem će biti više sklon da odbija pozitivnu identifikaciju iako se radi o istoj osobi. Ukoliko je ε relativno veliko, sistem će biti više sklon da prihvata pozitivnu identifikaciju iako se ne radi o istoj osobi. Pronalaženje „prave mere“ za ε zahteva testiranje više različitih metoda i vršenje više serija merenja.

Na osnovu iznetog, može se uočiti da je u većini biometrijskim sistemima moguće podešavanja praga prepoznavanja već u zavisnosti od nivo bezbednosti koji se želi postići. Nakon poređenja dva templejta donosi se ocena o stepenu sličnosti ili razlike, u zavisnosti od korišćenog sistema. Za algoritam koji posmatra uočenu sličnost između dva biometrijska reprezentata, stepen sličnosti mora biti iznad postavljenog praga sličnosti. Za algoritam koji posmatra uočenu razliku između dva biometrijska reprezentata, stepen sličnosti mora biti ispod postavljenog praga sličnosti [12].

Može se definisati nekoliko problema koji su povezani sa određenim algoritmima u kontekstu prepoznavanja otiska prsta. Termin prepoznavanje otiska prsta pretpostavlja postupke verifikacije i identifikacije.

Iako postoje definicije koje posmatraju ovu problematiku iz drugog ugla u ovom radu ćemo koristiti termine koji se najčešće koriste prilikom prepoznavanje otiska prsta.

Sistemi za *verifikaciju ili autentifikaciju* se koriste radi potvrde identiteta osobe. Ovi sistemi koriste dva ulazna podatka od kojih prvi može biti korisničko ime ili kartica sa mikrokontrolerom ili token na kojima su sadržani identifikacioni podaci, dok je drugi ulazni podatak reprezent otiska prsta koji je skeniran na licu mesta. Identitet osobe koji se navodi se pronalazi u bazi podataka i pronalazi se njemu pridruženi biometrijski reprezent (jedan prema jedan). Tada se utvrđuje stepen sličnosti između dva biometrijska reprezentata i na osnovu stepena sličnosti se donosi odluka o pozitivnoj ili negativnoj identifikaciji.

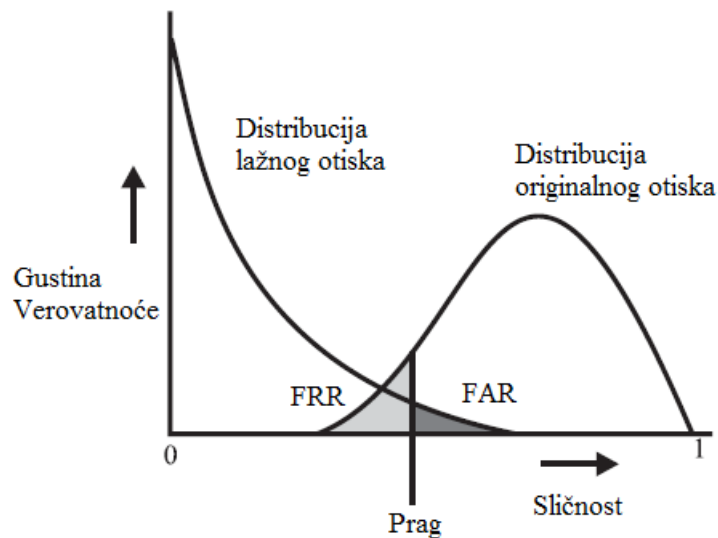
Sistemi za *identifikaciju* identifikuju osobu na osnovu otiska prsta. Takvi sistemi dobijaju samo jedan ulazni podatak, odnosno uživo skenirani otisak prsta. Vršiti se pretraživanje baze podataka (jedan prema više) kako bi se pronašao odgovarajući otisak prsta, odnosno njegov reprezent. Osoba se identifikuje ako se pronađe odgovarajući otisak prsta u bazi podataka. Sistem dodeljuje identitet osobi koji odgovara pronađenom otisku prsta. U slučaju da odgovarajući otisak nije pronađen identitet osobe ostaje nepoznat.

Iz prethodno rečenog očigledno je da je baza podataka otisaka neophodna kako za proces verifikacije odnosno autentifikacije tako ni proces identifikacije. Možemo zaključiti da je proces prikupljanja otisaka i njihovo upisivanje u bazu podataka veoma bitan proces bez koga ni verifikacija ni identifikacija ne bi bili mogući.

Merenje greške

U sistemima za verifikaciju i identifikaciju, provera podudaranja dva otiska predstavlja bitnu funkciju. Rezultat izvršavanja ove funkcije je tvrdnja da se dva otiska poklapaju ili se ne poklapaju. Potrebno je setiti se tvrdnje koja je izrečena u prethodnom tekstu a to je da ne postoji apsolutno poklapanje već određeni stepen poklapanja. Kako bi se stepen poklapanja numerički kvantifikovao uvodi se numerička vrednost koja definiše stepen poklapanja dva otiska. Nakon toga se, na osnovu pomenute numeričke vrednosti donosi odluka o poklapanju ili ne poklapanju dva otiska.

Generalno posmatrajući, raspodela sličnosti pokušaja identifikacije originalnog otiska i lažnog otiska nije potpuno odvojena pragom prepoznavanja. U jednoj oblasti postoji preklapanje navedenih raspodela što dovodi do podudaranja velične greške za pravi i lažni otisak. Dva otiska se smatraju podudarnim ukoliko njihova sličnost prelazi postavljeni prag, odnosno dva otiska nisu podudarna ukoliko je njihova sličnost ispod definisanog praga. Grafički prikaz raspodele sličnosti je prikazan na slici 7 [13].



Slika 7. Uporedni prikaz raspodele sličnosti za pravi i lažni otisak

Na ovom mestu je potrebno definisati gustinu ili raspodelu verovatnoće.

U teoriji verovatnoće funkcija raspodele ili gustine verovatnoće ili gustine kontinuirane slučajne promenljive je funkcija koja opisuje relativnu verovatnoću da posmatrana slučajna promenljiva uzme zadatu vrednost. Verovatnoća da se vrednost slučajne promenljive nađe u posmatranom intervalu se izračunava određenim integralom posmatrane promenljive u definisanom intervalu. Gustine verovatnoće je nenegativna funkcija na čitavoj svojoj oblasti definisanosti. Vrednost određenog integrala funkcije gustine verovatnoće na čitavom domenu jednaka je jedinici.

Neka je X neprekidna slučajna promenljiva. Tada je funkcija $f(x)$ definisana nad skupom realnih brojeva $\mathbb{R}$, funkcija gustine verovatnoće (*probability density function - pdf*) promenljive X za bilo koja dva broja a i b pri čemu je $a \leq b$:

$$P(a \leq X \leq b) = \int_a^b f(x) dx \quad (1)$$

Iz izraza (1) se može uočiti da je verovatnoća da će X uzeti neku od vrednosti na segmentu $[a, b]$ jednaka površini ispod grafika funkcije gustine verovatnoće $f(x)$ na posmatranom segmentu.

Performanse sistema za verifikaciju otiska prsta se može oceniti analiziranjem dve osnovne greške merenja:

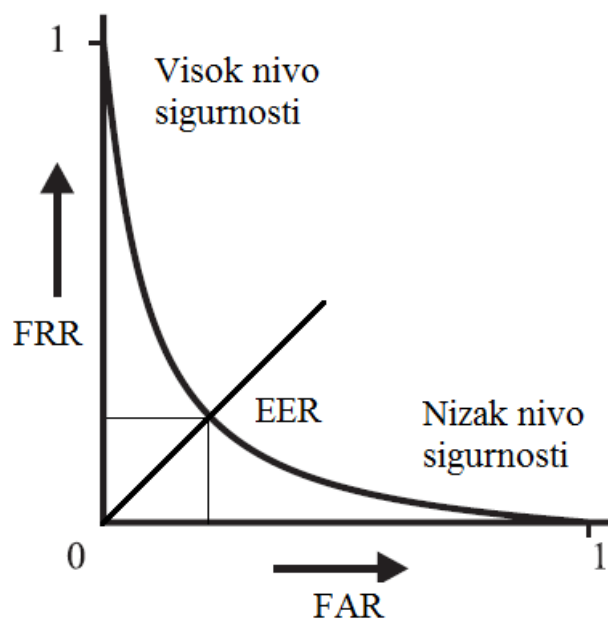
Pogrešno prihvatanje identiteta - FAR predstavlja verovatnoću da će sistem prihvatiti lažni otisak kao originalni. Vrednost FAR je prikazana na slici 7. osenčenom površinom desno od

praga prihvatanja.

Pogrešno negiranje identiteta - FRR verovatnoću da će sistem odbaciti originalni otisak prsta kao lažni. Vrednost FRR je prikazana na slici 7. osenčenom površinom levo od praga prihvatanja.

Savremeni sistemi koji se koriste za identifikaciju otiska prsta postižu različite vrednosti FAR i FRR, kako u zavisnosti od algoritama realizovanih u sistemu tako i od kvaliteta podataka u bazi podataka. Kompanija „Digital Persona“ (Redwood City, California, United States), na kvalitetnijim bazama podataka, ostvaruje rešenja pri kojima je $FAR = 10^{-4}$ dok je $FRR = 10^{-2}$. Na realnim bazama podataka kakve koristi FBI (*Federal Bureau of Investigation*) postižu se rezultati pri kojima je $FAR = 10^{-2}$ i $FRR = 5 \times 10^{-2}$.

U većini slučajeva procena kvaliteta sistema za identifikaciju korišćenjem otiska prsta se može izvršiti kako je to opisno u prethodnom pasusu. Pored toga, veoma korisni podaci se mogu dobiti i grafičkim prikazom FAR i FRR u istom koordinatnom sistemu kao što je prikazano na slici 8. Prikazana kriva se naziva ROC (*Receiver Operating Curve*) kriva. Prag se može podesiti u zavisnosti od namene same aplikacije i svrhe u koje se koristi. U nekim sistemima je potrebno ostvariti visoku sigurnost odnosno što manju grešku prilikom prepoznavanja (nisku vrednost FAR), dok se u nekim drugim sistemima može dozvoliti fleksibilniji pristup (nisku vrednost FRR). Tačka krive u kojoj su FAR i FRR jednaki je tačka jednake greške (*EER - Equal Error Rate*).



Slika 8. ROC (*Receiver Operating Curve*) kriva

Postizanje niske stope greške je mnogo teže u sistemima za identifikaciju nego što je to u sistemima za verifikaciju. Razmotrimo primer identifikacije otiska prsta u bazi podataka od $n = 10000$ arhiviranih otisaka pri čemu sistem upoređuje uneti otisak sa svakim arhiviranim otiskom u bazi podataka. Algoritam za upoređivanje dva otiska prsta visokih performansi pri čemu je $FAR = 10^{-4}$, prosečno će izvršiti $n \times FAR$ upoređivanja sa prihvatanjem pogrešnog otiska kao originalnog. U posmatranom primeru jedan pogrešni otisak će biti prihvaćen kao originalan. Verovatnoća prihvatanja pogrešnog otiska kao originalnog, nad celom posmatranom bazom podataka se može izračunati kao:

$$FAR_{1:n} = 1 - (1 - FAR)^n \quad (2)$$

što će u posmatranom primeru iznositi 0,63. Jasno je da je stepen greške preveliki i neupotrebljiv za praktične primene. U daljem tekstu će detaljnije biti razmotrene metoda koje kompenzuju ovo greške i čine sistem pouzdanijim za praktične primene [14].

2.1.2 Pregled biometrijskih veličina

Ljudski um ima razvijenu sposobnost da prepozna poznatu osobu kako na osnovu fizičkih karakteristika tako i na osnovu ponašanja osobe. Prepoznavanje se najčešće vrši na osnovu specifičnih karakteristika kao što su: lik, oči, glas, hod, rukopis, potpis i slično. Neki ljudi imaju mogućnost prepoznavanja više od drugih, imaju čak i mogućnost da prepoznaju nepoznate osobe, pri čemu je dovoljno da su ih samo jednom videli ili čuli.

Prilikom izbora biometrijske karakteristike koje se koriste za prepoznavanje potrebno je voditi računa o uslovima koje je potrebno da izabrana biometrijska veličina zadovolji. Osnovni postulati koje je potrebno da zadovoljavaju izabrane biometrijske veličine su:

- *Robusnost* – podrazumeva da se karakteristika vremenom ne menja, odnosno da se menja u veoma maloj meri. Potrebna je stalnost karakteristike.
- *Karakterističnost* – podrazumeva da posmatrana osobina poseduje značajnu varijabilnost između jedinki. Potrebno je da postoji jedinstvenost osobine za svaki osobu.
- *Dostupnost* – podrazumeva da svaka jedinka u populaciji poseduje izabranu biometrijsku veličinu. Odnosno potrebna je univerzalnost izabrane karakteristike.
- *Pristupačnost* – podrazumeva da se izabrana karakteristika može lako skenirati pomoću specijalizovanih uređaja i prevesti u oblik pogodan za dalju obradu.

U narednom delu teksta biće dati kratki opisi pojedinih biometrijskih veličina koje se najčešće koriste u savremenim biometrijskim sistemima.

Otisak prsta

Prvenstveno treba imati na umu da je svaki otisak prsta jedinstven. Ključno pitanje je da li i kako možemo doći do informacije koja je jedinstvena i prikazati je na način koji omogućava kvalitetno prepoznavanje. Dobijanje i predstavljanje pomenute informacije predstavlja predmet mnogih istraživanja.

Merenje jedinstvene fizičke karakteristike kao što je otisak prsta predstavlja jedno od kvalitetnih rešenja digitalne identifikacije. Digitalna identifikacija kreira digitalni identitet koji se može efikasno koristiti u virtuelnom svetu. Iako se mogu koristiti mnoge biometrijske karakteristike, otisak prsta predstavlja jednu od najšire korišćenih osobina.

Prednosti korišćenja biometrije otiska prsta u cilju kreiranja digitalnog identiteta u odnosu na druge metode se mogu argumentovati sledećim činjenicama:

- Svaki od naših deset prstiju ima jedinstveni otisak i razlikuje se od otiska bilo kog prsta svake druge osobe. Čak i jednojajčani blizanci imaju različite otiske prstiju.
- Za razliku od lozinke, PIN koda ili kartice sa čipom od kojih zavisi naša identifikacija, otisci prstiju se ne mogu zaboraviti, izgubiti ili biti ukradeni. Istini za volju postoje načini falsifikovanja otisaka prstiju, ali o tome će biti reči u daljem tekstu.
- Imamo deset prstiju u odnosu na jedan glas, jedno lice ili dva oka. Brojčana prednost uzoraka je više nego očigledna.
- Otisci prstiju se koriste za identifikaciju tako da su baze podataka otisaka prstiju najobimnije i najsveobuhvatnije, što je potvrdilo jedinstvenost otiska prsta. Korišćenje drugih biometrijskih veličina kao što je iris, je relativno nova metoda pa su samim tim i baze podataka manje.

Ukoliko pogledamo strukturu ljudske kože, možemo uočiti da se ona sastoji iz više slojeva, od koji se izdvajaju tri osnovna a to su [15]:

- epidermis
- dermis
- hipodermis

Epidermis predstavlja spoljni sloj kože koji vidimo, i koji prekriva veliku većinu ljudskog tela. Nabori na jagodicama prsta se nalaze u sloju epidermisa odnosno površinskog sloja. Izgled jagodice prsta je prikazan na slici 9.

Nabori na jagodicama prsta su se razvili tokom evolucije i razlog njihovog nastanka je povećanje sposobnosti hvatanja i držanja predmeta prstima. Kada prsti postanu mokri a samim tim i dodatno klizavi, koža na prstima se dodatno nabira, što je gotovo svaki čovek

primetio nakon kupanja ili plivanja. Nabori na koži jagodice prsta dobijaju specifične oblike kod svakog čoveka i, kako se do sada pokazalo, ti oblici su jedinstveni i neponovljivi.

Neki istraživači dovode u vezu veštinu upotrebe prstiju u hvatanju i držanju sitnih predmeta sa razvojem inteligencije. Mnogi dečiji psiholozi tvrde da igranje dece sitnim predmetima, kao što su „Lego“ kockice razvija inteligencija kod dece.



Slika 9. Nabori na jagodici prsta su predmet istraživanja

Otisci prstiju su izuzetno složeni i radi obrade otisaka potrebno je izvršiti njihovu klasifikaciju. U tom cilju se posmatraju dve vrste karakteristika otiska prsta koji i to su:

- globalne karakteristike
- lokalne karakteristike

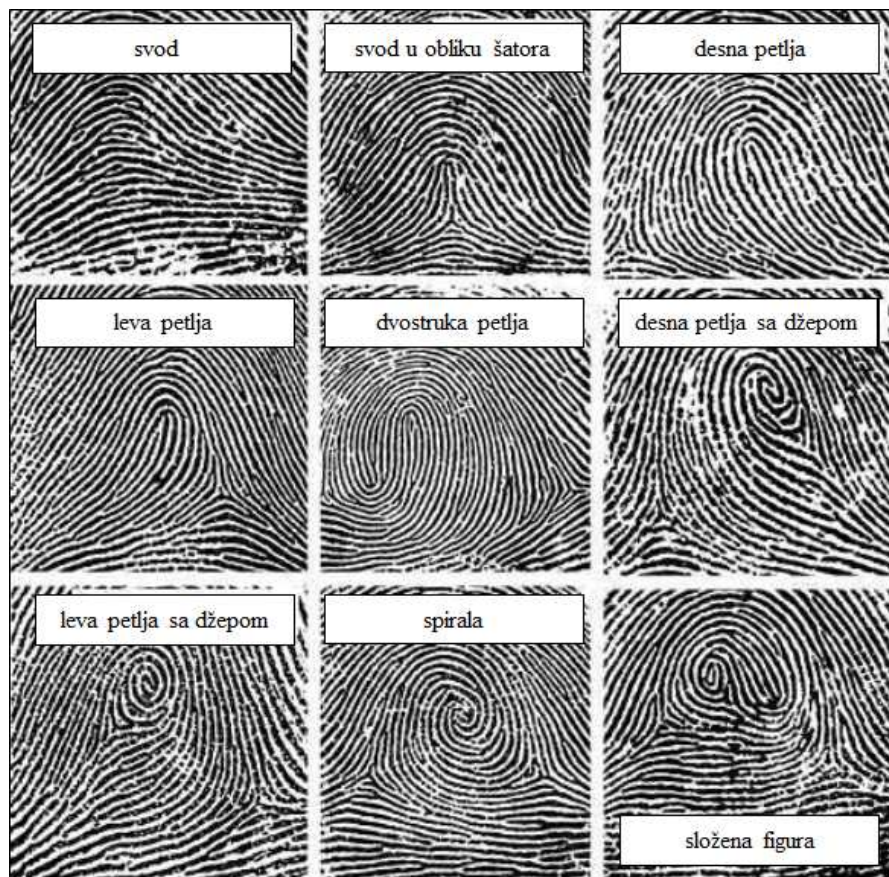
Globalne karakteristike definišu tipove krivih linija koje se mogu zapaziti na otiscima prstiju dok su lokalne karakteristike karakteristične tačke - minucije.

Moguće je da dve ili više osobe imaju identične globalne funkcije, ali imaju različite i jedinstvene otiske prstiju jer se lokalne karakteristike - minucije, međusobno razlikuju.

Globalne karakteristike definišu osnovne oblike krivih linija. Klasifikacija se može uraditi na više načina, dok će u ovom radu biti razmatrana podela prema kojoj su osnovni oblici [16]:

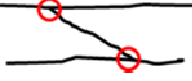
1. svod,
2. svod u obliku šatora,
3. desna petlja,
4. leva petlja,
5. dvostruka petlja,
6. desna petlja sa džepom,
7. leva petlja sa džepom,
8. spirala,
9. složena figura.

Izgled svakog od pomenutih oblika prikazan je na slici 10.



Slika 10. Globalne karakteristike otiska prsta

Minucije se definišu u karakterističnim tačkama. Određivanje karakterističnih tačaka se može odrediti na više načina, od kojih je jedno viđenje dato u Tabeli 1.

završetak brega	
račvanje – bifurkacija	
ostrvo	
tačka	
most	

kuka	
oko	
uzastopno račvanje	
delta	
dvostruko račvanje	

Tabela 1. Način određivanja karakterističnih tačaka - minucija

Na kraju ovog kratkog informativnog izlaganja može se postaviti pitanje da li su otisci prstiju zaista jedinstveni? Ovo pitanje je fundamentalno jer ceo sistem prepoznavanja na osnovu otiska prsta počiva na pretpostavci da su svi otisci prstiju kompletne svetske populacije različiti. Uverenje da su svi otisci prstiju međusobno različiti počiva na dva osnovna principa:

- Poznato je koliko informacija može da sadrži jedan otisak prsta. O statističkim i matematičkim modelima će biti reči prilikom analize entropije podataka koje nosi otisak prsta. Svaki otisak se može opisati globalnim i lokalnim karakteristikama i na otisku prsta se može uočiti i do 70 minucija. Mogućnost pronalaženja skupa identičnih minucija je zanemarljiva.
- Baza podataka otisaka prstiju F.B.I. postoji gotovo čitav vek i puni se realnim otiscima prstiju. Treba imati u vidu da se u SAD otisci prstiju mogu uzeti jedino registrovanim kriminalcima. Ova odredba se nakon terorističkih napada na kule blizakinje primenjuje i na sve strance koji ulaze na teritoriju SAD. Baza podataka otisaka prstiju F.B.I. sadrži preko 200 miliona otisaka, od kojih je svaki jedinstvena u odnosu na ostale. U proteklih sto godina F.B.I. nije identifikovala dva identična otiska.

Ovoliko dugu tradiciju i odgovarajuću bazu podataka ovolikog obima ne poseduje ni jedna druga biometrijska veličina.

Iris - Dužica

Iris sadrži karakteristične podatke koji se mogu koristiti za prepoznavanje osobe. Iris ima jedinstveni izgled i pouzdan je za prepoznavanje osobe. Izgled irisa je delom genetski uslovljen, naročito njegova boja. Međutim u toku razvoja oka dolazi do savijanja tkiva membrane irisa, što dovodi do pojave slučajnosti izgleda a samim tim i do

jedinstvenosti izgleda irisa [17].

U poređenju sa drugim tehnikama prepoznavanja iris ima veliku prednost jer postoji ogromna varijabilnost između pojedinaca što znači da se može vršiti veliki broj upoređivanja bez pogrešnog uparivanja neistovetnih podataka. Možemo zaključiti da je prepoznavanje skeniranjem irisa korisnije koristiti za identifikaciju nego samo za potvrdu identiteta. Primer korišćenja skeniranja zenice je provera identiteta na graničnim prelazima.

Proces kreiranja biometrijskog templejta irisa se sastoji od tri osnovna koraka:

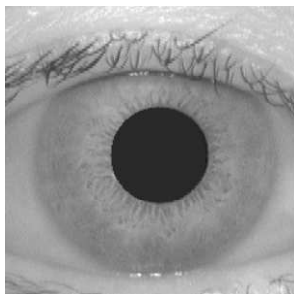
- snimanje slike oka,
- definisanje lokacije irisa i optimizacije slike,
- smeštanje templejta u bazu i upoređivanje.

Slika irisa se može dobiti korišćenjem standardnog fotoaparata korišćenjem oblasti vidljivog ili infracrvenog spektra. Udaljenost foto aparata bi trebalo da bude između 10 cm i 1m da bi se dobila kvalitetna fotografija, što zavisi od optičkog objektiva koji se koristi. Ukoliko se snimanje vrši ručno, potrebno je da osoba koja fotografiše zna kako da pozicionira oko na fotografiji. Postoje i automatski sistemi koji uočavaju pravi položaj i pravi trenutak za fotografisanje u kome se na fotografiji nalaze svi neophodni elementi za dalju obradu elemenata irisa. Snimak oka je prikazan na slici 11.

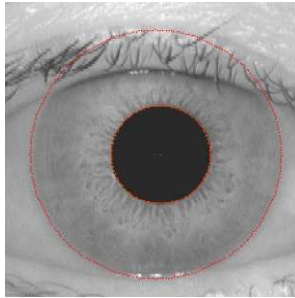
Matematički aparat za obradu irisa je složen koji ovde nije predmet proučavanja jer to nije tema ovoga rada. Proces će biti opisan na pojednostavljen način [18].

Nakon fotografisanja vrši se uočavanje lokacije irisa na snimku kako je to prikazano na slici 12. Sledeći korak je određivanje centra zenice koji nije istovremeno i centar irisa već se nalazi bliže nosu. Zatim se iris mapira polarnim koordinatama kako bi se dobio pravougaonik koji obuhvata oko $3/5$ irisa kako bi se eliminisale trepavice koje predstavljaju smetnju.

Tačnije, radi se o pseudo polarnim kooordinatama jer, kako je napomenuto, zenica se ne nalzi tačno u centru irisa. Izgled dobijenog pravougaonika - normalizovane slike irisa je dat na slici 13.



Slika 11. Snimak oka

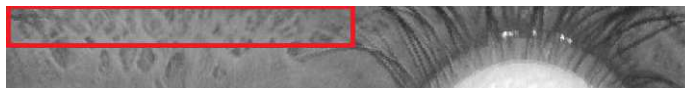


Slika 12. Uočavanje lokacije irisa



Slika 13. Pravougaona - normalizovana slika irisa

Ukoliko se potupi na opisani način, desni deo slike obično sadrži trepavice, pa se zato koristi levi deo slike, tačnije gornji levi deo slike pošto na sebi sadrži više karakterističnih informacija nego donji levi deo normalizovane slike. U opisanoj oblasti se definiše novi pravougaoni region kako je to prikazano na slici 14.



Slika 14. Pravougaona oblast koja se koristi za upoređivanje

Cilj ovog dela teksta je da informativno prikaže način korišćenja irisa u svrhu prepoznavanja osobe, dok se detalji, kao i matematički aparat koji obrađuje ovu oblast mogu pronaći u literaturi.

Lik (2D, 3D)

Prema statistikama, najviše korišćeni biometrijski podaci su otisak prsta i iris. Međutim korišćenje ova dva biometrijska podatka ima svoje nedostatke. Identifikacija skeniranjem irisa je veoma precizna, ali je istovremeno i veoma skupa za implementaciju i još uvek nije široko prihvaćena. Otisci prstiju su pouzdani i njihovo uzimanje je neinvazivno ali zahtevaju saradnju osobe kojoj se otisci uzimaju.

Prepoznavanje lika predstavlja dobar kompromis između pouzdanosti, društvene prihvatljivosti (osim u nekim arapskim zemljama) i zaštite privatnosti. Potrebno je napomenuti da se organizacije koje se bore za ljudska prava protive neovlašćenom snimanju lica osoba kako na javnim mestima tako i na zaštićenim lokacijama. Imajući u vidu da su kamere za nadzor postavljene u velikom broju u urbanim sredinama nemoguće je zaštititi

privatnost u tom smislu.

Potrebno je konstatovati da sistemi za identifikaciju osoba prepoznavanjem lika obezbeđuju niži stepen bezbednosti ali zato rade u uslovima velike frekvencije osoba i za njihov rad nije potrebna saradnja osoba čija se identifikacija vrši.

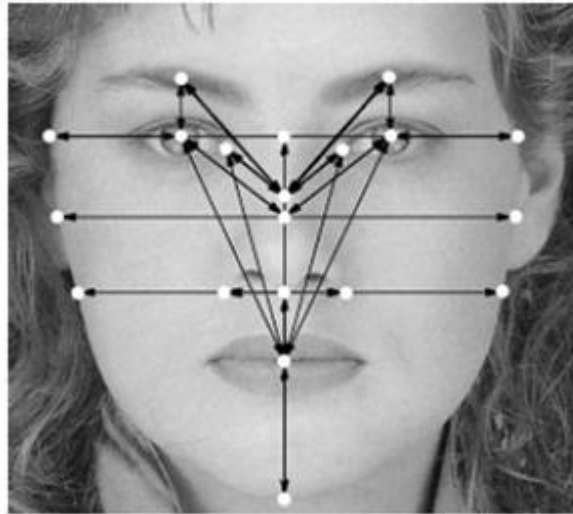
Kao i kod otiska prsta, sistemi za prepoznavanje lika se mogu podeliti u dve kategorije i to u sisteme za verifikaciju i sisteme za identifikaciju. Svi navedeni postulati koji su navedeni za prepoznavanje korišćenjem otiska prsta važe i za prepoznavanje korišćenjem lika.

U poslednjih desetak godina postignut je veliki napredak u oblasti prepoznavanja lika. U mnogim sistemima postignuta je tačnost prepoznavanja lika i do 90%. [19]. Ovi podaci se odnose na idealne uslove, dok se u realnim uslovima taj procenat smanjuje. Postoji pet ključnih faktora koji mogu značajno uticati na kvalitet prepoznavanja lika:

- Osvetljenje lika je bitan faktor. Različiti uglovi osvetljenja stvaraju različite senke što može izazvati konfuziju. Različiti intenzitet i vrsta osvetljenja takođe mogu uticati na stepen prepoznavanja i to kako zbog refleksije kože lica tako i zbog osetljivosti fotoćelija u samom foto aparatu. Neke metode 2D prepoznavanja su veoma osetljive na promene stepena i ugla osvetljenosti.
- Položaj lika u odnosu na foto aparat kao i ugao pod kojim se lik posmatra takođe mogu značajno da utiču na proces prepoznavanja. Ipak, primenom odgovarajućih softverskih i hardverskih sistema moguće je prevazići ovaj problem, odnosno rekonstruisati delove lika koji se ne vide iz određenog ugla. Savremeni algoritmi su relativno otporni na relativni položaj lika u odnosu na foto aparat.
- Vreme, odnosno izgled lika tokom starenje osobe je relativno teško predvideti. Iako su razvijeni efikasni alati za pomenutu namenu, i dalje je izgled osobe nakon određenog broja godina nepouzvano predvideti.
- Delimično prekriveni lik može znatno otežati proces prepoznavanja. Ako deo lika nije vidljiv usled zaklonjenosti iza neprovidne prepreke, nošenja tamnih naočara i slično, efikasnost prepoznavanja naglo pada.

Prilikom identifikacije lika u 2D tehnologiji kreira se standardna fotografija lika koji se grafički predstavlja ravnoj površini. Nakon toga se uočavaju karakteristične tačke lika na osnovu kojih se crta mreža spajanjem učenih tačaka. Ukoliko se mreže dva lika podudaraju smatra se da se radi o istom liku, odnosno istoj osobi. Kao i kod otiska prsta odnosno učenih minucija, i kod karakterističnih tačaka lika, dopuštaju se odstupanja kojima se definiše prag prepoznavanja.

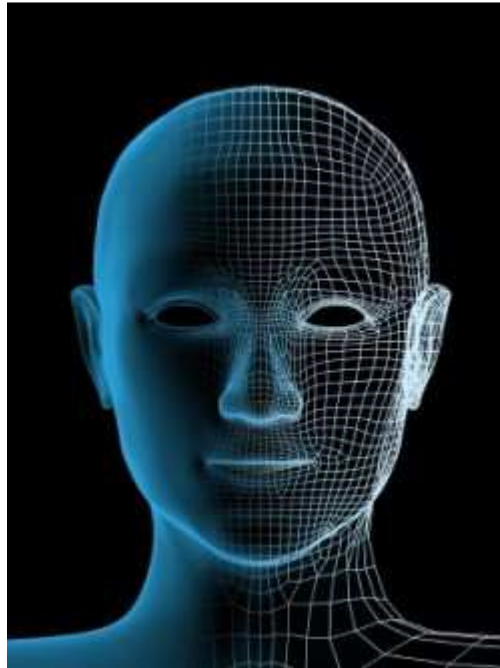
Jedna od mogućih šema karakterističnih tačaka prepoznavanja lika u 2D tehnologiji prikazana je na slici 15.



Slika 15. Karakteristične tačke koje se koriste za prepoznavanje lika u 2D tehnologiji

Uočeno je da karakteristične tačke zadržavaju svoj međusobni raspored relativno konstantnim tokom celog života. Prikazana mreža ostaje konstantna tokom godina i bez obzira da li osoba dobija ili gubi na težini.

Baze podataka 2D fotografija lika su istorijski posmatrano ranije nastale pa su iz tog razloga daleko obimnije i više koriste. Prepoznavanje lika korišćenjem 3D reprezenata lika je novijeg datuma tako da su i baze podataka siromašnije i zato se ova metoda manje koristi. Ipak 3D skeniranje lika, kreiranje odgovarajućih baza podataka i korišćenje 3D reprezenata lika za prepoznavanje osoba je u porastu. Princip 3D prepoznavanja se svodi na kreiranje 3D mreže koja opisuje lik. Svaka tačka mreže odgovara tački lika i opisuje se sa tri koordinate u Dekartovom pravouglom koordinatnom sistemu. Prepoznavanje se vrši upoređivanjem koordinata odgovarajućih tačaka. Prikaz 3D mreže lika je prikazan na slici 16.



Slika 16. 3D mreža lika

Skenirani lik može da sadrži od 5.000 do 100.000 tačaka u zavisnosti od tehnologije samog skenera. Očigledno je da je preciznost sa kojom se lik opisuje u 3D znatno veća nego što je to kod 2D lika. Samim tim je i proces prepoznavanja kompleksniji i rešava se specijalizovanim algoritmima [20].

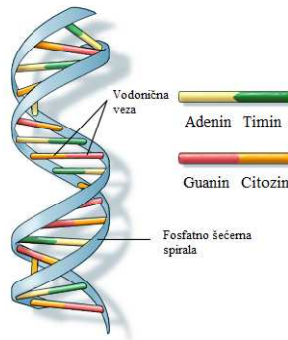
DNK - Dezoksiribonukleinska kiselina

DNK (*Dezoksiribonukleinska kiselina*) predstavlja mehanizam za prenošenje osobina sa roditelja na potomke. Većina DNK se nalazi u jedru ćelije. DNK se sastoji od dva lanaca nukleotida uvijenim u dvostroku spiralu koji su povezani vodoničnim vezama između komplementarnih baza i to pomoću četiri hemijske baze:

- adenin,
- guanin,
- citozin i
- timin.

Redosled nukleotida određuje individualne nasledne karakteristike. Ljudska DNK sadrži oko 3 milijarde baza i oko 99,7% DNK svih ljudi je identično. Redosled ovih baza u nizu definišu informacije koje određuju kako će se koja ćelija razvijati tokom prenatalnog razvoja kao i fizičke i mentalne osobine osobe nakon rođenja. „Govoreće“ podatke, odnosno podatke iz kojih se može dobiti neka informacija o samoj osobi, kao što su: rasa, pol, visina, boja očiju, nasledne bolesti itd. čini oko 0,2% DNK. Svega oko 0,1% DNK čine podaci koji ne daju

informacije o osobi. Zapravo, postoji mišljenje da i ovih 0,1% DNK daju informacije o osobi ali na ovom nivou saznanja se ne zna kakvi podaci iz ovog dela DNK se mogu dobiti. Iz tih razloga a zbog zaštite privatnosti i izbegavanja rasne i polne diskriminacije, za identifikaciju se koristi upravo ovih 0,1% DNK [21]. Grafička interpretacija DNK je data na slici 17.



Slika 17. Pojednostavljeni grafički prikaz DNK

Fosfatno-šećerna spirala čini strukturni okvir DNK. Pomenuta spirala se naizmenično sastoji od fosfatne grupe i šećera i ta spirala definiše usmerenja molekula .

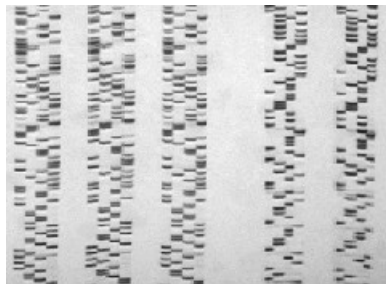
Ljudi imaju 23 para hromozoma koji sadrže svoju DNK šemu. Jedan član svakog hromozomskog para potiče od majke a drugi od oca. Dvadeset dva od ovih parova koji se nazivaju autozomi su isti i kod muškaraca i kod žena. Dvadeset i treći par su polni hromozomi se razlikuju između muškaraca i žena i taj hromozomski par određuje pol osobe. Žene imaju dve kopije X hromozoma, dok muškarci imaju jedan X i jedan Y hromozom.

Svaka ćelija u ljudskom telu sadrži kopiju ove DNK. Ljudski hromozom sadrži do 300 miliona osnovnih parova hemijskih baza. Imajući u vidu ovu činjenicu, može se zaključiti da se za svrhu identifikacije osobe koristi oko 3 miliona osnovnih parova.

Geni čine oko 5% ljudskog genoma, dok su preostalih 95% nekodirane sekvence. U nekodiranoj oblasti postoje identične sekvence DNK koje se mogu ponoviti više puta zaredom. Ove oblasti se nazivaju promenljivi broj parova ponavljanja (*variable number tandem repeats* - VNTRs). Broj parova koji se ponavljaju u određenoj oblasti regiona nazivaju se lokusima i njihov broj varira kod različitih osoba. Za bilo koji broj lokusa u VNTR u DNK pojedinca, može se odrediti tačan broj ponavljanja. Što je veći broj lokusa koji se analiziraju manja je verovatnoća da se pronađu dve različite osobe sa istim DNK profilom.

DNK profilisanje se vrši određivanjem broj ponavljanja u VNTR na nekoliko karakterističnih lokusa koji se koriste za kreiranje DNK profila pojedinca. Profil DNK se kreira tako što se najpre izoluje DNK iz tkiva, a zatim se podeli na kraće fragmente koje sadrže poznate VNTR oblasti. Poređenjem fragmenata dobijenih na opisani način utvrđuje se stepen sličnosti DNK odnosno identitet osobe. Na isti način se može utvrditi i stepen srodstva između dve osobe.

Primer poredjenja fragmenata DNK je prikazan na slici 18.



Slika 18. Upoređivanje segmenata DNK

U primeru prikazanom na slici 18 prva tri uzorka pripadaju jednoj osobi dok preostala dva uzorka pripadaju drugoj osobi [22].

Identifikacija osobe upoređivanjem DNK profila je 100% tačna. Postavlja se pitanje zbog čega se DNK ne koristi kao isključivi metod biometrijske identifikacije? Za to postoji više razloga. Osnovni razlog je nesavršenost tehnologije analize DNK. Oprema za analizu DNK je relativno skupa dok su analize komplikovane i dugotrajne. Opremativna upotreba provere identiteta putem DNK, na primer na graničnim prelazima, bila bi praktično nemoguća zbog dužine trajanja samog procesa.

Opšte je uverenje da će razvojem tehnologije, DNK postati dominantna biometrijska karakteristika za utvrđivanje identiteta ličnosti.

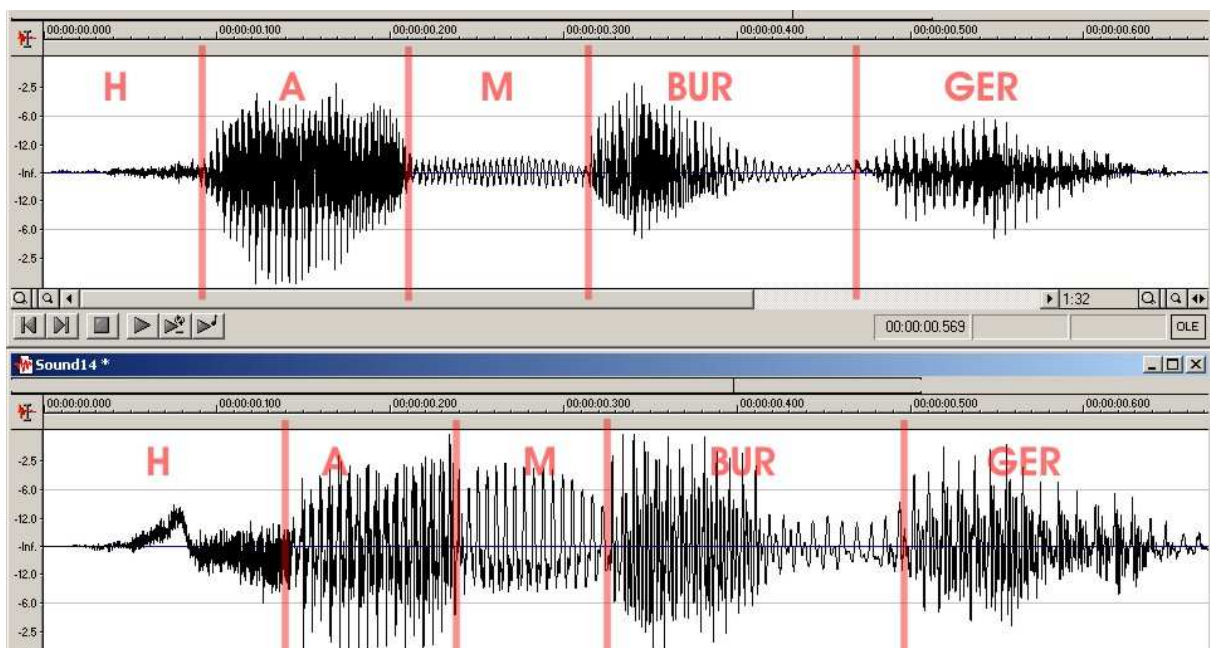
Prepoznavanje glasa

Glas je karakteristična biometrijska veličina koja se može koristiti za identifikaciju osobe. Prepoznavanje glasa podrazumeva proces snimanja izgovorene reči i prosleđivanje rezultata kao ulaza u softver za obradu glasa. Specijalizovani softver analizira frekvencu i amplitudu zvučnog talasa za svako karakteristično slovo i definiše osobu na osnovu dobijenih rezultata.

Najčešća metoda prepoznavanja glasa je formiranje templejta koji se smešta u bazu podataka a zatim se upoređuje sa templejtom glasa osobe čiji se identitet ispituje. Formiranje templejta se vrši tako što ispitanik ponavlja određena slova ili fraze koje se snimaju. Analogni podatak glasa se prevodi u digitalni oblik a zatim se digitalni podatak smešta u bazu podataka. Prevođenje signala glasa se iz analognog u digitalni oblik vrši putem mikrofona koji je u suštini analogno/digitalni (A/D) konvertor. Templejt sadrži osnovna slova i fraze koje se najčešće koriste. Očigledna je činjenica da se templejti za različite govorne jezike razlikuju po svom sadržaju.

Svaka osoba poseduje govorni aparat koji je specifičan i razlikuje se od govornog aparata druge osobe. Osnovni elementi govornog aparata su: pluća, grlo, nepce, usna duplja, nosna

duplja, jezik, zubi i usne. Očigledno je da će svaka anatomska specifičnost bilo kog od ovih elemenata dovesti do generisanja zvuka različitih karakteristika. Kako je navedeno, da bi se formirao templejt glasa, neophodno je da osoba najpre više puta izgovori određeni glas ili frazu, čiji se digitalni reprezent zatim smešta u bazu podataka. Ukoliko želimo da izvršimo poređenje, neophodno je da osoba ispitanik ponovi iste glasove ili fraze. Nakon toga se vrši upoređivanje dobijenog i prethodno sačuvanog templejta, što kao rezultat može dati odgovor da se glasovi poklapaju, odnosno da se radi o istoj osobi ili odgovor da se glasovi ne poklapaju, odnosno da se ne radi o istoj osobi. U opisanim, kontrolisanim uslovima tačnost uparivanja iznosi i do 98% [23]. U uslovima kada se upoređuju templejti koji ne sadrže predefinisane glasove ili fraze, što je čest slučaj u praktičnim primenama, efikasnost prepoznavanja glasa naglo pada. Primer upoređivanja digitalnog reprezenta, templejta glasova dve osobe, odnosno njihov grafički prikaz je prikazan na slici 19.



Slika 19. Grafički prikaz upoređivanja dva glasovna templejta

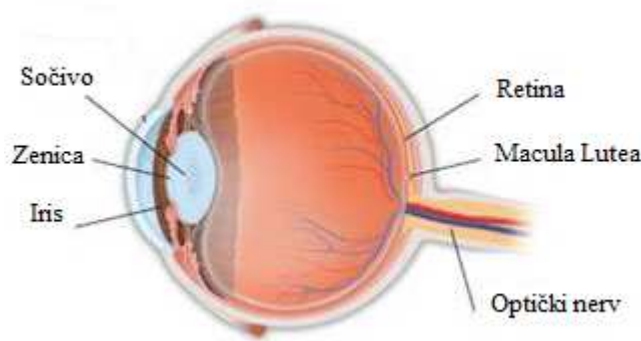
Od dve osobe je traženo da izgovore reč „hamburger“, kako bi se uporedili izgovoreni glasovi. Treba imati na umu da je eksperiment vršen u engleskom govornom području i da se slogovo „bur“ i „ger“ izgovaraju kao jedan glas. Na prikazanom dijagramu su razlike u frekvenci i amplitudi glasa kao i u trajanju izgovora pojedinih glasova očigledni.

Neophodno je napomenuti da je ljudski mozak, tokom evolucije, razvio algoritam prepoznavanja glasa do granica kojima se nauka još nije približila. Ljudski um je u stanju da nepogrešivo prepozna osobu po glasu čak i ukoliko osoba izgovara potpuno drugačije reči od uobičajenih, ukoliko zbog promuklosti ima drugu boju glasa ili se glas modulira na neki drugi način.

Dostizanje kvaliteta algoritma prepoznavanja glasa kojim se služi ljudski um predstavlja cilj kome nauka teži.

Biometrijska identifikacija mrežnjače

Retina ili mrežnjača je obloga unutrašnje površine oka koja je osetljiva na svetlost. Slika iz spoljašnjeg sveta, prolazi kroz sočivo oka koje se nalazi na zenici i formira se na površini retine [24]. Svetlost koja dopire do retine pokreće seriju elektro hemijskih reakcija što za posledicu ima generisanje električnog impulsa koji se prenosi kroz očni nerv do mozga. U mozgu se vrši obrada električnog signala u centrima za vid i stvara se doživljaj slike okolnog sveta. Pojednostavljeni prikaz građe oka je prikazan na slici 20.



Slika 20. Pojednostavljeni prikaz građe oka

Na retini se može uočiti čitav splet krvnih sudova koji se u anatomskom obliku razlikuje kod svake osobe.

Biometrijska identifikacija skeniranjem mrežnjače se koristi kako bi se definisao jedinstveni izgled retine osobe. Krvni sudovi unutar mrežnjače apsorbiraju svetlost brže od okolnog tkiva i lako ih je identifikovati korišćenjem odgovarajućeg osvetljenja. Skeniranje retine se vrši osvetljavanjem očnog dna svetlošću niskog intenziteta. Ovaj snop svetlosti prati standardizovani put skeniranja mrežnjače i konvertuje podatke u digitalni oblik. Podaci se zatim smeštaju u bazu podataka radi procesa identifikacije.

Intenzitet svetlosti je veoma bitan, pogotovo imajući u vidu osobine žute mrlje - *Macula Lutea*, koja se ne sme osvetliti prejakom svetlošću. Žuta mrlja sadrži fotosenzitivne ćelije i predstavlja tačku u kojoj se formira slika koja prolazi kroz sočivo oka. Svetlosne impulse konvertuje u električne signale i šalje ih kroz optički nerv do odgovarajućih centara u mozgu. Ukoliko bi se žuta mrlja osvetlila isuviše jakim svetlošću došlo bi do oštećenja fotosenzitivnih ćelija a time i do oštećenja ili gubitka vida.

Skeniranje mrežnjače [25] se najčešće koriste za autentifikaciju i identifikaciju. Skeniranje retine se koristi od strane nekoliko vladinih agencija u SAD kao što su: FBI, CIA i NASA . Ipak, u poslednjih nekoliko godina, skeniranje mrežnjače počinje da se primenjuje i u komercijalne svrhe.

Osim u primenama prepoznavanja osobe, skeniranje retine ima svoju primenu i u medicini. Koristi se za dijagnostikovanje i praćenje bolesti kao što su: HIV, sifilis, malarija, boginje i druge zarazne bolesti. Takođe se koristi i u dijagnostici hroničnih bolesti kao što su: srčane insuficijencije, arteriskleroza, povišeni holesterol itd.

Identifikacija vene na dlanu ili nadlanici

Proučavanjem se došlo do zaključka da je obrazac vena na dlanu ili nadlanici jedinstven za svaku osobu i ne menja se tokom vremena osim po veličini dlana ili nadlanice, što se pokazalo kao dobra osnova za prepoznavanje osobe [26]. Prepoznavanje na osnovu konfiguracije vena je relativno nova metoda. Koristi se u primenama gde je potreban visok nivo bezbednosti, kao što su policijski i vojni objekti, logovanje na informacione sisteme visoke bezbednosti čak i za identifikaciju na ATM (*Automatic Teller Machine*) terminalima za podizanje novca sa bankarskih računa. Pomenuta činjanica govori u prilog pouzdanosti tehnologije imajući u vidu da bankarski sistemi spadaju među najrestriktivnije sisteme koji se nalaze u širokoj upotrebi. Primena na ATM terminalima je naročito rasprostranjena u Japanu.

Primena identifikacije putem vena na dlanu nadlanici je relativno jednostavna. Dlan ili nadlanica se prinose uređaju koji je fotografiše - skenira svetlošću određene talasne dužine koja je bliska infracrvenoj svetlosti. Zaključujemo da se skeniranje vrši bez direktnog kontakta sa čitačem i ceo proces traje veoma kratko, poput skeniranja bar koda sa proizvoda na kasi u prodavnici. Nakon skeniranja konfiguracije vena, podaci se u digitalnom obliku smeštaju u bazu podataka. Prilikom sledećeg skeniranja podaci se upoređuju sa podacima koji se nalaze u bazi podataka. Postupak prepoznavanja je u potpunoj analogiji sa prepoznavanjem pri kome se koristi otisak prsta. Postoje mišljenja koja tvrde da je prepoznavanje osoba korišćenjem skeniranja vena pouzdanije nego što je to u slučaju korišćenja otiska prsta i da je obrazac vena praktično nemoguće falsifikovati. Pored brojnih novih biometrijskih karakteristika kao što su: DNK, prepoznavanje irisa... prepoznavanje vena sa svojim jedinstvenim karakteristikama i prednostima se pojavljuje kao jedna od najbrže rastućih biometrijskih tehnologija. Tehnologija prepoznavanja upoređivanjem spleta vena sve brže ulazi u komercijalnu primenu iako je do skoro bila samo eksperiment koji je sproveden u kontrolisanim laboratorijskim uslovima.

Jedan od razloga što prepoznavanje osobe korišćenjem konfiguracije vena ima odlične perspektive u budućnosti je jednostavnost njene primene. Više svetskih kompanija uspešno razvije tehnologiju za pomenutu namenu koje su razvile različite metode za primenu opisane tehnologije. U tu svrhu su razvijeni različiti uređaji za različite namene. Postoje čak i verzije koje se koristi na Android operativnim sistemima na mobilnim uređajima kao što su mobilni telefoni ili tablet računari.

naglasiti razliku između prepoznavanja napisanog teksta i razlikovanja rukopisa dve osobe. Alati za prepoznavanje teksta su u stanju da „pročitaju“ tekst i da ga protumače u smislu njegovog značenja. Pomenuti alati se nazivaju alatima za optičko prepoznavanje teksta (*Optical character recognition - OCR*). Prepoznavanje osobe na osnovu rukopisa ima za cilj raspoznavanje osobe na osnovu karakteristika teksta.

Raspoznavanje osobe na osnovu rukopisa se vrši u okviru dve kategorije analize rukopisa:

Statička analiza: korisnik piše na papiru i tekst se zatim digitalizuje – skenira korišćenjem optičkog skenera. Biometrijski sistem ne prepoznaje sam tekst i analizira njegov oblik. Pomenuta analiza se često naziva i „offline“ analiza [28].

Dinamička analiza proverava potpis tako što se uzima u obzir kako je potpis formiran. Korisnik piše na digitalizovanoj tabli čime se beleži kreiranje teksta u realnom vremenu. Ovom prilikom se beleži i dinamika samog pisanja, brzina, intenziteti pojedinih poteza, pritisak na podlogu i druge dinamičke karakteristike. Samo osoba koja je kreirala originalni tekst može na isti način, istom brzinom i istim pritiskom na podlogu formirati identičan tekst. Potrebno je uzeti u obzir i činjenicu da fizičko stanje osobe kao emotivni aspekt mogu uticati kako na dinamiku tako i na izgled napisanog teksta. Sasvim je moguće da ista osoba napiše isti tekst na drugačiji način u zavisnosti od trenutnog fizičkog i emotivnog stanja. Štaviše, posmatrajući napisani tekst istog autora pisan u različitim trenucima, moguće je odrediti fizičko i psihičko stanje osobe koje je napisala tekst. Uprkos varijacijama u dinamici pisanja svaka osoba piše tekst uvek po istom, prepoznatljivom obrascu što rukopis čini pogodnim za biometrijsku identifikaciju. Dobar falsifikator može da relativno verno reprodukuje sam izgled teksta dok je samu dinamiku pisanja veoma teško falsifikovati. Dinamička analiza se često naziva i „online“ analiza. Dinamika pisanja je biometrijska tehnologija koja se koristi za identifikaciju osobe na osnovu rukopisa.

Ne treba zaboraviti mnogobrojne eksperte koji su u stanju da na osnovu napisanog teksta na papiru rekonstruišu dinamiku potpisa pažljivim posmatranjem otiska olovke na papiru. Takvi stručnjaci – grafolozi su ljudi sa ogromnim iskustvom i koji su razvili nepisane algoritme, predstavljaju dragoceni izvor informacija za kreiranje nekih novih algoritama. Pitanje je da li će ljudi koji izučavaju identifikaciju korišćenjem svojeručnog potpisa to iskoristiti.

Osnovna prednost sistema biometrijske identifikacije posmatranjem rukopisa, u odnosu na druge vrste biometrijskih tehnologija, je ta što se rukopis već dugo koristi za identifikaciju i potvrđivanje saglasnosti na dokumentima. Ljudi lakše prihvataju već postojeće metode što smanjuje otpor prihvatanju novih metoda biometrijske identifikacije. Prednosti i nedostaci biometrijske identifikacije upoređivanjem rukopisa bi se ukratko mogli predstaviti na sledeći način:

Prednosti:

- Tehnologija se prihvata intuitivno, odnosno čovečiji um relativno lako raspoznaje različite rukopise,
- Veoma je teško falsifikovati dinamiku rukopisa,

- Rukopis je već dugo, istorijski posmatrano, prisutan kao metod identifikacije te je postao uobičajen i već je prohvaćen od strane populacije.

Nedostaci:

- Za dinamičku analizu je potrebna relativno skupa oprema,
- Postoje varijacije u načinu pisanja iste osobe.

Pored navedenih biometrijskih karakteristika koje se koriste u cilju biometrijske identifikacije postoje i druge biometrijske karakteristike koje se mogu koristiti u te svrhe. Neke od tih karakteristika su:

- Geometrija šake,
- Otisak dlana,
- Dinamika kucanja na računarskoj tastaturi,
- Otisak tabana,
- Oblik vilice odnosno raspored zuba u vilici,
- Geometrija ušne školjke,
- ...

Neophodno je pomenuti i mogućnost korišćenja više biometrijskih karakteristika za biometrijsku identifikaciju, istovremeno. Ovakav način biometrijske identifikacije se naziva Multimodalnom biometrijskom identifikacijom.

2.1.3 Proces biometrijske identifikacije

Proces biometrijske identifikacije u ovom poglavlju biće opisan na primeru otiska prsta. Problemi i predložena rešenja koji se razmatraju u ovom radu se baziraju na otisku prsta kao biometrijskoj karakteristici.

Sistemi za prepoznavanje otiska prsta igraju ključnu ulogu u mnogim situacijama u kojima posmatrana osoba potvrđuje svoj identitet. Otisak prsta predstavlja orijentisanu teksturu koja se sastoji od naizmenično raspoređenih grebena i dolina. Za raspored minucija, koje su opisane u prethodnom delu teksta, osnovano se pretpostavlja da su jedinstvene za svaki prst. Zbog pomenutih osobina jedinstvenosti, minucije su postale najrasprostranjeniji reprezent koji se koristi za formiranje reprezenta otiska prsta. Jedna od osnovnih postavki u vezi minucija je da na osnovu poznavanja rasporeda minucija nije moguće rekonstruisati kompletan otisak prsta.

Tokom poslednjih decenije predloženi su različiti algoritmi za utvrđivanje podudarnosti minucija i na osnovu toga, utvrđivanje identiteta osobe. Većina ovih algoritama se bazira na pretpostavci da su polazne slike otisaka, dobijene skeniranjem približno iste veličine. Uobičajena veličina skeniranog otiska je 1,3cm x 1,8cm. Međutim, razvojem tehnologije

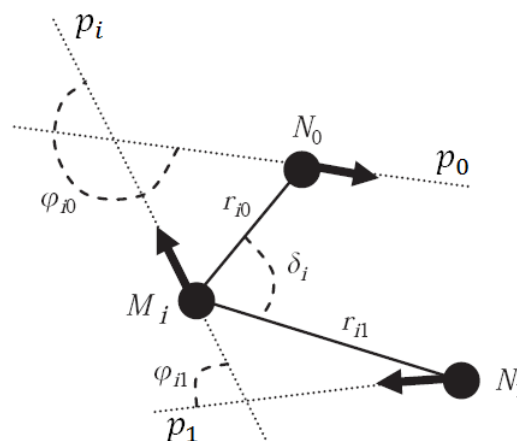
došlo je do minijaturizacije skenera otiska prsta a samim tim i do skeniranja pojedinih regija otiska prsta. Manje regije koje se skeniraju mogu biti i manje od navedenih dimenzija.

Ukoliko je otisak prsta nekompletan odnosno ukoliko posedujemo sliku samo određene manje regije može se javiti nekoliko vrsta problema:

- gubitak pojedinih minucija,
- smanjenje mogućnosti uparivanja templejta,
- izmena otiska prsta usled distorzije zbog karakteristika ljudske kože.

Upoređivanje otisaka prstiju zasnovano na minucijama kao rezultat daje broj uparenih minucija i na taj način se određuje stepen sličnosti dva otiska. Na osnovu stepena sličnosti određujemo da li se radi o istoj osobi ili je reč o različitim osobama. U praktičnim, forenzičkim pravilima se smatra da ukoliko se dva otiska podudaraju sa 12 minucija, pouzdano se smatra da se radi o istoj osobi.

Za definisanje međusobnog položaja minucija može se koristiti rastojanje minucija kao i ugao koji zaklapaju pravci tangenti na bregove otiska u tačkama minucija, kako je to prikazano na slici 22.



Slika 22. Definisanje međusobnog položaja minucija

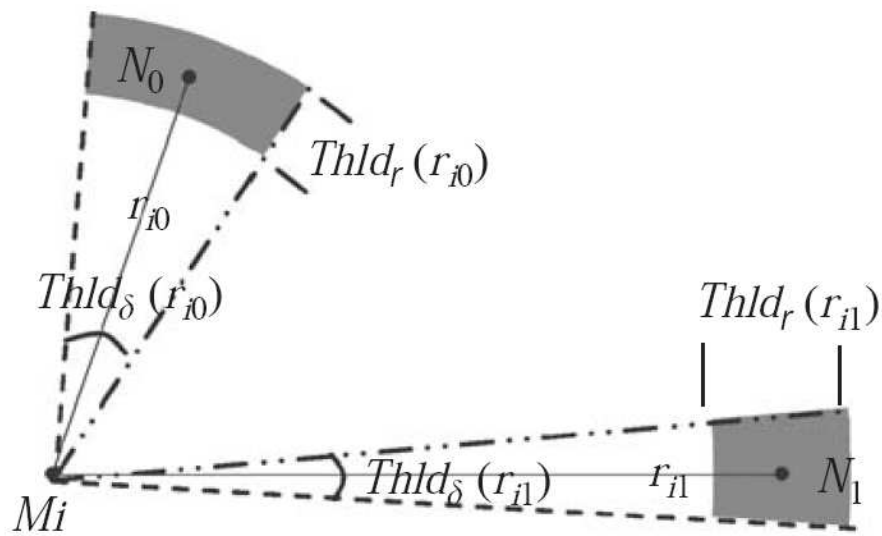
Kao što je ranije opisano, međusobni položaj minucija nije uvek istovetan iz više razloga. Jedan od razloga je elastičnost tkiva jagodice prsta. Usled nejednakog pritiska prilikom različitih merenja dolazi do različite deformacije tkiva i samim tim i do različitog međusobnog rasporeda minucija. Pored toga moguće je da neke minucije nisu vidljive na skeniranom otisku, usled lošeg kvaliteta skenirane slike. Dalju otežavajuću okolnost za prepoznavanje otiska može predstavljati i činjenica da templejt koji je arhiviran u bazi podataka, ne sadrži sve minucije zbog lošeg kvaliteta skeniranog otiska na osnovu koga je templejt formiran.

Na slici 22 su uočene tri minucije: M_i kao osnovna uočena minucija kao i njoj dve najbliže susedne minucije N_0 i N_1 [29]. Minucija M_i je definisana sa tri parametra, kako je to navedeno u prethodnom tekstu, a to su prvougle Dekartova pravougle koordinate x_i, y_i kao i ugao θ_i koji predstavlja ugao između tangente na breg u tački minucije i pozitivnog dela x ose, kako je to prikazano na slici 6. Na isti način su definisane i pozicije minucije N_0 koordinatama $(x_{n0}, y_{n0}, \theta_{n0})$ i minucije N_1 koordinatama $(x_{n1}, y_{n1}, \theta_{n1})$. Sada konstruišemo dve duži i to prvu između tačaka M_i i N_0 i nazovemo je r_{i0} a drugu duž između tačaka M_i i N_1 i nazovemo je r_{i1} . Očigledno je da r_{i0} i r_{i1} predstavljaju Euklidsko rastojanje između minucija M_i i N_0 kao i M_i i N_1 , respektivno. Opisane karakteristike minucije: $M_i, N_0, N_1, r_{i0}, r_{i1}$ se mogu posmatrati kao pet elemenata nekog vektora. S obzirom da su posmatrane velične osnovne ili primarne karakteristike minucija, i pomenuti vektor se može nazvati vektorom osnovnih ili primarnih karakteristika. Korišćenje trougla kao osnovne jedinice mreže minucija će biti detaljnije obrađeno u poglavlju u kome će biti detaljno opisana primena Deloneove triangulacije na otisak prsta. Primarni vektor daje osnovnu gradivnu jedinicu mreže otiska prsta. Nacrtajmo sada prave: p_i koja sadrži pravac tangente u tački M_i , p_0 koja sadrži pravac tangente u tački N_0 i pravu p_1 koja sadrži pravac tangente u tački N_1 . Posmatrajmo nacrtane prave kao orijentisane prave i to u smeru orijentacije minucija. Smer orijentacije minucije se određuje na način kako je to prikazano na slici 6. Smerovi orijentacije minucija u posmatranom slučaju su prikazani na slici 26. Označimo sada uglove između nacrtanih pravih i to: δ_i konveksan ugao između duži r_{i0} i r_{i1} , φ_{i0} ugao između pozitivnih pravaca tangenti p_i i p_0 i ugao φ_{i1} između pozitivnih pravaca tangenti p_i i p_1 . Sada se može kreirati vektor sekundarnih karakteristika $S_i(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$. Potrebno je primetiti da su N_0 i N_1 najbliže tačke minuciji M_i i da su označene ne po Euklidskom rastojanju već tako da zadovoljavaju jednačinu:

$$\frac{\overrightarrow{N_0 M_i}}{|N_0 M_i|} \times \frac{\overrightarrow{N_1 M_i}}{|N_1 M_i|} \geq 0 \quad (3)$$

N_0 je prva a N_1 je druga minucija prilikom kreiranja ugla $< N_0 M_i N_1$.

Izobličenja papila su neizbežna prilikom prevođenja trodimenzione jagodice prsta u dvodimenzionu sliku. Na to utiču faktori koji su opisani u prethodnom tekstu. Što su rastojanja r_{i0} i r_{i1} veća, može se primetiti da vektor sekundarnih karakteristika $S_i(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$ pokazuje znatnija odstupanja uglova $\varphi_{i0}, \varphi_{i1}$ i δ_i . Može se zaključiti da mala lokalna izobličenja mogu dovesti do znatnog ukupnog izobličenja slike samog otiska prsta. Dalje se može zaključiti da su odstupanja rastojanja manja ukoliko su rastojanja r_{i0} i r_{i1} manja. Sa druge strane, odstupanja uglova su veća što su rastojanja r_{i0} i r_{i1} manja. Imajući u vidu prethodno razmatranje bilo bi korisno definisati zonu tolerancije imajući u vidu vrednosti r_{i0} i r_{i1} .



Slika 23. Grafički prikaz zona tolerancije

Oblast tolerancije predstavljena osenčenom površinom na slici 23. Pretpostavka je da se skenirana minucija može naći u prikazanoj oblasti i da se pri tome njena lokacija može smatrati korektnom, odnosno može se smatrati da se lokacija posmatrane minucije podudara sa lokacijom originalne minucije. Očigledno je da se veličina oblasti tolerancije može podešavati. Ukoliko je oblast tolerancije veća, veća je i verovatnoća da će pozicija posmatrane minucije biti prihvaćena kao validna i obrnuto, ukoliko je oblast tolerancije manja, manja je i verovatnoća da će pozicija posmatrane minucije biti prihvaćena kao validna.

Veličina oblasti tolerancije se može definisati sledećim funkcijama i to:

- funkcije za određivanje pragova rastojanja $Thld_r(r_{i0})$ i $Thld_r(r_{i1})$
- funkcije za određivanje pragova uglova: $Thld_\delta(r_{i0})$ i $Thld_\delta(r_{i1})$

Funkcije $Thld_r(r_{i0})$ i $Thld_r(r_{i1})$ određuju manju oblast tolerancije ukoliko su r_{i0} i r_{i1} manji i obrnuto, funkcije $Thld_r(r_{i0})$ i $Thld_r(r_{i1})$ definišu širu oblast ukoliko su r_{i0} i r_{i1} veći.

Funkcije $Thld_\delta(r_{i0})$ i $Thld_\delta(r_{i1})$ određuju širu oblast ukoliko su r_{i0} i r_{i1} manji i obrnuto, funkcije $Thld_\delta(r_{i0})$ i $Thld_\delta(r_{i1})$ određuju manju oblast ukoliko su r_{i0} i r_{i1} veći.

Pomenute funkcije se koriste za definisanje tolerancije susednih minucija u odnosu na posmatranu minuciju.

U posmatranom modelu, pomenute funkcije referišu na normalizovana rastojanja. Faktori normalizacije rastojanja kao i faktori normalizacije uglova zavise od rastojanja r_{i0} i r_{i1} susednih minucija od posmatrane minucije. Faktori normalizacije rastojanja rastu sa povećanjem r_{i0} i r_{i1} , dok faktori normalizacije uglova opadaju sa povećanjem r_{i0} i r_{i1} .

Normalizovane karakteristike nam omogućavaju da obrađujemo različite vrste rastojanja što direktno utiče na oblast tolerancije.

Moguće je koristiti više različitih načina određivanja stepena podudarnosti u zavisnosti od broja minucija u otisku koji se ispituje (I) i od broja minucija referentnog otiska (R). Pun otisak podrazumeva sliku čije su dimezije oko 1,3 cm x 1,8 cm što najčešće omogućava da je broj uočenih minucija veći od unapred utvrđenog broja minucija - α .

Mogu se uočiti tri slučaja prilikom utvrđivanja podudarnosti dva otiska:

1. Broj uočenih minucija na I i R je manji od α ,
2. Jedan od otisaka I ili R , sadrži broj minucija manji od α ,
3. Oba otiska, I i R , sadrže broj minucija veći od α .

U prva dva slučaja kod bar jednog otiska prsta I ili R , uočava se manje minucija od graničnog broja minucija α . Ukoliko je otisak nekompletna, pronalaženje dve najbliže centralne minucije u cilju kreiranja sekundarnog vektora predstavlja problem. U takvim slučajevima, se vrši uparivanje brute-force metodom. Ispituju se sve moguće opcije i pronalazi se rešenje sa najviše podudarnosti između otiska koji se ispituje (I) i referentnog otiska (R). Primena ove prilike upoređivanja dva otiska dugo traje. Koristi se kada postoji mali broj minucija i kada su dostupni samo fragmenti otisaka koji se upoređuju. Ukoliko otisci koji se upoređuju sadrže više od α minucija, za određivanje podudarnosti se koriste sekundarni vektori. Na taj način se povećava brzina upoređivanja uz istovremeno povećanje preciznosti.

Metoda brute-force uparivanja radi neposredno sa podacima svake minucije. Posmatra se svaka minucija $p_i(x_i, y_i, \theta_i)$ sa otiska I i $q_j(x_j, y_j, \theta_j)$ sa otiska R . Biraju se dve referentne tačke p_i i q_j za koje se pretpostavlja da su uparene. Nakon toga se posmatraju ostale dostupne minucije i upoređuju se njihove polarne koordinate u odnosu na izabrane tačke p_i i q_j . Ukoliko se minucija $m_i(x_i, y_i, \theta_i)$ predstavi u polarnim koordinatama dobija oblik $(r_{i,k}, \Phi_{i,k}, \theta_{i,k})$ u odnosu na referentnu tačku $m_k(x_k, y_k, \theta_k)$ pri čemu je $\theta_{i,k}$ ugao između θ_i i θ_k . Za minucije $p'_i(r_{i,i'}, \Phi_{i,i'}, \theta_{i,i'})$ i $q'_j(r_{j,j'}, \Phi_{j,j'}, \theta_{j,j'})$ može se konstatovati da su uparene ukoliko je q_j i oblasti tolerancije koje je formirana okolo tačke p_i [30].

Postupak se ponavlja za sve kombinacije izbora tačaka sve dok se ne dobije uparenost svih tačaka i u tom slučaju se dva otiska proglašavaju jednakim. Ukoliko se ne dobije ni jedan slučaj poklapanja dostupnih minucija, dva otiska se proglašavaju različitim.

Kada I i R imaju dovoljno veliki broj minucija koji je veći od α , brute-force metoda bi trošila isuviše mnogo računarskih resursa. U takvim slučajevima se uparivanje otisaka vrši upoređivanjem sekundarnih karakteristika. Imajući u vidu da su sekundarne karakteristike lokalizovane i invarijantne na rotaciju, u tom slučaju se mora vršiti usklađivanje orijentacije dva otiska. Neka su $S_i(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$ i $S_j(r_{j0}, r_{j1}, \varphi_{j0}, \varphi_{j1}, \delta_j)$ sekundarne osobine otiska koji se ispituje i referentnog otiska, respektivno. S_i je podudarno sa S_j ukoliko je zadovoljeni sledeći uslovi:

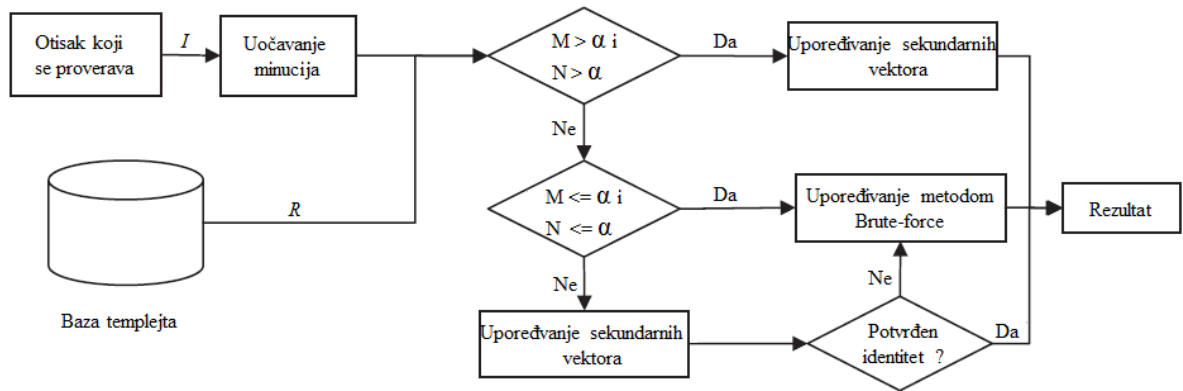
$$|r_{i0} - r_{j0}| \leq Thld_r(r_{i0}) \quad (4)$$

$$|r_{i1} - r_{j1}| \leq Thld_r(r_{i1}) \quad (5)$$

$$|\varphi_{i0} - \varphi_{i1}| \leq Thld_\theta(\varphi_{i0}) \quad (6)$$

$$|\delta_i - \delta_j| \leq Thld_\delta(\delta_i) \quad (7)$$

Šematski prikaz opisanog postupka dat je na slici 24.



Slika 24. Šematski prikaz procesa upoređivanja dva otiska prsta

Ukoliko konačnu odluku o identičnosti dva otiska prsta donosi čovek, korisno je da se pridržava forenzičkih preporuka da je za potvrdu identiteta potrebno da bude upareno najmanje 12 minucija. Ako se prepoznavanje otiska vrši automatski, navedeni princip nije primenjiv. Razlog je taj što čovek ima percepciju i kompletne slike otiska prsta a ne samo uočenih minucija dok se automatski sistem prepoznavanja bazira isključivo na uočenim minucijama.

Izračunavanje stepena sličnosti dva otiska prsta na klasičan način zasnovan na upoređivanju međusobnog položaja minucija vrši se na sledeći način:

$$\frac{n^2}{velicina_I - velicina_R} \quad (8)$$

Gde $velicina_I$ i $velicina_R$ predstavljaju broj minucija na otisku koji se proverava i na referentnom otisku, respektivno, dok je n broj uparenih minucija.

Bazen and Gerez [31] tvrde da pokazani način izračunavanja stepena sličnosti dva otiska daje dobre rezultate. Postoje različita mišljenja koja tvrde da je ovaj metod nepouzdan pogotovo ukoliko se posmatraju dva otiska prsta različite veličine.

Deo koda pisan u C# programskom jeziku za izračunavanje stepena sličnosti je predstavljen u daljem tekstu:

```
double visina; // visina posmatranog segmenta
double sirina; // sirina posmatranog segmenta
double max_visina; // maksimalna dozvoljena visina
double max_sirina; // maksimalna dozvoljena sirina
int O_I; // broj sekundarnih osobina na ispitivanom otisku u oblasti
koje se peklapa sa referentnim otiskom
int O_R; // broj sekundarnih osobina na referentnom otisku u oblasti
koja se peklapa sa ispitivanim otiskom
double S_prosek; // prosečna uparenost svakog sekundarnog vektora
int prag; // granicna vrednost broja minucija
if (visina > max_visina || sirina > max_sirina)
{
    stepen_slicnosti = 0;
}
else
{
    if (O_I < 5)
    {
        O_I = 5;
    }
    if (O_R < 5)
    {
        O_R = 5;
    }
    if (n >= prag & n > 3/5 O_I & n > 3/5 O_R)
    {
        stepen_slicnosti = S_prosek;
    }
    else
    {
        stepen_slicnosti = n*n*S_prosek / ( O_I * O_R);
        if (stepen_slicnosti > 1.0)
        {
            stepen_slicnosti = 1.0;
        }
    }
}
```

Pored opisanog metoda postoje i drugi, različiti pristupi ovoj problematici koji su obrađeni u literaturi. Treba imati na umu da ne postoje apsolutno tačni ili netačni sistemi već se radi o različitim pristupima, a samim tim i oblastima primenljivosti kao i o stepenu efikasnosti.

2.1.4 Zaštita privatnosti

U prethodnom tekstu su već pomenuti problemi zaštite privatnosti, kako u digitalnom tako i u realnom svetu. Razvojem računarskih mreža i Interneta kao globalne mreže, zaštita privatnosti postaje sve veći izazov. Kao što je pomenuto, kompromitovanje biometrijske veličine predstavlja nenadoknativ i trajan gubitak. Pre nego što budu razmotreni neki od

metoda zaštite biometrijskih veličina, neophodno je navesti neke od elemenata Fazi logike, koji će biti korišćeni u daljim razmatranjima.

2.1.4.1 Elementi Fazi logike

Koncept fazi logike je osmislio Lotfi Zadeh (*Lotfali Askar Zadeh, 1921*), profesor na Berkli Univerzitetu u Kaliforniji. Tehnologiju je predstavio kao način obrade podataka sa jednom od osnovnih ideja da se dozvoljava delimična pripadnost skupu umesto jednoznačne tvrdnje da li je neki element član skupa ili ne. Kasnije se pokazalo da su mogućnosti ovakvog pristupa teoriji skupova veoma pogodno za primenu u adaptivnom upravljanju. Ovakav pristup teoriji skupova nije bilo moguće primeniti sve dok savremeni računari nisu dostigli kapacitete neophodne za fazi obradu podataka [31].

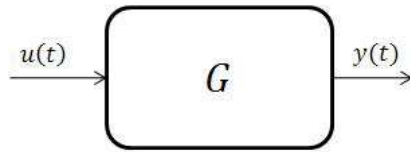
Adaptivno upravljanje je takav način upravljanja sistemom u kome se kontrolisani sistem prilagođava parametrima koji utiču na sistem a koji variraju i nisu potpuno poznati u početnom trenutku funkcionisanja sistema. Adaptivna kontrola se razlikuje od jednoznačne kontrole u smislu da nisu potrebne prethodne informacije o granicama neizvesnih ili vremenski promenljivih parametara. Za razliku od adaptivne, jednoznačna ili robusna kontrola garantuje da, ako su promene parametara u okviru datih granica, zakon kontrole ne treba menjati, dok se adaptivno upravljanje bavi kontrolom zakona promene.

Zadeh je primetio da ljudima nije potreban precizan, numerički podatak, kako bi doneli ispravnu odluku prilikom upravljanja sistemom. Pored toga, ljudi su u stanju da vrlo efikasno adaptivno upravljaju sistemom na koji utiču slučajno promenljivi parametri. Zadeh je takođe zaključio da povratne informacije sistema mogu biti neprecizne i nejasne, i da ljudi uprkos tome mogu veoma efikasno upravljati sistemom. U računarskom smislu, prijemnici povratnih informacija mogu biti programirani tako da prihvataju neprecizne podatke i podatke sa šumom. Ovakvi sistemi mogu bolje upravljati sistemom na koji utiču stohastičke promenljive dok je istovremeno njihova implementacija znatno jednostavnija.

Fazi logika omogućava da se na jednostavan način dođe do zaključka na osnovu nejasnih, nepreciznih, dvosmislenih i nekompletnih podataka sa šumom.

Istovremeno je uočeno da način ljudskog razmišljanja nudi rešenja i za veoma kompleksne sisteme. Prisup fazi logike je da imitira način zaključivanja i donošenja odluke koji se odvija u ljudskoj svesti, s tom razlikom što to računarski sistemi čine neuporedivo brže. Ipak, način funkcionisanja ljudskog mozga i dalje ostaje velika tajna a njegova kompleksnost prevazilazi i najsloženije računare.

Ukoliko posmatramo jednostavan sistem čiji ulaz predstavlja funkcija $u(t)$ a izlaz $y(t)$. Funkcija prenosa G se definiše kao odnos Laplasove (*Pierre-Simon Laplace, 1749 - 1827*) transformacije izlazne veličine $y(t)$ i ulazne veličine $u(t)$, uz pretpostavku da su svi početni uslovi nulti tj. da je $u(t) = y(t) \equiv 0 \quad \forall t < 0$.



Slika 25. Prenosna funkcija

Funkcija prenosa se, između ostalog, najviše koristi u digitalnoj obradi signala. Predstavlja diferencijalni operator, koji predstavlja odnos između ulaza i izlaza. Znajući ulaz sistema i funkciju prenosa, može se rekonstruisati izlazni signal. Drugim rečima, prenosna funkcija je matematički prikaz odnosa između ulaza i izlaza dinamičkog sistema. Za kontinualne signale, funkcija prenosa daje opšte informacije o sistemu, posebno informacije o njegovoj stabilnosti.

Naredna razmatranja važe samo za linearne stacionarne sisteme.

Linearnost podrazumeva da odnos između ulaza i izlaza sistema zadovoljava istu zakonitost, odnosno sistem mora da zadovoljava sledeći uslov:

Ako je ulazni signal $u(t) = Au_1(t) + Bu_2(t)$ tada izlazni signal mora biti $y(t) = Ay_1(t) + By_2(t)$.

Izlaz sistema $y_i(t)$ je odgovor na ulazni signal $u_i(t)$ za bilo koje konstante A i B .

Stacionarnost znači da je izlaz sistema, kao odgovor na bilo koji ulaz, istog vremenskog kašnjenja. Odnosno, vremensko kašnjenje izlaza biće uvek isto, u odnosu na ulazni signal, usled uticaja sistema.

Laplasova transformacija, iako je dobila ime u njegovu čast, jer je ovu transformaciju koristio u svom radu o teoriji verovatnoće, transformaciju je zapravo otkrio Leonard Ojler (*Leonhard Paul Euler*, 1707 - 1783), švajcarski matematičar. U matematici se Laplasova transformacija nalazi u širokoj upotrebi. Laplasova transformacija je integralna transformacija, koja datu funkciju $f(t)$ sa realnim argumentom t ($t \geq 0$) preslikava u funkciju $f(s)$ sa kompleksnim argumentom s . Laplasova transformacija se može iskazati sledećim obrascem:

$$L\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt \quad (s = \sigma + i\omega; \sigma > 0; t \geq 0) \quad (9)$$

Linearna diferencijalna jednačina koja opisuje linearan sistem sa jednim ulazom i jednim izlazom u opštem slučaju glasi:

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_2 \frac{d^2 y}{dt^2} + a_1 \frac{d^1 y}{dt^1} + a_0 y = \quad (10)$$

$$= b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + b_1 \frac{d^1 u}{dt^1} + b_0 u$$

Nakon primene Laplasove transformacije dobija se:

$$\begin{aligned} s^n Y + a_{n-1} s^{n-1} Y + \dots + a_2 s^2 Y + a_1 s Y + a_0 Y = \\ = b_m s^m U + b_{m-1} s^{m-1} U + \dots + b_2 s^2 U + b_1 s U + b_0 U \end{aligned} \quad (11)$$

Sada je funkcija prenosa:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_2 s^2 + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_2 s^2 + a_1 s + a_0} \quad (12)$$

Korisno je napomenuti nekoliko činjenica:

- Kod nestacionarnih sistema nije moguće primeniti Laplasovu transformaciju,
- Funkcija prenosa uzima u obzir samo zavisnost izlaza od ulaza i ne pruža informaciju o unutrašnjoj strukturi i ponašanju sistema,
- Važi samo za nulte početne uslove.

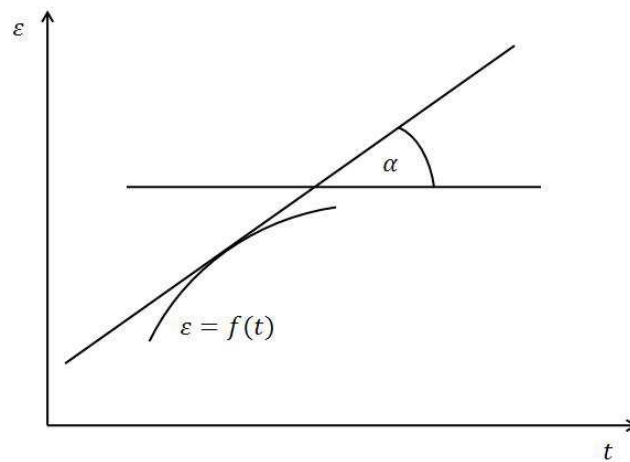
Nestacionarni sistemi, koji se nazivaju i vremenski promenljivi sistemi poseduju jedan ili više parametara koji su funkcije vremena - promenljivi parametri, i u tom slučaju primena Laplasove transformacije nije moguća.

Fazi logika se zasniva na jednostavnom pravilu:

$$IF X AND Y THEN Z \quad (13)$$

Drugim rečima, ako je zadovoljen uslov X i ako je zadovoljen uslov Y tada se izvršava akcija Z . Ovakav pristup je empirijski i koristi se umesto matematičkog modeliranja sistema.

Fazi logika zahteva određene numeričke parametre da bi sistem radio sa ulaznim parametrima koji u sebi sadrže ono što se smatra značajnom greškom kao i značajnom brzinom promene greške. Tačne vrednosti obično nisu kritične sve dok ne dođe do prevelikih odstupanja, u tom slučaju greške se određuju empirijski. Ukoliko se nacrtava kriva promene greške, njen nagib je definisan prvim izvodom funkcije u posmatranoj tački i na taj način se dobija brzina promene greške.



Slika 26. Grafička interpretacija brzine promene greške

Očigledno je da je brzina promene greške $\tan \alpha = \frac{d\varepsilon}{dt} = \dot{\varepsilon}$. Cilj fazi logike je zadržati brzinu promene greške a konačno i samu grešku u prihvatljivim granicama.

Fazi logika koristi neprecizan, ali veoma opisan jezik kako bi obrađivao ulazne podatke onako kako bi to činio čovek. Veoma je robustan u pogledu tolerancije ulaznih grešaka, za razliku od klasičnih sistema, iako nema saznanja o matematičkom modelu sistema. O robustnosti fazi sistema najbolje govori činjenica da fazi sistem najčešće radi prilikom prve instalacije uz minimalna podešavanja, ili čak i bez njih.

Motivi za korišćenje fazi logike se mogu sistematizovati u vidu sledećih stavki:

- To je robustan sistem koji ne zahteva precizan ulaz, signale bez šuma. Izlazna kontrolna funkcija je uvek glatka kriva uprkos najrazličitijim mogućim varijacijama ulaznih podataka. Glatke krive su funkcije koje imaju izvode svih redova.
- Fazi logika kao kontrolor procesa sadrži korisnički definisana pravila koja regulišu ciljno ponašanje sistema. Pravila se mogu lako menjati kako bi se poboljšale ili značajno izmenile performanse sistema.
- Fazi logika može da kontroliše nelinearne sisteme što je veoma teško ili čak nemoguće matematički modelovati.

Koncept lingvističkih ili jezičkih promenljivih predložio je Lotfi Zadeh. Ove promenljive se nazivaju još i fazi promenljive. O njima ne treba razmišljati kao o brojevima već kao o jezičkim objektima. Signal dobijen sa senzora se može posmatrati kao imenica, na primer: temperatura, osvetljenje, brzina, protok, pritisak... Pošto je greška razlika između zadatog i realnog ponašanja sistema i ona se može posmatrati na isti način. Vrednosti fazi promenljivih se mogu posmatrati kao pridevi koji modifikuju fazi promenljive. Na primer promenljiva greška se može opisati kao: mnogo pozitivna, malo pozitivna, nula, malo negativna i mnogo

negativna. Ukoliko se želi pojednostavljenje, svaka fazi promenljiva se može opisati kao: pozitivna, nula ili negativna. Dodatni pridevi se obično dodaju ukoliko se posmatraju nelinearni sistemi, dok je kod linearnih sistema dovoljan pojednostavljeni opis.

Polazimo od pretpostavke da se za grešku ponude tri moguće vrednosti: negativna, nula i pozitivna, a istovremeno se za brzinu promene greške ponude iste vrednosti: negativna, nula i pozitivna. Na osnovu pretpostavljenog može se formirati matrica dimenzija 3x3 sa ukupno 9 polja koja su označena brojevima od 1 do 9, kako je to prikazano u tabeli 1. Kolone predstavljaju vrednosti za grešku, i to: negativnu grešku, nultu grešku i pozitivnu grešku, posmatrajući sa leva u desno. Slično tome, redovi predstavljaju brzinu promene greške i to: negativna brzina promene greške, nulta brzina promene greške i pozitivna brzina promene greške, posmatrajući odozgo na dole. Planarna konstrukcija formirana na opisani način se naziva matrica pravila. Njen izgled je dat na slici 27. Očigledno je da matrica ima dva ulaza a to su greška i brzina promene greške dok je izlaz samo jedan, odnosno polje matrice koje se nalazi u preseku odgovarajućeg reda i odgovarajuće kolone. U posmatranom slučaju moguće je različitih logičkih proizvoda od kojih svaki daje jedan zaključak kao odgovor.

		ε		
		neg	nula	poz
$\dot{\varepsilon}$	neg	1	2	3
	nula	4	5	6
	poz	7	8	9

Tabela 2. Osnovni oblik matrice pravila

Matrice pravila najčešće imaju neparan broj redova i kolona kako bi nula zauzimala centralno mesto u matrici. Potrebno je napomenuti da to nije uslov i matrica pravila se formira za svaki specifičan slučaj. Ipak je pogodno imati nultu poziciju sve dok se funkcije sa obe strane centralne pozicije u matrici u dobroj meri preklapaju. Može se uočiti da centralna odnosno nulta pozicija korespondira sa izlazom po kome ne treba vršiti nikakve izmene jer je sistem stabilan. Ukoliko bi ova pozicija nedostajala, praktično bismo neprekidno tražili stabilno stanje. Pored toga, broj redova i kolona ne mora biti jednak, tj. matrica ne mora biti kvadratna. Različiti broj redova i kolona se najčešće primenjuje kada je potrebno više različitih tipova ulaznih parametara. Maksimalan broj mogućih pravila se izračunava kao proizvod broja redova i kolona. Potrebno je obratiti pažnju prilikom definisanja pojedinih pravila da li su zadati uslovi realni, tj. da li se mogu zaista ostvariti u realnom funkcionisanju sistema. Uslove koji nisu realno ostvarivi treba ukloniti iz matrice pravila pogodnim izborom promenljivih kako bi se problem pojednostavio. Osnovni cilj matrice pravila je definisanje

prostora mogućih ulaznih parametara koje je potrebno regulisati radi ostvarivanja stabilnosti rada posmatranog sistema.

Funkcije pripadnosti

Nakon što je definisana matrica pravila, bilo bi neophodno razmotriti kako se ova pravila primenjuju. Zato će u narednom tekstu biti razmotrene funkcije koje definišu pripadnost skupu ili funkcije pripadnosti.

Funkcija pripadnosti predstavlja grafički prikaz visine učešća svakog od ulaznih parametara na izlaz sistema. Funkcije pridružuju ponderisano učešće svakom od ulaznih parametara. Nakon toga se definiše međusobno preklapanje funkcija i na osnovu toga se određuje odgovor sistema. Izlazni, fazi skup zaključaka je determinisan pravilima definisanim i prikazanim u matrici pravila. Pored toga na izlazne parametre utiče i stepen učešća svakog od ulaznih parametara. Jednom kada se funkcije definišu, usklade i kombinuju, predstavljaju se u obliku koji upravlja sistemom. Postoje različite funkcije pripadnosti za svaki ulazni i izlazni parametar.

Neki od osobina funkcija pripadnosti koje je potrebno pomenuti su:

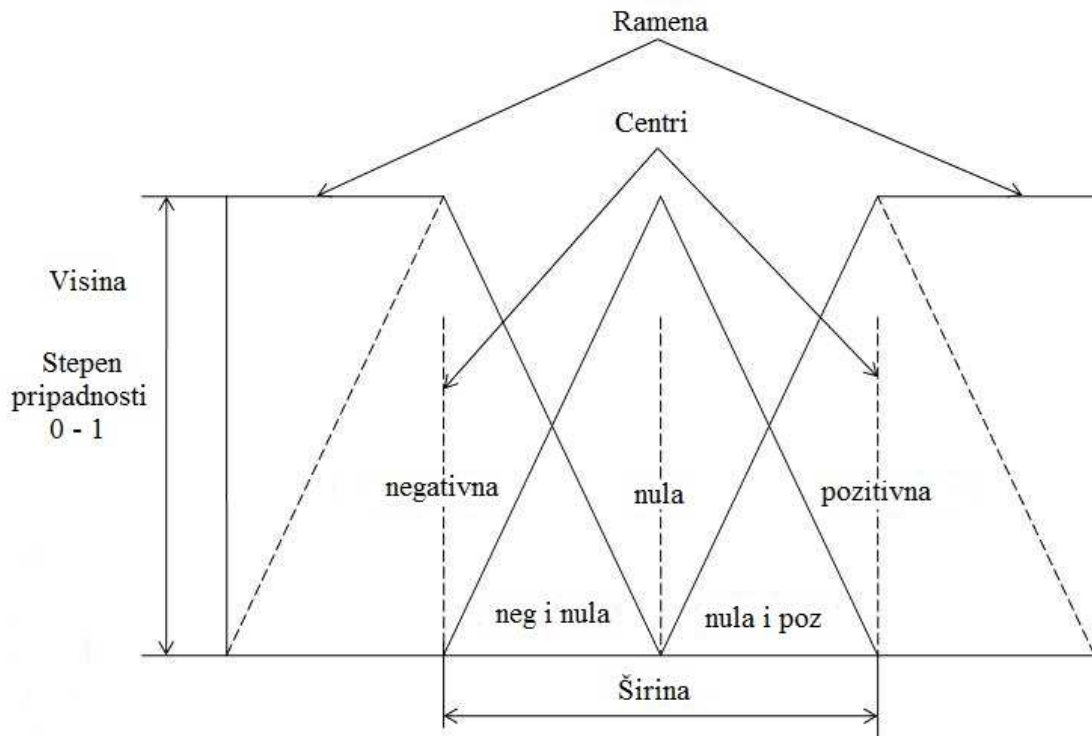
- *Oblik* funkcije je najčešće trouglast, međutim može biti i zvonastog, trapezoidnog oblika ili oblika definisanog eksponencijalnom funkcijom. Moguće su i složenije funkcije, međutim za njihovu primenu potrebna su složenija izračunavanja a samim tim potrebni su i veći računarski kapaciteti za njihovu primenu,
- *Visina* ili veličina se obično normalizuje na vrednost 1,
- *Širina* se izražava veličinom koja opisuje ulazni parametar,
- *Ramena* definiše visinu maksimuma ostalih funkcija i njihova najčešća vrednost je 1,
- *Centar* predstavlja centralnu liniju oblika pojedine funkcije,
- *Tačke Preklapanja* se obično nalaze na oko 50% širine, ali je moguće da bude i na kraćem rastojanju od centra.

Karakteristike funkcije pripadnosti, opisane u prethodnom tekstu, mogu se videti na slici 27.

Bilo bi moguće koristiti i ostale oblike, kao što je to navedeno. Na horizontalnoj osi se nalaze vrednosti izabranog parametra koji može biti greška ili brzina promene greške. Stepenu pripadnosti se određuje se tako što se izabrana vrednost ulaznog parametra projektuje do gornje granice funkcije pripadnosti.

Ponderisane vrednosti greške i brzine promene greške su logički pomnožene, upotrebom logičkog operatora AND. U logičkom proizvodu, kao rezultat se uzima manja od vrednosti koje u proizvodu učestvuju. Kao i u prethodnom razmatranju dobijene su vrednosti čija pripadnost skupu mogućih izlaza nije nula, a to su polja 4, 5, 7 i 8, s tim što se sada računaju njihove pripadnosti skupu izlaznih vrednosti. Pod pretpostavkom da sve mogućnosti imaju

istu stepen pripadnosti i da ona iznosi 0,5 odnosno 50%. Ostale mogućnosti: 1, 2, 3, 6 i 9 se ne uzimaju se u obzir pošto je njihov stepen pripadnosti skupu izlaznih akcija jednak nuli odnosno 0%.



Slika 27. Karakteristike funkcije pripadnosti

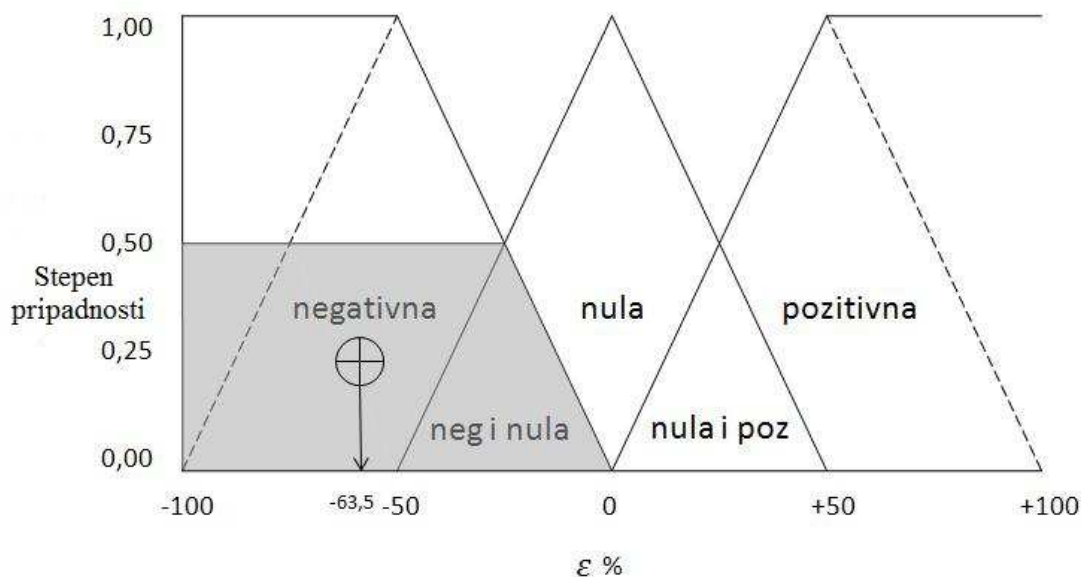
Ponderisane vrednosti greške i brzine promene greške su logički pomnožene, upotrebom logičkog operatora AND. U logičkom proizvodu, kao rezultat se uzima manja od vrednosti koje u proizvodu učestvuju. Kao i u prethodnom razmatranju dobijene su vrednosti čija pripadnost skupu mogućih izlaza nije nula, a to su polja 4, 5, 7 i 8, s tim što se sada računaju njihove pripadnosti skupu izlaznih vrednosti. Pod pretpostavkom da sve mogućnosti imaju istu stepen pripadnosti i da ona iznosi 0,5 odnosno 50%. Ostale mogućnosti: 1, 2, 3, 6 i 9 se ne uzimaju se u obzir pošto je njihov stepen pripadnosti skupu izlaznih akcija jednak nuli odnosno 0%.

Kako bi se odredila odgovarajuća akcija sistema cilju korekcije greške moguće je koristiti nekoliko različitih metoda.

Min-Max metoda

Ispituje stepen pripadnosti svakog pravila skupu izlaznih akcija i bira onu opciju sa najvećom vrednosti. Kao izlaz se uzima horizontalna koordinata fazi težišta ispod posmatrane funkcije. Fazi težište se pronalazi kao težište planarne geometrijske figure oivičene samom funkcijom pripadnosti kao i horizontalnom i vertikalnom osom. Ovaj metod ne uzima u obzir sva pravila

koja se mogu primeniti međutim, njen izlaz je neprekidna funkcija koje je jednostavna za primenu. Na slici 28. se može videti grafička interpretacija pojma fazi težišta. Za postupak izračunavanja vrednosti iz nejasnih - fazi podataka koristi se još i termin „defazifikacija“.



Slika 28. Grafička interpretacija fazi težišta

Max-sa tačkom ili Max-Proizvod metoda

Svodi vrednosti lokalnog maksimuma svake od funkcija pripadnosti na istu vrednost. Uzima horizontalnu koordinatu fazi težišta tako dobijene figure i na osnovu nje određuje izlaznu akciju. Ovaj metod kombinuje uticaj svih aktivnih pravila i kao rezultat daje glatki kontinuirani izlaz.

Metoda uprosečavanja

je funkcionalna ali ne uspeva da poveća ponderisanje za više pravila za pojedinačne izlazne funkcije članice. Na primer ako se u obzir uzmu tri negativna pravila i jedno nulto, ova metoda će kao prosečnu vrednost uzeti 0,5 bez obzira što je negativnih pravila tri puta više.

Root-Sum-Square (RSS) metod

kombinuje efekte svih važećih pravila, svodi funkcije na svoje maksimalne veličine, i izračunava fazi centroid zbirne oblasti. Ovaj metod je mnogo komplikovaniji za izračunavanje od drugih metoda, ali daje najbolje ponderisanje uticaja na sva pravila koja utiču na ishod.

Fazi centroid algoritam

Defazifikacija podataka i dobijanje vrednosti izlaza se postiže kombinovanjem rezultata procesa zaključivanja a zatim izračunavanja fazi centroida te oblasti. Ponderisane prednosti svake izlazne funkcije članice se množe svojim stepenom pripadnosti a zatim se sabiraju. Na kraju se ova oblast deli zbirom ponderisane funkcije članice i taj rezultat se uzima kao numerička izlazna vrednost.

Može se zaključiti da fazi logika pruža potpuno drugačiji, neuobičajen način pristupa problemu uočavanja grešaka i korigovanju istih [33]. Ovaj metod se fokusira na ono što sistem treba da uradi, a ne pokušava da shvati kako to funkcioniše. Na taj način se postiže koncentracija za rešavanje problema umesto pokušaja da se kreira matematički model sistema. Ovo skoro uvek dovodi do bržeg i efikasnijeg rešenja. Fazi tehnologiju nije teško primeniti, a rezultati su najčešće neočekivano dobri.

2.1.4.2 Zaštita privatnosti biometrijskih podataka

Zaštita privatnosti biometrijskih podataka se može sprovesti na više različitih načina. Zaštita privatnost podrazumeva čuvanje reprezentata biometrijske veličine – templejta u obliku koji nije prepoznatljiv neovlašćenim entitetima. Preciznije, potrebno je omogućiti da se biometrijska veličina može transformisati u svoj templejt dok je istovremeno potrebno onemogućiti da se iz templejta može rekonstruisati sama biometrijska veličina. U ovom radu se ima na umu otisak prsta, kao biometrijska veličina koja je predmet proučavanja.

Pandan opisanoj funkcionalnosti su heš funkcije, koje će detaljnije biti opisane u narednom poglavlju. Ukratko, heš funkcija je jednosmerna funkcija koja nema svoju inverznu funkciju. Svako, pa i najmanje odstupanje binarnog niza nad kojim se primanjuje heš funkcije uzrokuje značajnu promenu izračunate vrednosti - heš vrednosti.

Imajući u vidu navedenu činjenicu, kao i činjenicu da se biometrijska veličina reprezentuje uvek na drugačiji način, odnosno sa određenim odstupanjima, više je nego očigledno da bi izračunata heš vrednost značajno varirala ukoliko bi se primenila na reprezentate biometrijskih veličina. Može se izvesti zaključak da su klasične heš funkcije neupotrebljive kada su biometrijske veličine u pitanju.

U literaturi je opisano više metoda zaštite biometrijskih podataka [34], [35]. U narednom delu teksta će biti opisana zaštita biometrijskog podatka primenom sigurnosnog skeča. Radi se o konvertovanju informacije sa šumom u kriptografski ključ koji je upotrebljiv u različitim kriptografskim operacijama. U posmatranom slučaju se obrađuju biometrijske informacije. Imajući u vidu da se generalno radi o informacijama sa šumom, očigledno je da se principi koji će biti opisani mogu primeniti i u mnogim drugim oblastima. Pored navedenog, postiže se kvalitetna zaštita biometrijskih podataka kao i visok nivo stepena prepoznavanja dva otiska prsta ukoliko se koriste ove metode.

Pre daljih razmatranja, potrebno je imati u vidu sledeće dve činjenice:

1. Biometrijska veličina se nikada ne reprodukuje na isti način
2. Prilikom prevođenja reprezentata biometrijske veličine u binarni niz, distribucija

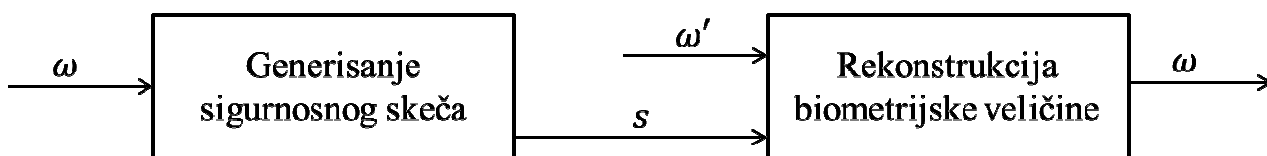
binarnog niza nije ni približna uniformnoj raspodeli

Generalni problem predstavlja konvertovanje ulazne biometrijske, neuniformne veličine sa šumom u ravnomerno distribuirani niz.

Da bi se pristupilo rešavanju opisanih problema uvode se dve veličine [36]:

1. Fazi (*Fuzzy*) ekstraktor koji ima sposobnost da izdvoji uniformni binarni niz R iz reprezentata biometrijske veličine. Može se zaključiti da je fazi ekstraktor otporan na šumove koji se nalaze u osnovnom, biometrijskom podatku. Naročito je bitna činjenica da R ostaje nepromenjeno uprkos različitim očitavanjima biometrijske veličine, sve dok se razlike kreću u određenim granicama. Tačnije, niz R ima približno ravnomernu raspodelu. Na taj način se R može pouzdano reprodukovati i koristiti kao kriptografski ključ u kriptografskim operacijama. Pouzdanim reprodukovanjem niza R smatra se da se reprodukovana verzija niza R kreće u dozvoljenim, predefinisanim granicama, koje su eksponencijalno male.
2. Bezbednosni skeč predstavlja javnu informaciju koja sadrži sve neophodne informacije o ulaznom podatku, biometrijskoj veličini. Ukoliko se prezentuje biometrijska veličina ω , odnosno njen binarni reprezent koji je identičan originalnom nizu ili bar dovoljno blizak ω' , može se rekonstruisati originalna biometrijska veličina ω . Navedni metod se može efikasno koristiti prilikom zaštite identiteta biometrijske veličine kao i za pozdanu identifikaciju osobe.

Šematski prikaz bezbednosnog skeča je dat na slici 29.



Slika 29. Šematski prikaz bezbednosnog skeča

Prednosti prikazanog metoda kreiranja skeča su višestruke. Prilikom prenosa skeča kroz javne komunikacione kanale ne može doći do kompromitovanja same biometrijske veličine. Ukoliko bi napadač i došao do sigurnosnog skeča, ne bi mogao da otkrije izvorni biometrijski podatak. Na taj način je biometrijski podatak siguran a samim tim i identitet posmatrane osobe.

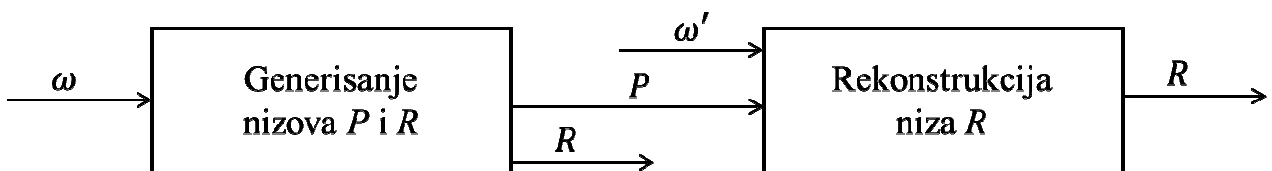
Bezbednosni skeč se može koristiti i za skladištenje u bazama podataka, ili na karticama sa mikrokontrolerom ili na karticama sa laserskom memorijom i čipom. U tom slučaju kompromitovanje baze podataka ili podataka sa kartice ne znači istovremeno i kompromitovanje same biometrijske veličine i identiteta osobe.

Osnovni nedostatak je što se iz bezbednosnog skeča, uz prisustvo originalne biometrijske veličine ipak može rekonstruisati originalna biometrijska veličina što otvara prostor za

moguće napade i ugoržavanje bezbednosti biometrijske veličine kao i identiteta ličnosti.

Da bi se otklonio pomenuti nedostatak, odnosno mogućnost rekonstrukcije izvorne biometrijske veličine, uvodi se fazi ekstraktor koji iz binarnog reprezentata biometrijske veličine izdvaja niz R , sa osobinama koje su opisne. Pored toga, uvodi se niz P koji je identične dužine kao i niz R . Niz R predstavlja tajni podatak koji se dobija iz sigurnosnog skeča ω i generisanog niza P . Niz R se može koristiti kao kriptografski ključ u kriptografskim operacijama i njegova raspodela je uniformna. Uvođenjem ovih veličina otklanja se nedostatak sigurnosnog skeča u smislu mogućnosti rekonstrukcije originalne biometrijske veličine. Suština je u bezbednom prenosu niza R kroz otvorenu mrežu. Niz R se može rekonstruisati jedino ukoliko je prisutna biometrijska veličina ω' koja je jednaka ili dovoljno bliska biometrijskoj veličini ω . I u ovom slučaju se dozvoljavaju određena odstupanja između ω i ω' .

Šematski prikaz opisanog pristupa je prikazan na slici 30.



Slika 30. Šematski prikaz fazi ekstraktora

Imajući u vidu da se niz P formira na osnovu niza R , može se reći da je:

$$P = f(R) \quad (14)$$

Radi pojašnjenja termina uniformnosti podataka, u narednom delu teksta biće dati kratka objašnjenja ovog pojma. U posmatranom slučaju razmatramo binarni niz koji se sastoji od nula i jedinica pa će se objašnjenja na takav niz i odnositi. U slučaju binarnog niza reč je o diskretnoj binarnoj raspodeli.

Niz poseduje uniformnu raspodelu ukoliko svaki od njegovih elemenata ima istu verovatnoću pojavljivanja u nizu. Ukoliko se radi o nizu koji u kome se pojavljuje n različitih elemenata, verovatnoća pojavljivanja svakog od elemenata je:

$$P_n = \frac{1}{n} \quad (15)$$

Pošto se razmatra binarni niz koji se sastoji od nula i jedinica, odnosno dva člana, verovatnoća pojavljivanja svakog od članova je:

$$P_2 = \frac{1}{2} = 0,5 \quad (16)$$

Pojednostavljeno rečeno, binarni niz, sastavljen od nula i jedinica ima uniformnu raspodelu ukoliko sadži jednaki broj nula i jedinica. Imajući u vidu da je ovaj uslov teško ispuniti a nemoguće ukoliko je broj članova neparan, binarni niz se smatrata približno uniformnim ukoliko ima približno jednak broj nula i jedinica.

Bezbednosni skeč se formira procedurom koja se može obeležiti sa SS. Na sličan način se može posmatrati i rekonstrukcija biometrijske veličine iz bezbednosnog skeča. Proceduru koja vrši rekonstrukciju biometrijske veličine iz bezbednosnog skeča možemo označiti sa Rec. Očigledno je da SS i Rec predstavljaju dve procedure koje su međusobno zavisne.

Pre daljeg razmatranja potrebno je pomenuti da se biometrijske veličine posmatraju u različitim metričkim prostorima.

1. *Hamingov metrički prostor - Richard Wesley Hamming (1915. - 1998.)*

Hamingovo rastojanje između dva niza jednake dužine je predstavljeno brojem pozicija na kojima su odgovarajući članovi različiti. Takođe se može reći da se Hamingovo rastojanje meri minimalnim brojem zamena potrebnih da bi se jedan string transformisao u drugi, ili minimalan broj grešaka koje bi trebalo ispraviti koje bi trebalo korigovati da bi se jedan string transformisao u drugi.

2. *Metrika razlike skupova*

Polazi se od pretpostavke da postoje dva skupa čiji se pojedini elementi međusobno razlikuju. Razlika dva skupa se može definisati na sledeći način:

$$\omega \Delta \omega' \stackrel{\text{def}}{=} \{x \in \omega \cup \omega' | x \notin \omega \cap \omega'\} \quad (17)$$

3. *Levenštajnovno rastojanje (Владимир Иосифович Левенштейн, 1935)*

U literaturi se može naći i pod nazivom „edit distance“. Rastojanje između ω i ω' se definiše kao najmanji broj karaktera koje je potrebno ubaciti i izbrisati da bi se ω transformisalo u ω' . Levenštajnovno rastojanje je veoma slično Hamingovom rastojanju.

Razmatrajući biometrijske veličine u različitim metričkim prostorima postiže se nezavisnost rezultata od metričkog prostora u kome se problem rešava.

Bezbednosni skeč se može definisati na sledeći način:

Neka je M metrički prostor u kome se rastojanje računa funkcijom *dis*.

Bezbednosni skeč je par randomizovanih procedura SS i Rec . SS je procedura koja kreira bezbednosni skeč dok je Rec procedura koja iz bezbednosnog skeča vrši rekonstrukciju originalne biometrijske veličine. Dve osnovne osobine sigurnosnog skeča su:

1. Skeč procedura SS , za ulaznu veličinu $\omega \in M$ vraća binarni niz $s \in \{0, 1\}$
2. Procedura Rec za rekonstrukciju ulazne veličine ω , uzima elemente $\omega' \in M$ i binarni niz $s \in \{0, 1\}$. Verodostojnost sigurnosnog skeča je potvrđena ukoliko je $dis(\omega, \omega') \leq t$, i tada je $Rec(\omega', SS(\omega)) = \omega$. Ukoliko je $dis(\omega, \omega') > t$, tada se rezultat funkcije Rec ne može smatrati pouzdanim.

Fazi ekstraktor se može definisati na sledeći način:

Fazi ekstraktor je par randomizovanih procedura. Procedura koja generiše nizove R i P , obeležava se sa Gen , dok se procedura koja generiše niz P označava sa Rep .

1. Procedura Gen , za ulaznu veličinu $\omega \in M$ kao izlaznu veličinu daje niz $R \in \{0, 1\}$ i pomoćni niz $P \in \{0, 1\}$.
2. Procedura Rep kao ulazne veličine uzima elemente $\omega' \in M$ kao i binarni string $P \in \{0, 1\}$. Verodostojnost izlaza fazi ekstraktora je potvrđena ukoliko je $dis(\omega, \omega') \leq t$ i ako su R i P generisani na način $(R, P) \leftarrow Gen(\omega)$ pri čemu je $Rep(\omega', P) = R$. Ukoliko je $dis(\omega, \omega') > t$, tada se rezultat funkcije Rep ne može smatrati pouzdanim.
3. Za bilo koju distribuciju W , metričkog prostora M , niz R ima približno ravnomernu raspodelu kao i posmatrano P ako je $(R, P) \leftarrow Gen(W)$.

Vrste grešaka

U narednom delu teksta biće razmotreni osnovni modeli grešaka. Za rešavanje problema neophodno je da sigurnosni skeč i fazi ekstraktor budu u stanju da obrade svako ω' u okviru dozvoljenog odstupanja t , originalne biometrijske veličine ω bez mogućnosti greške. Ukoliko je dozvoljeno odstupanje t veće od dozvoljenog, mogućnost greške postoji.

Ovakav model je dobar jer se pretpostavlja da prilikom očitavanja biometrijske veličine dolazi do stohastičkih grešaka, odnosno grešaka koje nije moguće predvideti. Ukoliko bi se dopustile neznatne greške stepen korekcije greške bi se znatno povećao. Potrebno je imati u vidu činjenicu da čak ukoliko bi se dozvolilo odstupanje izlaznog rezultata, ukoliko je dozvoljeno odstupanje t iznad određene granice nije moguće izvršiti korekciju greške. Druga,

još bitnija činjenica je ta što i najmanje odstupanje u kriptografskom ključu daje potpuno drugačije rezultate u kriptografskim operacijama.

Zaključak je da sigurnosni skeč i fazi ekstraktor moraju da daju rezultate bez greške ukoliko se koriste za generisanje kriptografskih ključeva. Što se tiče same zaštite biometrijske veličine, kao i procesa prepoznavanja, moguće je dopustiti nešto veći stepen greške.

U narednom delu teksta će biti razmotrena tri modela većeg tolerisanja greške kad se vrši zaštita i prepoznavanje biometrijske veličine. Modeli će biti definisani razmatranjem sigurnosnog skeča. Modeli će biti sagledani korišćenjem Hamingove metrike dok je analogne zaključke moguće izvesti i u drugim metričkim sistemima.

Ono što je zajedničko za sva tri modela greške je zahtevana verovatnoća tačnosti reprodukcije originalnog niza koja mora biti veća od $1 - \alpha$. Kao i za svaku verovatnoću, očigledno je da će biti zadovoljen uslov.

$$1 - \alpha < 1 \quad (18)$$

Slučajne greške

Polazi se od pretpostavke distribucija grešaka u podacima poznata. U Hamingovom metričkom prostoru je najčešća distribucija je binarni simetrični kanal BSCp. To znači da je verovatnoća svakog bita u binarnom nizu da je promenio svoje mesto jednaka p . Odavde očigledno sledi činjenica da je verovatnoća ostanka svakog od bitova na svom mestu $1 - p$.

Da bi se slučajne greške uspešno obrađivale, neophodno je da za svako ulazno ω bude zadovoljen sledeći uslov:

$$Rec(W', SS(\omega)) = \omega \quad (19)$$

Sa verovatnoćom od najmanje $1 - \alpha$.

U posmatranom slučaju se za korekciju greške može koristiti Šenonova teorma za korekciju kanala sa šumom. Korišćenjem pomenute teoreme dobijaju se veoma uske granice greške koje se mogu korigovati. Takođe, treba imati u vidu da je poznata distribucija greške malo verovatna što navedeni model čini nepodesnim za većinu primena u oblasti koju razmatramo. Uprkos toj činjenici, ovaj model daje dobru osnovu za razmatranje rezultata drugih modela.

Greške koje zavise od ulazne veličine

Pretpostavka je da greška zavisi od odstupanja $dis(\omega, \omega')$ koje je ograničeno maksimalnim odstupanjem t . Sledeća pretpostavka je da greške u binarnom nizu zavise od izlazne veličine.

Pri tome mora biti zadovoljen uslov:

$$dis(\omega, \omega') < t \quad (20)$$

Kao i uslov da je:

$$Rec(\omega', SS(\omega)) = \omega \quad (21)$$

Sa verovatnoćom od najmanje $1 - \alpha$.

Ovaj model tretira bilo koji složeni model šuma koji se uoči. Za razliku od unapred pretpostavljene distribucije grešaka, u ovom slučaju veličina greške može proveriti eksperimentalno. Ovaj model, takođe, može obraditi greške koje su reda veličina kao i u prvom prikazanom modelu slučajnih grešaka. To je iz razloga što u prvom modelu možemo obraditi greške samo do granica kodiranja koje su definisane Šenonovom teoremom, ali ne i preko tih granica.

Računski ograničene greške

Ove greške mogu biti značajne i zavise kako od ω tako i od javno dostupnih informacija $SS(\omega)$. Ove greške nastaju prilikom izračunavanja ω' na osnovu ulazne veličine ω . Pretpostavlja se da su uzroci ove greške računarski kapaciteti koji sprovode pomenuto izračunavanje. Nije poznato da li se ove greške mogu korigovati ukoliko se nalaze u granicama kodiranja koje su definisane Šenonovom teoremom kao u prethodna dva navedena modela. U tom slučaju je korisno formirati listu grešaka koje se mogu ispraviti. Ako je broj grešaka manji od definisane granice, njihov broj može biti relativno mali one se mogu korigovati posebnom vrstom liste-decodable kodova. Na ovom mestu neće se ići dalje u razmatranje ovog pojma.

Ukoliko je broj grešaka previše mali ili previše velik za ovaj način dekodiranja, može se pokušati sa dekodiranjem primenom Šenonove teoreme.

Postojeći rezultati na ova tri modela za Hamingovu metriku se mogu primeniti i u drugim metričkim prostorima koristeći izračunavanje koje se generalno može predstaviti sledećim izrazom:

$$SS(\omega, x) = \omega \oplus C(x) \quad (22)$$

Može se konstatovati da će svaki kod za korekciju greške koji ispravlja greške u pomenutim modelima omogućiti kreiranje sigurnosnog skeča odnosno fazi ekstraktora koji vrše korekciju greške u modelu koji se posmatra.

2.2 Elementi kriptografije

Kriptografija je reč koja potiče od grčkih reči κρυπτος, „skrivena“, i γραφειν „pisanje“. Kriptografija je nauka koja se bavi metodima očuvanja tajnosti informacija. Ukoliko se poverljive informacije prenose sa mesta na mesto, one postaju ranjive na prisluškivanje. Ukoliko neovlašćeno lice vrši prisluškivanje, može doći do kompromitovanja osetljivih informacija. Ovakvi problemi se mogu sprečiti šifrovanjem informacija koje ih čini

nedostupnim neovlašćenim licima ili organizacijama.

Porastom broja dostupnih informacija, uvećava se problem njihove zaštite [37]. Šifrovanje i digitalni potpis su kriptografske tehnike koje se koriste da bi se implementirali bezbednosni servisi. U narednim poglavljima će neki od osnovnih pojmova biti detaljnije razmotreni.

Osnovni element koji se koristi naziva se šifarski sistem ili algoritam šifrovanja. Svaki šifarski sistem obuhvata par transformacija podataka, koje se nazivaju šifrovanje i dešifrovanje. Šifrovanje je procedura koja transformiše originalnu informaciju (otvoreni tekst) u šifrovane podatke (šifrat). Obrnut proces, dešifrovanje, rekonstruiše otvoreni tekst na osnovu šifrata. Prilikom šifrovanja, pored otvorenog teksta, koristi se jedna nezavisna vrednost koja se naziva kriptografskim ključem za šifrovanje. Analogno, transformacija za dešifrovanje koristi kriptografski ključ za dešifrovanje. Broj simbola koji predstavljaju ključ odnosno, dužina ključa, zavisi od šifarskog sistema i predstavlja jedan od parametara sigurnosti tog sistema. Kriptoanaliza je nauka koja se bavi razbijanjem šifri, odnosno otkrivanjem sadržaja otvorenog teksta na osnovu šifrata, pri čemu kriptografski ključ nije poznat. Šire posmatrano, kriptoanaliza obuhvata i proučavanje slabosti kriptografskih elemenata. Različite tehnike kriptoanalize nazivaju se napadima.

2.2.1 Proces razmene šifrovanih podataka

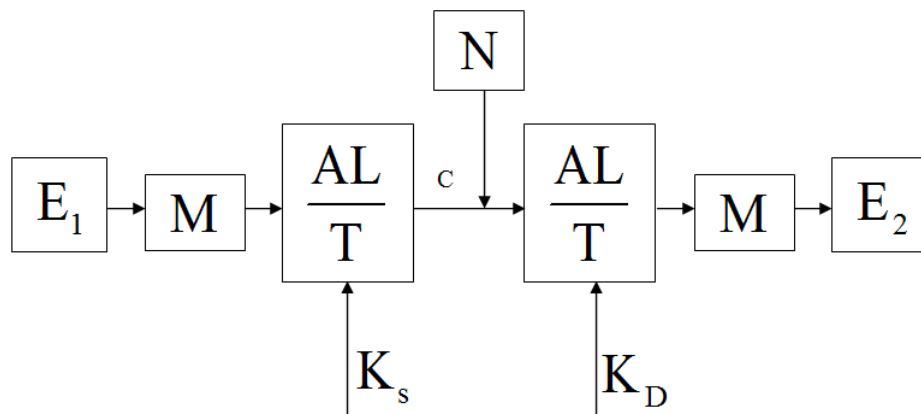
Osnovno pitanje koje se postavlja je kako uspostaviti sigurnu komunikaciju preko nesigurnog komunikacijskog kanala? Metode za rešavanje ovog problema proučava naučna disciplina koja se zove kriptografija [38]. Osnovni zadatak kriptografije je omogućavanje komunikacije između dva entiteta koje ćemo nazvati pošiljalac i primalac. Komunikacija mora da se odvija na takav način da treći entitet, koji nema prava pristupa, ne može razumeti tekst poruke koja se razmenjuje između pošiljaoca i primaoca. Treći entitet možemo nazvati napadačem.

Poruku koju pošiljalac šalje primaocu zovemo otvoreni tekst. Pošiljalac transformiše otvoreni tekst na određeni način korišćenjem kriptografskog ključa. Taj postupak se naziva šifrovanjem, dok se dobijeni rezultat naziva šifratom. Pošiljalac zatim šalje šifrat primaocu koristeći neki komunikacioni kanal.

Napadač koji prisluškuje komunikacioni kanal može doći do šifrata. Međutim, pošto mu nije poznat kriptografski ključ, napadač ne može doći do otvorenog teksta.

Za razliku od napadača, primaocu je poznat kriptografski ključ kojim je šifrirana poruka, pa može da dešifruje šifrat i tako dobije otvoreni tekst.

Opisani proces je u osnovnim crtama predstavljen na slici 31.



Slika 31. Šematski prikaz procesa razmene šifrovanih podataka

Pri čemu je:

- E_1 - Pošaljalac poruke
- E_2 - Primalac poruke
- M - Poruka, Message
- AL - Algoritam
- T - Transformacija
- K_s - Ključ za šifrovanje
- K_D - Ključ za dešifrovanje
- C - Šifrat, Cifer Message
- N - Neprijatelj koji presreće poruku

2.2.2 Elementi matematičkog aparata koji se koristi u kriptografiji

Algebarska struktura

Pre daljih razmatranja biće izloženi elementi algebarske struktura koji se koriste u kriptografskim operacijama.

Algebra, od arapskog al-gabr, se može prevesti kao uklapanje delova, ili uspostavljanje ravnoteže. Geneza naziva vodi do knjige arapskog matematičara Al Horezmija, persijskog matematičara, astronoma i geografa (*Abu Abdulah Muhamed bin Musa el Horezmi, oko 800. - oko 850.*).

Postupak se prvobitno odnosio na uređivanje leve i desne strane jednačina ali je kasnije, razvojem matematike, znatno proširen.

Izvorno se pod algebrom podrazumevala teorija algebarskih jednačina, u kojima su pomoću algebarskih računskih operacija: sabiranje, oduzimanje, množenje, deljenje i druge, povezane

poznate i nepoznate veličine. Danas ovaj pojam označava i opštiju teoriju matematičkih struktura, u kojoj se posebno istražuju strukturne jednakosti između različitih oblasti kao što su teorija brojeva, geometrija ili algebra u tradicionalnom smislu.

Važne algebarske strukture su, recimo, grupe, prsteni, tela i asocijacije. Sve tvorevine koje pokazuju neku određenu strukturu nazivaju se modelima ove strukture [39].

Bulova algebra je deo matematičke logike - algebarska struktura čiju osnovu čine operacije: I (konjunkcija), ILI (disjunkcija) i NE (negacija) kao i skup teorijskih operacija kao što su unija, presek i komplement. Za razliku od elementarne algebre, gde promenljive za vrednosti imaju brojeve, u Bulovoj algebri vrednosti promenljivih mogu biti samo tačno ili netačno (istina ili laž), što se obično označava sa 1 i 0, gde 1 predstavlja tačno a 0 netačno.

Bulova algebra je dobila naziv po tvorcu, Džordžu Bulu, engleskom matematičaru (*George Boole, 1815.- 1864.*).

Bulova algebra je, osim kao deo apstraktne algebre, izuzetno uticajna kao matematički temelj računarskih nauka. Takođe se koristi u teoriji skupova i statistici. Algebra se može podeliti na: linearnu algebru, apstraktnu algebru, univerzalnu algebru, elementarnu algebru.

U univerzalnoj algebri, grani čiste matematike, algebarska struktura se sastoji od jednog ili više skupova zatvorenih u odnosu na jednu ili više operacija, koji zadovoljavaju određene aksiome. Apstraktna algebra se prvenstveno bavi proučavanjem algebarskih struktura i njihovih svojstava.

Apstraktno, algebarska struktura je kolekcija svih mogućih modela datog skupa aksioma. Tačnije, algebarska struktura je bilo koji pojedinačan model nekog skupa aksioma.

Ovu definiciju algebarske strukture ne treba uzeti restriktivno. Sve što zadovoljava aksiome koji definišu neku algebarsku strukturu je instanca te strukture, čak iako ta instanca ima i proizvoljno mnogo drugih aksioma.

Grupoid

je najjednostavniji primer algebarske strukture.

Neka je G neprazan skup i $*$ binarna operacija definisana na tom skupu, grupoid je uređen par $(G, *)$ za koji važi aksiom zatvorenosti tj, za svaka dva elmenta a i b skupa G važi da je $a * b$ element skupa G .

Za grupoid $(G, *)$ kažemo da je komutativan ako i samo ako je operacija $*$ komutativna.

Algebarske grupe su grupe za koje postoji neutralni element: $a * 1 = a$, kao i inverzni element, $a * b = 1$, $a = b^{-1}$.

Abelova grupa

Ako za grupu važi zakoni komutacije $a * b = b * a$, tada se grupa naziva Abelovom grupom.

Drugim rečima, redosled kojim se binarna operacija sprovodi nije bitan. Ovakve grupe je generalno lakše razumeti, iako su beskonačne Abelove grupe i dalje tema istraživanja. Grupe koje nisu komutativne se nazivaju ne-Abelove (ili nekomutativne).

Abelove grupe su dobile ime po norveškom matematičaru Nilsu Abelu (Niels Abel, 1802. – 1829.).

Algoritam

U matematici i informatici, algoritam predstavlja proceduru za proračune u kojoj je opisan svaki korak kao i redosled njihovog izvršavanja. Algoritmi se koriste za izračunavanje, obradu podataka i automatski rezonovanje.

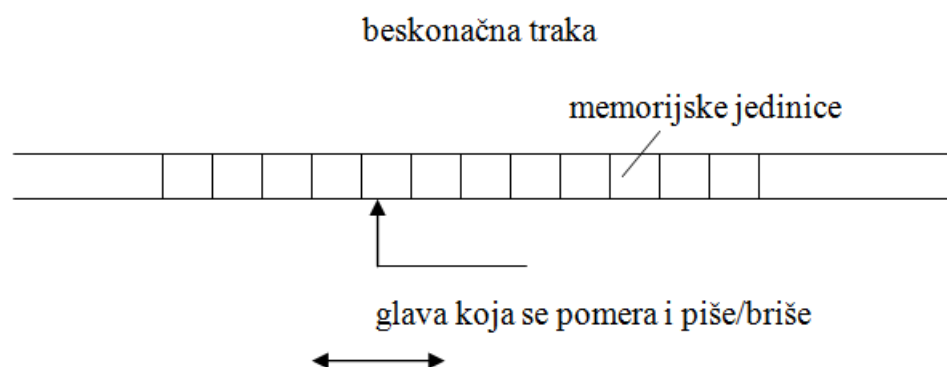
Algoritam je efikasan metod izražen kao konačan spisak dobro definisanih uputstava za izračunavanje funkcije. Polazi se od početnog stanja početnih ulaznih parametara, koji mogu biti i prazan skup. Nakon toga sledi konačan broj precizno definisanih instrukcija koje opisuju proračun koji je predmet rada algoritma. Na kraju se dobija završno stanje koje predstavlja izlazni rezultat. Prelaz iz jednog stanja u sledeće nije obavezno determinističko.

Problem koji pokušavamo da rešimo može biti rešiv ili nerešiv.

Tjuringova mašina

Tjuringova mašina (*Alan Mathison Turing, 1912. - 1954.*) je zamišljeni uređaj koji manipuliše simbolima na traci prema tabeli pravila. Uprkos svojoj jednostavnosti, Tjuringova mašina može da se prilagodi i simulira logiku bilo kog algoritma. Posebno je korisna u objašnjavanju funkcija procesora unutar računara.

Ukoliko se problem može rešiti na Tjuringovoj mašini sledi tvrdnja da je problem rešiv i obrnuto, ukoliko se problem ne može rešiti na Tjuringovoj mašini sledi tvrdnja da problem nije rešiv.



Slika 32. Tjuringova mašina

Tjuringova mašina nije namenjen kao praktičan uređaj koji se može koristiti u računarskoj tehnologije, već kao hipotetički uređaj koji predstavlja sve računarske mašine. Svrha Tjuringove mašine je da pomogne u shvatanju granica mehaničkog računanja.

Neke od osobina Tjuringove mašine su:

- Neograničen kapacitet memorije dobijen u obliku beskonačne trake,
- Traka je podeljena na kvadrate, pri čemu se u svaki od kvadrata može upisati simbol,
- U svakom trenutku postoji samo jedan simbol u mašini, i taj simbol se naziva skenirani simbol,
- Mašina može izmeniti skeniranu simbol pri čemu je ponašanje ponašanje mašine delimično definisano skeniranim simbolom. Pri tome treba imati na umu činjenicu da simboli koji se nalaze u drugim kvadratima na traci, u tom trenutku, ne utiču na ponašanje Tjuringove mašine,
- Traka može da se pomera napred i nazad kroz mašinu, što je jedan od osnovnih operacija mašine. Pri čemu glava može da menja, upisuje ili briše simbole koji se nalaze na traci.

Čerčova hipoteza

Naziva se još i Čerč-Tjuringova hipoteza (*Alonzo Churc, 1903. - 1995.*).

Osnova tvrdnja hipoteze je da su izračunljive samo one funkcije za koje postoji program za Tjuringovu mašinu.

Sledeće pitanje je koji je algoritam najbrži?

Ovde je potrebno napomenuti da u vreme naglog rasta procesorske i memorijske snage računara, mnogi zanemaruju ovu osobinu algoritma. Ipak ukoliko se radi o kompleksnim algoritmima, brzina izračunavanja algoritma nije zanemariva.

Za algoritam A koji rešava problem P , može se izneti tvrdnja:

$$P: A(P) \in M \quad (23)$$

Pri čemu je M klasa izračunljivosti kojoj pripada funkcija koja rešava problem:

$$f(N) \in M \quad (24)$$

gde je N veličina problema.

Naredna tvrdnja Čerčove hipoteze je da je nemoguće napraviti funkciju koja rešava sve ostale funkcije.

Teorija brojeva

Skup celih brojeva Z se može predstaviti na sledeći način:

$$Z = \{\dots - 2, -1, 0, 1, 2 \dots\} \quad (25)$$

Tako da je:

$$a = q * b + r \mid 0 \leq r \leq b \quad (26)$$

Ili ako se iskaže rečima svaka dva cela broja se mogu podeliti sa ostatkom. Izuzetak je deljenje nulom koje nije definisano u skupu celih brojeva.

Modularna aritmetika

Kongruencija ili kongruencija po modulu je matematička operacija koja za rezultat daje ostatak pri deljenju dva prirodna broja. Može se predstaviti na sledeći način:

$$a \equiv b \pmod{n} \quad (27)$$

gde je a ostatak, b je deljenik a n je delilac ili moduo.

Definicija:

$$a, b \in Z \mid a = {}_nb \quad (28)$$

Čita se a je kongruentno b po modulu n ili mod(n)

Osobine kongruentnosti:

1) Refleksivnost

$$a = {}_na \quad (29)$$

Odnosno svaki svaki broj deli nulu.

2) Komutativnost

$$a = {}_nb \Rightarrow b = {}_na \quad (30)$$

3) Tranzitivnost

$$a = {}_nb \wedge b = {}_nc \Rightarrow a = {}_nc \quad (31)$$

Ukoliko je c moduo, svaki prirodan broj pripada nekoj od klasa. Analogno važi i za ceo broj, samo sa znakom minus.

Konačan skup celih brojeva sveden po modulu je:

$$Z_n = \{0, 1, 2, \dots, (n-1)\} \quad (32)$$

Nad ovim skupom se definišu sabiranje i množenje:

$$a, b \in Z_n \quad (32)$$

$$a + {}_nb = c \quad (33)$$

$$c \in Z_n \mid c = {}_n(a + b) \quad (34)$$

Analogno važi i za množenje:

$$a * {}_n b = c \quad (35)$$

$$c \in Z_n \wedge c = {}_n(a * b) \quad (36)$$

Svaki element skupa Z ima svoj inverzni element, sa kojim kad se pomnoži, kao rezultat se dobija broj 1:

$$a = {}_n b \wedge c = {}_n d \quad (37)$$

$$a + c = {}_n(b + d) \wedge a * c = {}_n(b * d) \quad (38)$$

$$a + {}_n b = c \wedge c = {}_n(b + d) \quad (39)$$

Ukoliko su brojevi veliki:

$$a_1 = {}_n a \quad (40)$$

$$b_1 = {}_n b \quad (41)$$

$$a_1 + {}_n b_1 = {}_n(a + b) \quad (42)$$

Postavlja se pitanje kako se može pronaći broj iz Z za veliki broj koji mu je kongruentan?

$$T: a, b \in Z \wedge a = {}_n b \Leftrightarrow r_a + r_b \quad (43)$$

$$a = q_a * n + r_a \quad (44)$$

$$b = q_b * n + r_b \quad (45)$$

Pri čemu je:

$$a \leq r_a, r_b < n \quad (46)$$

Odakle sledi da je:

$$n|(a - b) \quad (47)$$

Odnosno:

$$(a - b) = k * n \quad (48)$$

$$(a - b) = n * (q_a - q_b) + (r_a - r_b) \Rightarrow r_a = r_b \quad (49)$$

Pošto su i jedan i drugi sabirak deljivi sa n .

Kod RSA algoritma, o kome će kasnije biti više reči, koristi se jednačina :

$$a * x = {}_n b \quad (50)$$

$$a * x = {}_n 1 \quad (51)$$

Traženje najvećeg zajedničkog delioca

NZD, u oznaci $\{a, b\} = (a, b)$

$d = (a, b) :$

- 1) $d|a, d|b$
- 2) d je najveći takav broj

Kada je:

$$d = 1 = (a, b) \quad (51)$$

Tada su a i b uzajamno prosti brojevi, ili u oznaci: $a \perp b$

Pojam uzajamno prostih brojeva će biti iskorišćen prilikom razmatranja RSA algoritma.

Skup zajedničkih delilaca je ograničen odozgo, postavlja se pitanje kako pronaći najveći?

Euklidov algoritam

$(a, b) = d$ i pretpostavimo da je $a > b$.

Koraci algoritma su sledeći:

- 1) $a = r_{-1}, b = r_0, k = 0$
- 2) $r_{k-1} = q_k * r_k + r_{k+1}$
- 3) Ako je $r_{k+1} \neq 0$ tada je $k = k + 1$ i vraćamo se na korak 2)
- 4)

$$r_k = (a, b) \quad (52)$$

Mogu se dobiti brojevi x_0, y_0 takvi da je:

$$a * x_0 + b * y_0 = d \quad (53)$$

Dalje razmatranje ove pretpostavke dovodi do proširenog Euklidovog algoritma.

Prošireni Euklidov algoritam

Kada jednačina:

$$a * x = {}_n1 \quad (54)$$

ima rešenje u skupu Z_n :

- 1) $(a, n) = 1$, uzajamno prosti,

Tada jednačina ima tačno jedno rešenje po modulu n .

$$a * x_0 + n * y_0 = 1 \quad (55)$$

$$a * x_0 = (1 - n * y_0) = {}_n 1 \quad (56)$$

Ako ovaj uslov nije ispunjen jednačina nema rešenja.

- 2) $(a, n) \neq 1$

$$a * x_0 - 1 = k * n \quad (57)$$

$$a * x_0 - k * n = 1 \quad (58)$$

$$d|(a * x_0 - k * n) = 1 \quad (59)$$

$$d|1 \Rightarrow d = 1 \quad (60)$$

Što je suprotno početnoj pretpostavci da je $d \neq 1$.

Za opštu jednačinu:

$a * x = {}_n b$ u skupu Z_n , ako $d(a, n)$

- 1) $d|b$, jednačina ima d rešenja
- 2) d ne deli b , jednačina nema rešenja

Ojlerova funkcija

$\Phi \rightarrow \varphi(n)$

$\varphi(n)$ je skup prirodnih brojeva manjih od n koji su uzajamno prosti sa n

Ako su $p_1, p_2, \dots, p_k$ prosti brojevi a $\alpha_1, \alpha_2, \dots, \alpha_k$ odgovarajući eksponenti, tada je:

$$n = p_1^{\alpha_1} * p_2^{\alpha_2} * \dots * p_k^{\alpha_k} \quad (61)$$

$$\varphi(n) = \left(1 - \frac{1}{p_1}\right) * \left(1 - \frac{1}{p_2}\right) * \dots * \left(1 - \frac{1}{p_k}\right) \quad (62)$$

Ako je n prost broj:

$$\varphi(p) = p * \left(1 - \frac{1}{p}\right) = (p - 1) \quad (63)$$

$$n = p * q \quad (64)$$

$$\varphi(p * q) = (p - 1) * (q - 1) \quad (65)$$

Mala Fermaova teorema

Ako su a i n uzajamno prosti brojevi:

$$T: (a, n) = 1 \quad (66)$$

$$a^{\varphi(n)} = {}_n1 \quad (67)$$

Matematička verovatnoća i statistika

Verovatnoća je odnos broja povoljnih i broja mogućih ishoda:

$$P = \frac{m}{n} \quad (68)$$

Osobine verovatnoće su:

1)

$$0 \leq P \leq 1 \quad (69)$$

2) Ako su A i B neki događaji i ako je $A \cap B = \emptyset$, tada je:

$$P(A \cup B) = P(A) + P(B) \quad (70)$$

3) Ako su $A_1, A_2, \dots, A_n$ disjunktni događaji, tada je skup događaja $\Omega = A_1 \cup A_2 \cup \dots \cup A_n$ sa osobinom da je:

$$P(\Omega) = P(A_1) + P(A_2) + \dots + P(A_n) = 1 \quad (71)$$

Ako događaji utiču jedan na drugi tada su zavisni, a ako ne utiču jedan na drugi tada su nezavisni.

Verovatnoća događaja A pod uslovom da se desio događaj B je :

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (72)$$

ili:

$$P(A \cap B) = P(A|B) * P(B) \quad (73)$$

Diskretni događaji su događaji koji imaju konačno mnogo ishoda.

Ako su $a_1, a_2, a_3, \dots, a_k$ događaji a $P_1, P_2, P_3, \dots, P_k$ odgovarajuće verovatnoće navedenih događaja, tada je raspodela verovatnoća za događaj A:

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \\ P_1 & P_2 & P_3 \end{pmatrix} \quad (74)$$

Tada je moguća primena funkcije $y = f(A)$.

Entropija

Entropija predstavlja naše očekivanje da se neki događaj desi.

Entropija je mera za srednju količinu bita koja je neophodna da bi se kodirali elementi jedne grupe :

$$X = \begin{pmatrix} a_1 & a_2 & a_3 \\ P_1 & P_2 & P_3 \end{pmatrix} \quad (75)$$

$$H(X) = \sum_{i=1}^k p_k * \log_2 p_k \quad (76)$$

Pri čemu je:

$$0 \leq H(X) \leq \log_2 k \quad (77)$$

Sistem je najmanje određen kada su sve verovatnoće jednake.

$$P_k = -k * \frac{1}{k} * \log_2 \frac{1}{k} = -(\log_2 1 - \log_2 k) = \log_2 k \quad (78)$$

Sistem je najviše određen ako su sve verovatnoće nula a samo jedna je 1.

Združena entropija

Entropija za dvodimenzionu promenljivu je:

$$H(X, Y) = \sum_{x,y} P(X = x; Y = y) * \log(X = x; Y = y) \quad (79)$$

$$H(X, Y) \leq H(X) + H(Y) \quad (80)$$

$$H(X|Y = y) = \sum_x P(X = x; Y = y) * \log(X = x; Y = y) \quad (81)$$

$$H(X|Y) = \sum_y^x P(X = x; Y = y) * \log(X = x; Y = y) \quad (82)$$

Osobine uslovne entropije

$$H(X|Y) \geq 0 \quad (83)$$

$$H(X|Y) = 0 \quad (84)$$

$$H(X, Y) = H(X) + H(X|Y) = H(Y) + H(Y|X) \quad (85)$$

$$H(X|Y) \leq H(X) \quad (86)$$

2.2.3 Simetrični šifarski sistemi

Kod simetričnih šifarskih sistema isti ključ se koristi za šifrovanje i dešifrovanje. Postoje raznovrsni i mnogobrojni algoritmi koji koriste simetrični šifarski sistem. Postignuta je visoka bezbednost razvijenih algoritama.

Brzina izračunavanja šifrata kod simetričnih šifarskih sistema je relativno velika.

Osnovni problem kod simetričnih šifarskih sistema je razmena kriptografskih ključeva. Ukoliko postoji više učesnika u razmeni šifrovanih podataka, razmena ključeva postaje isuviše komplikovna, pogotovo ukoliko svaki par učesnika koji razmenjuju podatke koristi različiti kriptografski ključ.

Ukoliko napadač uspe da presretne šifrat i ukoliko mu je poznat kriptografski ključ, nema nikakvih smetnji da rekonstruiše otvoreni tekst poruke.

Simetrični šifarski sistemi se dele na :

a) Sekvencijalne

elementi za šifrovanje se proizvode uzastopno, imaju veličinu jedne jedinice, veličina bloka = 1,

b) Blokove

veličina bloka > 1.

DES algoritam

Kao primer blok šifarskog sistema, u narednom delu teksta, razmotrićemo DES (*Data Encryption Standard*) algoritam je najčešće korišćen algoritam za kriptovanje. Dugo godina, DES je među ljudima bio sinonim za sigurno šifriranje. DES je jedan od prvih algoritama za šifrovanje koji je odobrila NSA (*National Security Agency*) još 1977. godine. DES je brzo prihvaćen u nedigitalnim medijima, kao u javnim telefonskim linijama. Ipak, treba imati na umu da je prošlo dosta vremena od objavljivanja i verifikacije algoritma. Prema mnogim tvrdnjama DES algoritam je razbijen još 1988. godine kada je obelodanjeno da je za potpuno razbijanje DES ključa potrebno 56 sati. DES ključ ukupno sadrži 64 bita, od čega 56 bitova čine ključ, dok su preostalih osam kontrolni bitovi. Uređaj specijalno konstruisan za razbijanje algoritma je nazvan *Deep Crack*, i sadržao je 27 ploča sa po 64 mikroprocesora. Bio je sposoban da testira 90 milijardi ključeva u sekundi. Očigledno je da je razbijanje DES kriptografskog ključa danas znatno efikasnije.

Šifrovanje DES algoritmom se odvija u 16 faza. Od ulaznog ključa K, generiše se šesnaest 48-bitnih podključeva, svaki za po jedan ciklus.

Ulazni tekst se podeli na blokove od 64 bita koji se šifruju. Toliko iznosi i dužina DES ključa. Zatim se svaki blok podeli na dva dela od po 32 bita levi L_0 i desni R_0 kao početne vrednost i leve i desne strane. Ulazni kriptografski ključ sadrži 64 bita od čega je 8 parity bitova. Sam ključ je dug 56 bita.

Svaki ciklus šifrovanja je ekvivalentan. U svakom ciklusu se uzima 32 bitni ulazni niz L_{i-1} i R_{i-1} koji se dobija kao rezultat iz prethodnog ciklusa izračunavanja.

Izračunavanje leve i desne polovine niza za svaki ciklus se vrši na sledeći način:

$$L_i = R_{i-1} \quad (87)$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \quad (88)$$

DES algoritam može prikazati na sledeći način:

Izračunavanje ključeva

Ulazni podatak: 64-bitni ključ $K = k_1, k_2, \dots, k_{64}$. U niz je uključeno i 8 parity bitova.

Izlazni podatak: šesnaest 48-bitnih ključeva $K_i, 1 \leq i \leq 16$.

- 1) Definisanje $v_i, 1 \leq i \leq 16$ na sledeći način:

$$v_i = 1 \text{ za } i \in \{1, 2, 9, 16\} . \text{ Za sve ostale slučajeve } v_i = 2$$

- 2) $T \leftarrow PC1(K)$ predstavlja T kao 28-bitne polovine (C_0, D_0)

- 3) Za $1 \leq i \leq 16$, izračunava se K_i na sledeći način:

$$C_i \leftarrow (C_{i-1} \leftarrow v_i), D_i \leftarrow (D_{i-1} \leftarrow v_i), K_i \leftarrow PC2(C_i, D_i)$$

DES šifrovanje

Ulazni podatak: Otvoreni tekst $m_1, m_2, \dots, m_{64}$; 64-bitni ključ $K = k_1, k_2, \dots, k_{64}$; sa uključenim parity bitovima.

Izlazni podatak: 64-bitni blok šifrata $C = c_1, c_2, \dots, c_{64}$.

- 1) Izračunavanje ključeve za svaki ciklus šifrovanja, kako je opisano u prethodnom delu teksta.
- 2) $(L_0, R_0) \leftarrow IP(m_1, m_2, \dots, m_{64})$, formiranje permutacija ulaznog niza.
- 3) 16 ciklusa izračunavanja za $1 \leq i \leq 16$. Izračunavanje L_i i R_i prema obrascima 87 i 88. Dalje se vrše izračunavanja na sledeći način:

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i)) \quad (89)$$

i to:

- a) Proširivanje $R_{i-1} = r_1 r_2 \dots r_{32}$ sa 32 bita na 48 bita koristeći E – definisana tabela sa permutacijama i to: $T \leftarrow E(R_{i-1})$.

- b) $T' \leftarrow T \oplus K_i$. Pri čemu T' predstavlja osam 6-bitnih nizova:
 $(B_1, B_2, \dots, B_8) = T'$
- c) $T'' \leftarrow (S_1(B_1), S_2(B_2), \dots, S_8(B_8))$ – definisana tabela
- d) $T''' \leftarrow P(T'')$
- 4) $b_1, b_2, \dots, b_{64} \leftarrow (R_{16}, L_{16})$, razmena konačnih blokova
- 5) $C \leftarrow IP^{-1}(b_1 b_2 \dots b_{64})$

DES algoritam je, nakon inicijalne verzije, poslužio za razvoj sličnih algoritama kao što su:

- 2-DES
- 3-DES
- AES - Advanced Encryption Standard
- Blowfish
- ...

Svi navedeni algoritmi funkcionišu na sličan način kao i osnovni DES algoritam. Više detalja o svakom od ovih algoritma se može pronaći u literaturi koja je široko rasprostranjena [40], [41].

Razmena ključa

Problem razmene kriptografskog ključa za simetrične šifarske sisteme se može rešiti upotrebom Difi-Helmanovog algoritma (*Bailey Whitfield Diffie, 1944; Martin Edward Hellman, 1945*).

Definicija 1:

Element g nazivamo generatorom grupe G ako svaki element grupe G može biti izražen kao proizvod konačno mnogo stepena g .

Definicija 2:

Ako je $p \geq 1$, gde je p ceo broj, onda brojevi uzajamno prosti sa p , po modulu p , formiraju grupu. Ili u oznaci $(\mathbb{Z}/p\mathbb{Z})^x$ ili $\mathbb{Z}_p^*$.

Definicija 3:

Broj invertovanih elemenata po modulu n se označava sa $\phi(n)$, gde je $\phi(n)$ Ojlerova funkcija.

Definicija 4:

Pretpostavimo da je g prost koren po modu p . Ako je $g^x \equiv y(mod p)$ ta da je diskretni logaritam ili indeks, za osnovu g :

$$\text{ind}_g(y) = x \pmod{\phi(p)} \quad (90)$$

Difi-Helmanov algoritam

U generisanju kriptografskog ključa učestvuju dva entiteta A i B. A i B dogovaraju javne brojeve g i p , sa svojstvima kako je to opisano u prethodno navedenim definicijama [42].

- 1) A bira slučajan broj a i izračunava: $u \equiv g^a \pmod{p}$ i dobijeni rezultat šalje B
- 2) B bira slučajan broj b i izračunava: $v \equiv g^b \pmod{p}$ i dobijeni rezultat šalje A
- 3) B izračunava ključ: $k \equiv u^b \equiv (g^a)^b \pmod{p}$
- 4) A izračunava ključ: $k \equiv v^a \equiv (g^b)^a \pmod{p}$

Očigledno je da sada oba entiteta koja učestvuju u razmeni, A i B, imaju isti ključ k koji je poznat samo njima.

2.2.4 Kriptografija sa javnim ključevima

Migracija računarskih okruženja ka distribuiranim, mrežnim rešenjima radikalno je promenila način na koji kompanije realizuju poslovanje. Kriptografski mehanizmi predstavljaju najrasprostranjeniji način za ostvarivanje bezbednosti u otvorenim sistemima i mrežama. U novije vreme je definisan širok dijapazon novih sistema za zaštitu Internet sistema, i u većini se koriste kriptografski mehanizmi zaštite. Ove tehnologije se primenjuju za različite sisteme elektronske razmene kao što su zaštićeno slanje poruka, zaštićeni e-mail servis, sistemi elektronskog plaćanja, zaštita softverskih aplikacija i mrežnih komunikacija.

Infrastruktura sistema sa javnim ključevima (*PKI - Public Key Infrastructure*) omogućava okruženje za pouzdanu primenu elektronskog poslovanja i najčešće se bazira na kombinovanoj primeni asimetričnih i simetričnih šifarskih sistema [43].

PKI infrastruktura se sastoji od više komponenata, aplikacija, i dokumenata koji definišu način realizacije četiri osnovne kriptografske funkcije u elektronskom poslovanju:

- Zaštita tajnosti – realizuje se simetričnim kriptografskim sistemima,
- Autentičnost – asimetrični šifarski sistemi,
- Integritet - asimetrični šifarski sistemi,
- Neporecivost transakcija - asimetrični šifarski sistemi,

Dok se funkcije zaštite tajnosti i integriteta mogu realizovati i primenom tradicionalnih simetričnih tehnika, funkcije autentičnosti i neporecivosti transakcija zahtevaju primenu sofisticiranijih sistema - PKI sistema. Najbolje karakteristike pokazuju sistemi u kojima su realizovane sve pomenute četiri funkcije.

U cilju eventualnih zahteva za interoperabilnošću, PKI sistem mora zadovoljavati osobinu da

se bazira na otvorenim standardima, od kojih je najvažniji X.509v3 standard za format digitalnog sertifikata.

Asimetrični šifarski sistemi, za razliku od simetričnih šifarskih sistema, koriste različite ključeva za šifrovanje i dešifrovanje šifrata. Generiše se par ključeva, tako da se jednim ključem može šifrovati otvoreni tekst a drugim dešifrovati i obrnuto. Obično se jedan od ključeva naziva tajnim ključem i poznat je samo vlasniku dok se drugi ključ naziva javnim ključem i poznat je svim entitetima sa kojima vlasnik tajnog ključa razmenjuje šifrovane poruke. Na ovaj način se rešava osnovni problem prilikom korišćenja simetričnih šifarskih sistema, a to je problem upravljanja velikim brojem kriptografskih ključeva.

RSA algoritam

RSA algoritam je standardni asimetrični algoritam i dobio je ime po svojim tvorcima: *Ronald Linn Rivest, 1947; Adi Shamir, 1952; Leonard Max Adleman, 1945*. RSA algoritam je namenjen prvenstveno šifrovanju podataka ali se koristi i u sistemima elektronskog potpisivanja. RSA danas predstavlja industrijski standard u oblasti asimetrične kriptografije i zaštiti podataka, tako da je široko primenjen u mnogim sigurnosnim protokolima i sistemima elektronskog poslovanja.

Definisanje RSA algoritma:

- 1) Generišu se dva uzajamno prosta broja p i q , pri čemu je $p \neq q$
- 2) Izračunava se proizvod $n = p * q$
- 3) Izračunava se Ojlerova funkcija $\phi(n) = (p - 1)(q - 1)$
- 4) Izabere se celobrojna vrednost e pri čemu je $1 < e < \phi(n)$
- 5) Izračunava se d i to na sledeći način:

$$d * e = 1(\text{mod}(\phi(n))) \quad (91)$$

- 6) Javni ključ je par (n, e) dok je privatni ključ par (n, d) .

Javni ključ se javno i objavljuje. Bilo ko ko zna javni ključ može šifrovati otvoreni tekst pomenutim javnim ključem dok dešifrovanje može izvršiti jedino vlasnik tajnog ključa svojim tajnim ključem.

Heš funkcije

Heš funkcija je jednosmerna funkcija koja komprimuje niz, što znači izlazni je kraći od ulaznog. Jednosmerna funkcija označava da funkcija nema svoju inverznu funkciju. To praktično znači da se iz početnog niza može izračunati heš vrednost, ali obrnuto nije moguće, jer ne postoji inverzna funkcija. Zapravo ne postoji strogi matematički dokaz da nije moguće ali je sigurno da je teško moguće.

I najmanja promena u ulaznom nizu uzrokuje gotovo potpunu promenu heš vrednosti [44].

Dužina izlaznog niza je fiksna u zavisnosti od algoritma koji se koristi, kao što su: SHA1, MD4, MD5... Svoju primenu pronalaze u različitim oblastima kriptografije.

Hash funkcija poznat kao SHA1 je jednostavna. Karakteriše je svojstvo da niz bilo koje dužine svede na niz dužine 160 bita. Funkcija je u svom sadašnjem obliku finalizirana 1995. kada je FIPS (*Federal Information Processing Standard*) verifikovao SHA1 algoritam.

Digitalna anvelopa

Predstavlja način zaštite poruka koji koristi dva sloja šifrovanja. Prvo, sama poruka je šifrovana korišćenjem nekog od simetričnih kriptografskih algoritama. Zatim se ključ kojim je poruka šifrovana šifrjuje javnim ključem primaoca. Šifrat se šalje zajedno sa šifrovanim ključem. Primalac najpre dešifruje simetrični ključ, kojim je poruka šifrovana svojim tajnim ključem. Nakon toga dešifruje samu poruku jer mu je tada poznat ključ kojim je poruka šifrovana.

U okviru digitalne anvelope se može naći i heš vrednost originalne poruke. Primalac nakon dešifrovanja izračunava heš vrednost dešifrovanog teksta. Upoređuje dobijenu heš vrednost sa heš vrednosti koju je dobio u poruci. Ukoliko si dve heš vrednosti identične, nije bilo intrvencija na poruci. Ukoliko se dve heš vrednosti razlikuju, znači da je poruka u toku prenosa pretrpela izmene i da dobijena poruka nije identična poslatoj [45].

Uzroci izmene poruke u prenosu mogu biti loši komunikacioni kanali koji stvaraju šum i oštećuju poruku ili napad na poruku od strane napadača koji je poruku izmenio sa određenom namerom.

Digitlani potpis

Digitalni potpis je predstavlja matematički model za potvrdu autentičnost digitalne poruke ili digitalnog dokumenta. Potpisnik koristi svoj tajni ključ za generisanje potpisa za datu poruku, dok primaoc poruke koristi javni ključ pošiljaoca/potpisnika poruke kako bi potvrdio identitet pošiljaoca. Važeći digitalni potpis daje potvrdu primaocu da je poruka kreirana od strane poznatog pošiljaoca, tako da pošiljaoc ne može poreći da je poslao poruku. Ovim su realizovani postulati potvrde identiteta i neporecivosti. Takođe se potvrđuje da poruka nije promenjena u toku prenosa kroz komunikacione kanale [46].

Imajući u vidu oznake uvedene u prethodnim razmatranjima, sa H ćemo označiti heš funkciju. Proces digitalnog potpisivanja se može prikazati i na sledeći način:

- 1) Generiše se slučajni ključ k pri čemu je $0 < k < q$
- 2) Izračuna se $r = (g^k \bmod p) \bmod q$
- 3) Ako se slučajno dobije da je $r = 0$, generiše se novi ključ i računanje se ponavlja
- 4) Izračuna se $s = k^{-1} (H(m) + x * r) \bmod q$

- 5) Ako se slučajno dobije da je $s = 0$, generiše se novi ključ i računanje se ponavlja
- 6) Digitalni potpis je (r, s)

Proces verifikacije digitalnog potpisa ima za cilj da potvrdi primaocu da je poruka došla od osobe koja se predstavila. Takođe, imajući u vidu ranije uvedene oznake, proces validacije digitalnog potpisa se može predstaviti na sledeći način:

- 1) Ukoliko nije zadovoljen uslov $0 < r < q$ ili $0 < s < q$ digitalni potpis se odbacuje
- 2) Izračuna se $\omega = s^{-1} \bmod q$
- 3) Izračuna se $u_1 = H(m) * \omega \bmod q$
- 4) Izračuna se $u_2 = r * \omega \bmod q$
- 5) Izračuna se $((g^{u_1} y^{u_2}) \bmod p) \bmod q$
- 6) Ukoliko je $v = r$ potpis je validan

Digitlani sertifikat

je elektronski dokument koji se koristi da se digitalni potpis poveže sa javnim ključem i identitetom vlasnika javnog ključa. Pod identitetom se obično podrazumeva ime osobe ili organizacije, adresa... Osnovna namena digitalnog sertifikata je da se proverí da li je javni ključ vlasništvo pojedinca ili organizacije.

U klasičnoj PKI šemi potpis će biti potpisan javnim ključem sertifikacionog tela koje je izdalo sertifikat. Sertifikat može biti i samopotpisan ukoliko se potpiše javnim ključem vlasnika sertifikata.

Ukoliko posmatramo platne kartice sa čipom, recimo Visa, svaka od banaka koja izdaje Visa digitalne sertifikate sama kreira sertifikat i samopotpiše ga, tj. potpiše ga svojim javnim ključem. Nakon toga se digitalni sertifikat šalje u Visa kompaniju koja dodatno potpiše digitalni sertifikat svojim javnim ključem i vrati ga banci. Banka zatim koristi te sertifikate na karticama sa čipom. U slučaju bankarskih kartica korisnik je ATM (*Automatic Teller Machine*) ili POS terminal (*Point Of Sale*). Na taj način softver koji obrađuje podatke sa čip kartice prepoznaje da se radi o određenoj banci i Visa platnoj kartici. Ostali parametri služe za identifikaciju pojedinca [47].

Važeća verzija formata digitalnog sertifikata je X.509 prem kojoj digitalni sertifikat sadrži sledeće podatke:

Verzija

Serijski broj

ID algoritma

Izdavaoc digitalnog sertifikata

Period važenja sertifikata

Od

Do

Vlasnik

Informacija o javnom ključu vlasnika

Algoritam koji koristi javni ključ vlasnika

Javni ključ vlasnika

Jedinstveni ID izdavaoca (opciono)

Jedinstveni ID vlasnika (opciono)

Ekstenzija (opciono)

...

2.2.5 Kriptoanaliza

Pokušaj da se kriptološki sistem kompromituje naziva se napadom. Napad može biti uspešan ili neuspešan.

Prema kriterijumu šta je napadaču poznato, vrste napada se mogu klasifikovati na sledeći način :

- 1) Poznat je samo šifrat - cifer text attack,
- 2) Poznati su parovi (otvoreni tekst - šifrat) - plain text attack. Napadač nema mogućnost da bira parove,
- 3) Poznati su parovi (otvoreni tekst - šifrat) - chosen plain text attack. Napadač ima mogućnost da bira parove,
- 4) Adaptivni napadi – napadač izračunava parametre i bira par koji mu je najpovoljniji.

Za napadače je najpovoljnije otkrivanje ključa.

Lokalni uspeh znači da se rekonstruiše jedna poruka, što predstavlja parcijalnu informaciju [48].

Prilikom određivanja sigurnosti sistema bitna je vremenska komponenta, tj. koliki je period za koji je potrebno da sistem bude siguran.

Prema sigurnosti sistemi se dele na :

- 1) Bezuslovno sigurne – postoji egzaktn dokaz da protivnik uz postojanje beskonačnih resursa nije u stanju da dođe do zaštićene informacije,
- 2) Uslovno sigurne – postoji procedura na osnovu koje se može pretpostaviti polazna poruka, ali zahteva takve resurse koje je nemoguće obezbediti. U ovom slučaju koriste se referentni sistemi poznati iz teorije.

U narednom delu teksta biće razmotreni elementi matematičkog aparata koji se koriste u svrhu kriptanalize.

Pseudoslučajni niz

Elementi koji se generišu sadrže samo 0 i 1, mada se može koristiti nad bilo kojom azbukom. Ukoliko posmatramo niz $s_0, s_1, s_2, \dots, s_n$. Postoji algoritam kojim se može izračunati član niza prema nekom od prethodnih članova:

$$s_n = A(s_0, s_1, s_2, \dots, s_{n-1}) \quad (92)$$

Golombovi postulati – osnovne postavke

Ukoliko posmatramo već pomenuti niz $s_0, s_1, s_2, \dots, s_n$ i pretpostavimo da se radi o pseudoslučajnom nizu. Odnosno pretpostavimo da posmatrani član niza zavisi od nekog od prethodnih članova. N-ti član niza se tada se može izračunati na osnovu prethodnih članova niza i to:

$$s_N = A(s_{N-1}, s_{N-2}, \dots, s_{N-K}) \quad (93)$$

$$s_{N+1} = A(s_N, s_{N-1}, \dots, s_{N-K+1}) \quad (94)$$

Zaključujemo da N-ti element niza zavisi od K prethodnih elemenata.

Posle nekog vremena se elementi ponavljaju :

$$s_M = A(s_{M-1}, s_{M-2}, \dots, s_{M-K}) \quad (95)$$

$$s_M = A(s_M, s_{M-1}, \dots, s_{M-K+1}) \quad (96)$$

Može se zaključiti da je pseudoslučajni niz periodičan, odnosno:

$$\{S_n\}_{n=0}^{\infty} \mid \exists t \in \mathbb{N} \mid \forall n > 0; S_n = S_{n+t} \quad (97)$$

Najmanji od brojeva t je osnovni perod a ima ih $k * t$, što znači beskonačno mnogo.

Serija

U binarnom nizu mogu postojati serije uzastopnih članova, serije nula ili serije jedinica. Serije nula su rupe ili razmaci (gapovi) dok su serije jedinica blokovi.

Autokorelaciona funkcija $c(t)$

Autokorelaciona funkcija se definiše sledećim izrazom:

$$c(t) = \frac{1}{N} \sum_{i=0}^{N-1} (2s_i - 1) * (2s_{i+t} - 1) \quad (98)$$

Pri čemu je $0 \leq t \leq 1$.

Korišćenjem izraza $\sum s_i * s_{i+t}$, može se dobiti informacija na koliko mesta se poklapaju

jedinice nakon translacije niza, što nije od naročite koristi.

Autokorelacionu funkciju ćemo, radi jasnoće, razmotriti na konkretnom primeru.

Pretpostavimo da imamo niz u kome se određene sekvence ponavljaju. Sada transliramo ceo niz za dva mesta udesno.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	1	0	1	0	1	1	0	1	0	1	0
niz koji se ponavlja							niz koji se ponavlja						
	1	1	0	1	0	1	0	1	1	0	1	0	

Tabela 3. Autokorelaciona funkcija - primer

U navedenom primeru je $N = 7$ i $t = 2$, što znači da se ponavlja 7 elemenata a pomereni su za 2 mesta udesno.

Golombovi postulati

Za niz $s_0, s_1, s_2, \dots, s_n$ sa periodom N važi:

- 1) n_0 je broj nula na periodu
 n_1 je broj jedinica na periodu
 Pri čemu je $|n_0 - n_1| \leq 1$
- 2) Najmanje polovina serija ima dužinu 1
 Najmanje četvrtina serija ima dužinu 2
 ...
 Najmanje $\frac{1}{2^n}$ serijaima dužinu 2^{n-1}

$$3) c(t) = \begin{cases} N, & t = 0 \\ K, & 1 \leq t \leq N - 1 \end{cases}$$

Ako niz S^N zadovoljava uslove od 1 do 3, takav niz nazivamo pseudoslučajnim nizom.

Golombovi postulati utvrđuju isključivo ispunjenost ulaznih parametara, odnosno da li se radi o pseudoslučajnom nizu.

Radi utvrđivanja kvaliteta pseudoslučajnog niza definišu se testovi (statističko ponašanje bigrama, trigrama,...; poker test; autokorelacioni test,...) ili u oznaci $\{T_1, T_2, \dots, T_n\}$

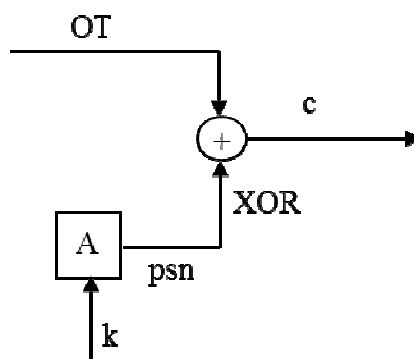
Test je zapravo algoritam kojim se utvrđuje zakonitost niza.

Moderne kriptografske tehnike

Tehnološki razvoj dovodi i do razvoja kriptografskih tehnika. Hardverski kapaciteti se eksponencijalno uvećavaju, čime se nekada neprobojni algoritmi sa lakoćom i u veoma kratkom vremenu razbijaju. U prethodnom delu teksta pomenuto je razbijanje DES algoritma.

Ipak, mnogi zanemaruju matematičke modele koji doprinose razbijanju veoma kompleksnih kriptografskih sistema. Ne treba smetnuti sa uma da snažni hardverski resursi ponekad nisu dovoljni za kriptanalizu i da se korišćenjem matematičkih metoda mogu definisati znatno efikasniji sistemi kriptanalize.

Proces šifrovanja se može šematski predstaviti kao na slici 33.



Slika 33. Šematski prikaz procesa šifrovanja

Proces šifrovanja se može predstaviti sledećom funkcijom:

$$f(OT, A(k)) \quad (99)$$

Postavlja se pitanje zašto se ne uzme neka komplikovanija funkcija od XOR funkcije ?

Pokazalo se da ako možemo razbiti sistem za XOR možemo i za svaku drugu funkciju i obrnuto, tako da se ništa ne dobija upotrebom složenije funkcije. Naprotiv, ako je funkcija složenija produžava se vreme šifrovanja.

Takođe se pokazalo da svaki govorni jezik ima izražene svoje statističke osobine, što se može iskoristiti za razbijanje sistema.

Ukoliko je poznat pseudoslučajni niz zaključujemo da nije potrebno znati ključ da bi se razbio sistem.

Metod Kasiskog

Neka je ključ za šifrovanje k i neka je pseudoslučajni niz $a_1, a_2, \dots, a_t$. Neka pri tome postoje dva otvorena teksta $m_1, m_2, \dots, m_t$ i $n_1, n_2, \dots, n_t$.

Ukoliko posedujemo dva otvorena teksta šifrovana istim ključem mogu se dobiti oba

otvorena teksta:

$$c_1^m = a_1 \oplus m_1; c_2^m = a_2 \oplus m_2; \dots \quad (100)$$

$$c_1^n = a_1 \oplus n_1; c_2^n = a_2 \oplus n_2; \dots \quad (101)$$

Odnosno dalje, nakon primene XOR operacije, dobija se sistem jednačina:

$$c_1^m \oplus c_1^n = m_1 \oplus n_1 \quad (102)$$

$$c_2^m \oplus c_2^n = m_2 \oplus n_2 \quad (103)$$

...

Dalje se rešava dobijeni sistem jednačina (102), (103)...

Može se zaključiti nikada ne treba dva otvorena teksta ne šifrovati istim ključem, već je dobro da se posle svakog uspešnog šifrata ključ uništava i generiše se novi.

Algoritmi kojima se generišu pseudoslučajni nizovi su veoma složeni. Kvalitet ovog algoritma direktno utiče na sigurnost šifrata.

Linearni kongruentni generator

predstavlja jedan od načina generisanja pseudoslučajnog niza.

Zadaje se početna vrednost niza s_0 , dok se ostali članovi niza izračunavaju na sledeći način:

$$s_{n+1} = m(a * s_n + b) \quad (104)$$

Poželjno je da period bude što duži, pri čemu je očigledno da najduži može biti n .

Teorema:

- 1) $(b, m) = 1$, b i m su uzajamno prosti
- 2) $p|m \Rightarrow p|(a - 1)$
- 3) $4|(a - 1) \Rightarrow 4|m$

Obrnuto, kada napravimo uzastopne razlike možemo pogoditi a :

$$s_{n+1} - s_n = m(a - 1) * (s_n - s_{n-1}) \quad (105)$$

Kriptoanaliza predstavlja proučavanje metoda za dešifrovanje šifrovanih informacija, bez posedovanja tajnih podataka koji su obično potrebni da bi se iz šifrata dobio otvoreni tekst. Kriptoanaliza obično podrazumeva pronalaženje tajnog ključa. Kriptoanaliza se takođe koristi da označi svaki pokušaj zaobilaženja drugih tipova kriptografskih algoritama i protokola uopšte. Međutim, kriptoanaliza obično ne razmatra metode napada čija primarna meta nisu slabosti posmatranog kriptografskog sistema, kao što su podmićivanje, fizička sila, provaljivanje ili socijalno inženjerstvo, mada ovi tipovi napada jesu važna stavka koja se svakim danom usavršava i češće dovodi do rezultata nego tradicionalna kriptoanaliza.

Iako je cilj kriptoanalize bio i ostao isti, metode i tehnika kriptoanalize su se vremenom

radikalno promenile. Tehnike se prilagođavaju sve većoj kompleksnosti kriptografije, počevši od metoda ručnog izračunavanja, preko izuzetne mašine kao što je nemačka Enigma za vreme drugog svetskog rata, do korišćenja savremenih računarskih sistema. Sredinom sedamdesetih godina dvadesetog veka uveden je novi tip kriptografije, asimetrična kriptografija. Metode za razbijanje ovih kriptosistema se znatno razlikuju od razbijanja simetričnih šifarskih sistema.

Nije zvanično potvrđeno, ali postoje informacije da je NSA u stanju da razbije 90% asimetričnih parova ključeva. Ova informacija, makar i nepotvrđena znak je da je došlo vreme za prelazak na nove generacije kriptografskih sistema.

3. Kodovi za korekciju greške

Prilikom prenosa poruke kroz komunikacione kanale u kojima postoji šum, najčešće dolazi do pojave greške u samoj poruci. Greške u prenosu su neminovne. Dalje se postavlja pitanje njihovog uočavanja odnosno detekcije kao i pitanje njihovog korigovanja. Cilj je da primalac dobije poruku koja je identična ili bar dovoljno bliska polaznoj poruci. Na taj način će primalac shvatiti značenje poruke.

Oblast primene kodova za detekciju i korekciju greške se bavi uočavanjem grešaka u izlaznoj poruci i načinima otklanjanja uočenih grešaka. Komunikacioni kanal sa šumom može izazvati različite greške uključujući promene pojedinih bitova kao i njihovo brisanje. Očigledno je da nedostatak bitova predstavlja problem koji zahteva kompleksniji pristup njegovom rešavanju.

Pretpostavićemo da se poruka sastoji od k bitova $x_1, x_2, \dots, x_k$. Pretpostavićemo da se radi o binarnoj azbuci, odnosno da se niz sastoji isključivo od nula i jedinica. Iako se analogna razmatranja koja slede mogu izvesti na bilo kojoj azbuci, činjenica je da se elektronski signali svode na binarni sistem odnosno na niz nula i jedinica [49].

Enkoder pretvara poruke u kodne reči fiksne dužine, recimo k bita. Zatim se ta poruka šalje kroz komunikacione kanale sa šumom. Šum ometa poruku i dovodi do izmene ili gubitka pojedinih bitova. Dekoder prevodi binarni niz u poruku koja u sebi sadrži više ili manje grešaka. Cilj kodova za detekciju i korekciju greške je da greške najpre uoči a zatim i ispravi u najvećoj mogućoj meri.

Ukoliko se posmatra govorni signal koji se najpre iz analognog prevodi u digitalni sistem činjenica je da se sam govor obrađuje u ljudskom mozgu. Ljudski um može da prepozna govor čak i ukoliko postoje šumovi ili smetnje u govornom signalu. Treba imati u vidu činjenicu da je ljudski mozak u stanju da razume govorni signal sa greškama do određene mere. Ukoliko su greške u prenosu veće od određene granice govor postaje nerazumljiv, i ljudski um ga ne može shvatiti.

Imajući u vidu činjenicu da će u narednom delu teksta biti reči o greškama koje se javljaju prilikom prevođenja biometrijske veličine u binarni niz, očigledno je da se radi o specifičnom problemu. Pogotovo ako se uzme u obzir da je svrha korišćenja dobijenog binarnog niza u kriptografske svrhe, problem postaje još kompleksniji.

U prethodnim poglavljima su opisane osnovni pojmovi kriptografije iz kojih se može zaključiti da nisu dozvoljene čak ni minimalne greške u kriptografskim ključevima. U suprotnom, kriptografski ključ je neupotrebljiv. U narednom delu teksta biće predloženo

rešenje koje u najvećoj mogućoj meri rešava ovaj problem.

Pre izlaganja rešenja biće predstavljene osnovne postavke kodova za korekciju greške. S obzirom da postoji više pristupa korekciji greške, biće prikazani karakteristični pristupi kao i pristup korišćen u predloženom rešenju.

3.1 Šenonova teorema

Šenonova teorema predstavlja jednu od fundamentalnih teorema u oblasti teorije infomacija [50].

Posmatra se poruka koja se sastoji od simbola $x_1, x_2, \dots, x_n$ pri čemu je verovatnoća pojavljivanja svakog od simbola $p_1, p_2, \dots, p_n$. Definiše se entropija skupa navedenih verovatnoća kao:

$$E = - \sum_{i=1}^n p_i * \log_2 p_i \quad (106)$$

Definicija:

Prefiks binarni se odnosi na binarne nizove koji se sastoje od nula i jedinica koji čine takozvane kodne reči. Pri tome ne postoji kodna reč koja se javlja kao početak druge kodne reči. Treba imati na umu da ukoliko poznajemo dužini kodne reči, poruka može jednoznačno da se dekodira.

Teorema:

Za svaki binarni niz koji se kodira $x_1, x_2, \dots, x_n$ prosečna dužina kodne reči mora biti veći od E . Tačnije, izabran dužina l_i kodne reči x_i mora da zadovolji sledeću jednačinu:

$$\sum_{i=1}^n p_i * l_i \geq E \quad (107)$$

Za dokaz ove teoreme koriste se dve leme:

Lema 1:

Gibsova nejednakost

Pretpostavimo da je $p_1, p_2, \dots, p_n$ posmatrana raspodela verovatnoća. Tada za svaku drugu raspodelu verovatnoća $q_1, q_2, \dots, q_n$ sa istim brojem elemenata važi:

$$\sum_{i=1}^n p_i * \log_2 p_i \geq \sum_{i=1}^n p_i * \log_2 q_i \quad (108)$$

Lema 1:

Kraftova nejednakost

Neka su $x_1, x_2, \dots, x_n$ simboli posmatrane azbuke i da su kodirani binarno. I neka su $l_1, l_2, \dots, l_n$ dužine kodnih reči odgovarajućih $x_1, x_2, \dots, x_n$ tada je:

$$\sum_{i=1}^n 2^{-l_i} \leq 1 \quad (109)$$

Dokaz Šenonove teoreme:

Imajući u vidu Leme i ne ulazeći u njihove dokaze, uvodimo novi broj $\frac{1}{C}$ takav da $C2^{-l_1}, C2^{-l_2}, \dots, C2^{-l_n}$ predstavlja raspodelu verovatnoća.

Tada je:

$$\sum_{i=1}^n p_i \log_2 p_i \geq \sum_{i=1}^n p_i \log_2 (C2^{-l_i}) = \sum_{i=1}^n p_i (\log_2 C - l_i) = \log_2 C \sum_{i=1}^n p_i l_i \quad (110)$$

i dalje je:

$$\sum_{i=1}^n p_i l_i \geq - \sum_{i=1}^n p_i \log_2 p_i + \log_2 C \geq - \sum_{i=1}^n p_i \log_2 p_i \quad (111)$$

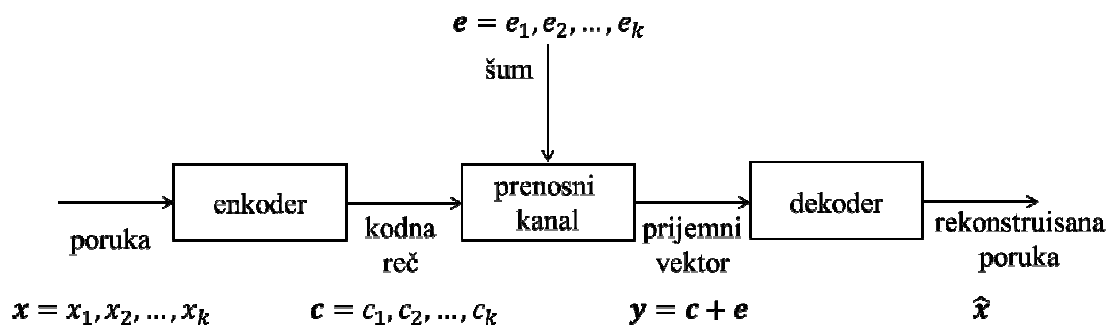
Čime je Šenonova teorema dokazana.

Teorija kodiranja je nastala u kasnim 1940-tim godinama, vremenom se razvijala i postala deo matematike a naročito informatike. Kodovi su prvobitno razvijeni za ispravljanje grešaka u komunikacionim kanalima sa šumom, koji su na svom izlazu reprodukovali podatke sa greške. Za korekciju pomenutih grešak linearni kodovi su veoma korisni. Linearni kodovi su jednostavni za razumevanje i nalaze svoju primenu u praktičnim primenama kodiranja i dekodiranja poruke poslate preko komunikacionih kanala. Klod Šenon (*Claude Elwood Shannon, 1916. – 2001.*) je 1948., došao do rezultata i jednu od osnovnih teorema u teoriji kodiranja. Teorema tvrdi da je moguć prenos podataka bez greške maksimalnom brzinom u zavisnosti od šuma u komunikacionom kanalu, što je, teorijski posmatrano, brzina prenosa putem posmatranog kanala [51].

3.2 Osnovni koncepti lineranih kodova

Imajući u vidu činjenicu da komunikacioni kanal može biti izložen smetnjama u tom slučaju dolazi do oštećenja signala koji se kroz posmatrani kanal prenosi. Šenon je izračunao broj koji je nazvao kapacitetom kanala i dokazao da je moguć proizvoljno veliki broj komunikacija kroz posmatrani kanal sve dok je broj komunikacija manji od kapaciteta kanala. Šenonovi rezultati ustanovljavaju da podaci mogu biti kodirani pre prenosa kroz

komunikacioni kanal a zatim dekodirani, nakon prolaska kroz kanal. Dekodiranje rekonstruiše originalnu poruku do određenog stepena tačnosti.



Slika 34. Šematski prikaz prenosa poruke kroz kanal sa šumom

Primeri primene nisu samo komunikacioni kanali već i binarni zapisi na diskovima ili u primeni na bilo kojim telekomunikacionim uređajima kao što su mobilni telefoni. Primena koja je predmet ovog rada odnosi se na biometrijske podatke i to na njihovu akviziciju, skladištenje, prenos i rekonstrukciju [52].

Šum se može opisati i kao pojava da podaci koji su poslani nisu istovetni onima koje dobijamo na izlazu iz komunikacionog kanala. Ukoliko posmatramo binarni niz, i ako je polazni bit nula može se očekivati da je i primljen bit jednak nuli. Ako postoji šum, primljeni bit može biti nula, ali isto tako i jedinica ili može biti neodređen odnosno neprepoznatljiv.

Osnovni problem koji rešavaju kodovi za korekciju greške je određivanje polazne poruke na osnovu primljene poruke, uz saznanje da je došlo do izmene poruke u prenosu.

Posmatrajući Sliku 34., u idelnom slučaju, kad ne bi bilo šuma, bilo bi $\hat{x} = x$. Šenonova teorema pruža dokaze da je to moguće u određenom vremenskom intervalu ili u određenom procentu količine podataka. Sa izborom korektnog sistema za kodiranje, karakteristike prenosnog kanala se mogu znatno poboljšati. Analogno, prilikom prevođenja biometrijskih podataka u binarni niz, izborom korektnog sistema za kodiranje, greška prilikom pomenutog postupka se može znatno smanjiti. Iako, istini za volju, ne može se greška svesti na nulu.

Šenonova teorema tvrdi da postoji način za postizanje željene tačnosti u komunikacionom kanalu ali ne definiše koji je to kod kojim se može postići željeni nivo greške za određeni komunikacioni kanal. Cilj istraživanja teorije kodiranja je kreiranje kodova koji zadovoljavaju Šenonovu teoremu. U ovom radu će biti predloženi kodovi za korekciju greške koji bi mogli biti korisni za otklanjanje grešaka prilikom reprodukovanja otiska prsta kao biometrijskog podatka.

Linerani kodovi su najviše proučavani kodovi. Zbog algebarske strukture lineranih kodova njih je lakše matematički opisati nego kodiranje i dekodiranje nelinearnih kodova. Polje elemenata za linerane kodove je konačno.

Hamingovo rastojanje

Ričard Haming (*Richard Wesley Hamming, 1915. - 1998.*) je američki matematičar koji je ostvario značajne doprinose na polju informatike i telekomunikacija [53].

U narednom delu teksta biće definisano Hamingovo rastojanje na polju $\mathbb{F}$ koje ukupno ima q elemenata. Kodna reč može sadržati n elemenata pri čemu je očigledno $n \leq q$, ili u oznaci $(x_1, x_2, \dots, x_n) \in \mathbb{F}_q^n$.

Neka su $\mathbf{x} = (x_1, x_2, \dots, x_n)$ i $\mathbf{y} = (y_1, y_2, \dots, y_n)$ dva jednodimenziona vektora veličine n . Hamingovo rastojanje između ova dva vektora se označava sa $d(\mathbf{x}, \mathbf{y})$ pri čemu za oba vektora važi $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$. Hamingovo rastojanje predstavlja broj odgovarajućih elemenata koji se međusobno razlikuju, respektivno po koeficijentima.

Za Hamingovo rastojanje važe sledeće pretpostavke:

- 1) $d(\mathbf{x}, \mathbf{y}) \geq 0$ i $d(\mathbf{x}, \mathbf{y}) = 0$ ako i samo ako je $\mathbf{x} = \mathbf{y}$
- 2) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
- 3) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ za bilo koje $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$

Najbliži susedni element

Za dati kod $C \subset \mathbb{F}_q$ i vektor $\mathbf{y} \in \mathbb{F}_q$ tada je $\mathbf{x} \in C$ najbliži sused vektoru $\mathbf{y}$ ako je

$$d(\mathbf{x}, \mathbf{y}) = \min(d(\mathbf{z}, \mathbf{y}) | \mathbf{z} \in C) \quad (112)$$

Potrebno je primetiti da vektor može imati više od jednog najbližeg suseda, što znači da najbliži sused nije uvek jedinstven.

Matrica jednakosti

Matrica generator generiše sve moguće kodne reči nad poljem $\mathbb{F}_q$.

Matrica dualnih kodova je definisana izrazom:

$$C^\perp = \{\mathbf{x} \in \mathbb{F}_q^n | \langle \mathbf{x}, \mathbf{c} \rangle = 0 \forall \mathbf{c} \in C\} \quad (113)$$

Matrica jednakosti H za kod C je matrica generator dualnih kodova $C^\perp$. Može se zaključiti da je matrica $C^\perp$ dimenzija $(n - k) * n$.

Neka je $r \geq 1$ prirodan broj i neka je $n = 2^r - 1$. Neka je dalje H matrica, dimenzija $m * n$ čije su kolone vektori polja $\mathbb{F}_2^n$ koji nisu nule.

Kod $C = \text{Ham}(r)$ čija je matrica provere jednako sto H , naziva se Hamingovom kodom dužine n .

Osnovna algebarska struktura koju ćemo koristiti je konačno polje sa dva elementa, $\mathbb{F}_2 = \{0, 1\}$. Sabiranje i množenje nad ovim poljem je definisano sledećim pravilima:

+	0	1
0	0	1
1	1	0

*	0	1
0	0	0
1	0	1

Tabela 4. Sabiranje i množenje nad konačnim poljem $\mathbb{F}_2 = \{0, 1\}$

Ciklični kodovi

Klasa cikličnih kodova uključuje i Hamingove kodove.

Definicija

Kod $C \subseteq \mathbb{F}_q^n$ je ciklični kod ako je C linarno i ako za svaku kodnu reč $(c_0 c_1 \dots c_{n-1}) \in C$ postoji $(c_{n-1} c_0 c_1 \dots c_{n-2})$ koja je takođe kodna reč.

To znači da se za svaku cikličnu permutaciju kodne reči ponovo dobija kodna reč. U oznaci:

$$\pi: (c_0 c_1 \dots c_{n-1}) \mapsto (c_{n-1} c_0 c_1 \dots c_{n-2}) \quad (114)$$

Prikazano svojstvo se naziva automorfizmom kodne reči C .

3.3 Druge klase kodova

U narednom delu teksta biće razmotreni neki od karakterističnih kodova koji se mogu koristiti u svrhu korekcije reprezentata biometrijskih podataka.

Rid-Milerovi kodovi

Kodovi su dobili ime po američkim matematičarima Ridu i Solomonu (*Irving Stoy Reed, 1923. – 2012; David Eugene Muller, 1924. – 2008.*). Oznaka za Rid-Milerove kodove je $RM(r, m)$ i mogu se definisati na sledeći način:

- 1) $RM(0, m) = \{0 \dots 0, 1 \dots 1\}$ je kod dužine 2^m koji se sastoji od reči koje čine sve nule i sve jedinice.
- 2) $RM(m, m) = \mathbb{F}_2^{2^m}$
- 3) $RM(r, m) = RM(r, m-1) * RM(r-1, m-1)$ za $0 < r < m$

Rid-Milerovi kodovi su zasnovani na sledeći teoremama:

Teorema

Kod $RM(r, m)$ je linarni kod dužine 2^m dimenzija $\binom{m}{0} + \binom{m}{1} + \dots + \binom{m}{r}$ sa minimalnim

rastojanjem 2^{m-r} .

Teorema

Kod $RM(1, m)$ je dualni kod $Ham^+(m)$ što je pariti ekstenzija Hamingovog koda $Ham(m)$.

Teorema

Kod $RM(r, m)^\perp = RM(m - r - 1, m)$

Na ovom mestu neće se ulaziti u dokaze navedenih teorema.

Rid-Solomonovi kodovi

Rid-Solomonovi kodovi (*Gustave Solomon, 1930. - 1996.*) su kombinacija BCH i cikličnih kodova. Bitna karakteristika BCH kodova je da ostvaruju kontrolu nad brojem grešaka za sve vreme procesa kodiranja. Ime predstavlja akronim dobijen od imena autora koda: *Raj Chandra Bose, 1901. - 1987.; Dwijendra Kumar Ray-Chaudhuri, 1933.; Alexis Hocquenghem, 1908. - 1990.* Rid-Solomonovo kodiranje je linearno blok kodiranje. Definisani su kao $RS(n, k)$ sa s -bitnim simbolima.

Definicija

Za ceo broj k koji zadovoljava uslov $1 \leq k < n$ nad poljem $\mathbb{F}$ čija je veličina $|\mathbb{F}| \geq n$ i pri čemu je skup $S = \{\alpha_1, \alpha_2, \dots, \alpha_{n-1}\} \subseteq \mathbb{F}$ definišu se Rid-Solomonovi kodovi:

$$RS_{\mathbb{F}, S}[n, k] = \{(p(\alpha_1), p(\alpha_2), \dots, p(\alpha_{n-1})) \in \mathbb{F}^{n-1} \mid p \in \mathbb{F}[X]\} \quad (115)$$

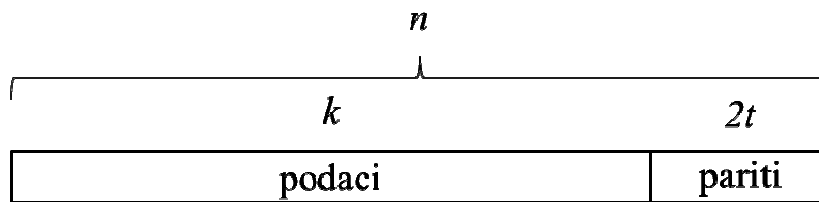
Pri čemu je $\mathbb{F}[X]$ polinom stepena $\leq k - 1$.

Radi kodiranja poruke $m = (m_0, m_1, \dots, m_{k-1}) \in \mathbb{F}^k$ poruka se predstavlja u vidu sledećeg polinoma:

$$p(X) = m_0 + m_1X + \dots + m_{k-1}X^{k-1} \in \mathbb{F}[X] \quad (116)$$

Dalje se vrši razvoj polinoma p u tačkama $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$ kako bi se dobile kodne reči koje odgovaraju poruci m .

Pojednostavljeno rečeno, kada enkoder uzme k simbola podatka od s bitova, svakom dodaje pariti bitove kako bi napravio kodne reči od n simbola. U svakom bloku postoki $n - k$ pariti simbola u svakom od s -bitova. Dekoder može da vrši korekciju do t simbola u kodnoj reči gde je $2t = n - k$. Podatak sa leve strane je nepromenjen dok se pariti simboli dodaju sa desne strane.



Slika 35. Struktura bloka podataka kod Rid-Solomonovog kodiranja

Greška u kodnoj reči se može manifestovati tako što je pogrešan jedan bit u kodnoj reči, ili mogu biti pogrešni svi bitovi.

Kada se kodna reč dekodira postoje tri moguća rezultata:

- 1) Ako je $2s + r < 2t$ gde je s broj grešaka a r broj obrisanih bitova, originalna kodna reč će uvek biti nadoknađena.
- 2) Dekoder neće moći da nadoknadi originalnu kodnu reč i daće informaciju o toj činjenici.
- 3) Dekoder će pogrešno dekodirati i nadoknadiće netačnu kodnu reč bez ikakvih informaciju o toj činjenici.

Rid-Solomonovo dekodiranje može vršiti i korekciju grešaka koje za posledicu imaju brisanje bitova. Uspešno otklanjanje ove greške se vrši pod pretpostavkom da se pozicija obrisanog simbola zna. Dekoder može ispraviti do t grešaka, ili do $2t$ brisanja. Obrisana informacija se najčešće mogu zameniti u demodulatoru kada posmatramo digitalne komunikacije.

Rid-Solomonovi kodovi su se odlično pokazali i veoma efikasnim za korekciju takozvanih rafalnih ili uzastopnih grešaka (*burst errors*) koji predstavljaju seriju bitova u kodnoj reči koji su primljeni kao greška.

Adamarovi kodovi

Adamarovi kodovi (*Jacques Salomon Hadamard, 1865. - 1963.*) su se u praksi pokazali kao veoma efikasni prilikom uklanjanja pojedinačnih pogrešnih bitova.

Definicija

Kvadratna matrica H_n dimenzija $n * n$ naziva se Admarovom matricom ako su njihovi članovi iz skupa $\{-1,1\}$ i ako je $H_n * H_n^T = nI_n$.

Postoji nekoliko načina kako se može konstruisati Adamarova matrica. Sagledaćemo konstrukciju koja funkcioniše za svako n koje je stepen broja 2, odnosno zadovoljena je jednakost $n = p^m + 1$. Konstrukcija Adamarove matrice se može predstaviti na sledeći način:

1) Definiše se $H_1 = (1)$

2) Za n koje je stepen broja 2, definiše se: $H_{2n} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}$

Pretpostavka

Označe se redovi u Adamarovoj matrici H_n sa $(h_1, h_2, \dots, h_n)$. Za svako $i \neq j$ tada je $d(h_i, h_j) = \frac{n}{2}$.

Definicija

Neka je H_n Adamarova matrica. Adamarov kod dužine n ili u oznaci $H ad_n$, je binarni kod izveden od H_n kada se zamene sve vrednosti -1 sa nulama u H_n . Tada se uzimaju sve vrednosti redova H_n i njihovih komplemenata kao kodne reči.

Dve Adamarove matrice se nazivaju ekvivalentnim ili izomorfimnim ako se transformacija iz jedne u drugu matricu. Potrebno je da sve se promeni znak svih kolona koje počinju sa -1 . Ako nakon te zamene svi redovi, respektivno, počinju sa -1 dobija se ekvivalentna Adamarova matrica čiji jedino prvi red i prva kolona sadrži samo $+1$. Takve matrice se nazivaju normalizovanim Adamarovim matricama.

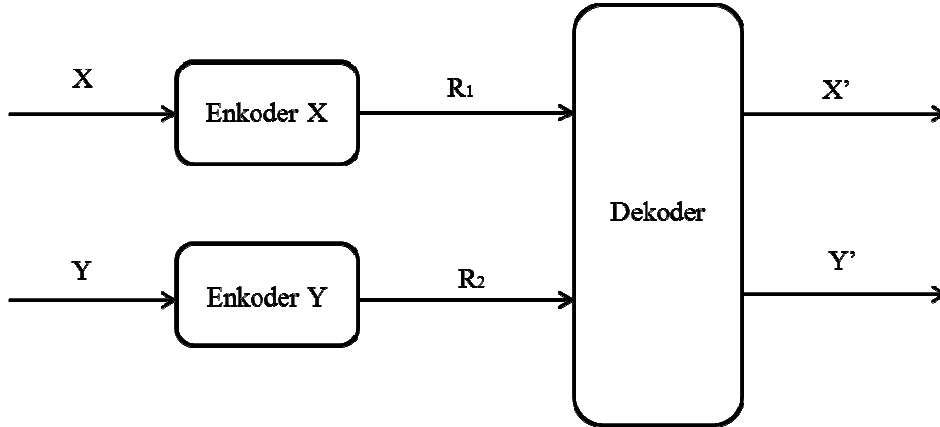
U narednom delu teksta će Adamarovi kodovi biti razmatrani u smislu primene nad biometrijskim podacima

3.4 Slepian-Vulfovi kodovi

Kodovi su dobili ime po svojim autorima Dejvidu Slepianu i Džeku Vulfu (*David S. Slepian, 1923.- 2007.; Jack K. Wolf, 1935. - 2011.*) [54]. Slepian-Volf a teorema se bavi kompresijom podataka bez gubitaka informacija. Pretpostavlja se da postoje dva ili više tokova podataka koji se nalaze u međusobnoj korelaciji. Teorema se može pronaći u više varijacija pri čemu je najpoznatija od njih varijacija u kojoj se svaki od korelisanih kanala koji se kodiraju, kodiraju odvojeno a zatim zajednički dekodiraju. Ova osobina će biti iskorišćena kao jedna od osnovnih ideja za predlog primene Slepian-Vulfovog kodiranja otiska prsta u ovom radu. U narednim poglavljima će ideja biti detaljan predlog rešenja. Slepian-Vulfovo kodiranje predstavlja oblik distribuiranog izvora kodiranja. Slepian-Vulfova teorema ima praktične posledice za sisteme gde su u tokovi podataka, iako u korelaciji, fizički odvojeni. U opštem slučaju, posmatramo dva predajnika koji emituju signale putem kanala sa šumom. Očigledno je da su signali koji stižu do prijemnika u korelaciji iako različiti zbog uticaja šuma. Prilikom prijema signala na prijemnoj strani oba primljena signala se kombinuju u jedan pri čemu se signali iz dva kanala mogu koristiti kao jedan za korekciju drugog.

Slepian-Vulfova teorema

Efikasnost sistema se meri stopama kodiranih bitova po simbolu na izlazu iz enkodera. Slepian-Vulfova teorema omogućava definisanje skupa različitih stopa kodiranih bitova kako bi dekodeer mogao da rekonstruiše posmatrana dva podatka sa proizvoljno malom greškom



Slika 36. Jedna varijacija Slepian-Vulfovog kodiranja

Kao što se može videti sa Slike 36., enkoder X posmatra ulaznu veličinu X , zatim šalje poruku dekoderu u vidu broja iz skupa $\{1, 2, \dots, 2^{n_{R_1}}\}$. Analogno, enkoder Y posmatra ulaznu veličinu Y , zatim šalje poruku dekoderu u vidu broja iz skupa $\{1, 2, \dots, 2^{n_{R_2}}\}$. Ulazi iz oba enkodera ulaze u isti dekodeer. Dekoder kao izlaz daje dva vektora $\hat{X}$ i $\hat{Y}$ koji izračunavaju X i Y respektivno. Predmet sagledavanja je verovatnoća da $\hat{X}$ nije jednako X kao i da $\hat{Y}$ nije jednako Y , kao i mogućnost smanjenja verovatnoće greške. Za takav sistem se može reći da je ostvariv doke se par stopa kodiranih bitova (R_1, R_2) nazivaju ostvarivim stopama. Oblast u kome je par (R_1, R_2) ostvariv definiše se sledećim uslovima:

$$1) \quad R_1 + R_2 > H(X, Y) \quad (117)$$

$$2) \quad R_1 > H(X|Y) \quad (118)$$

$$3) \quad R_2 > H(Y|X) \quad (119)$$

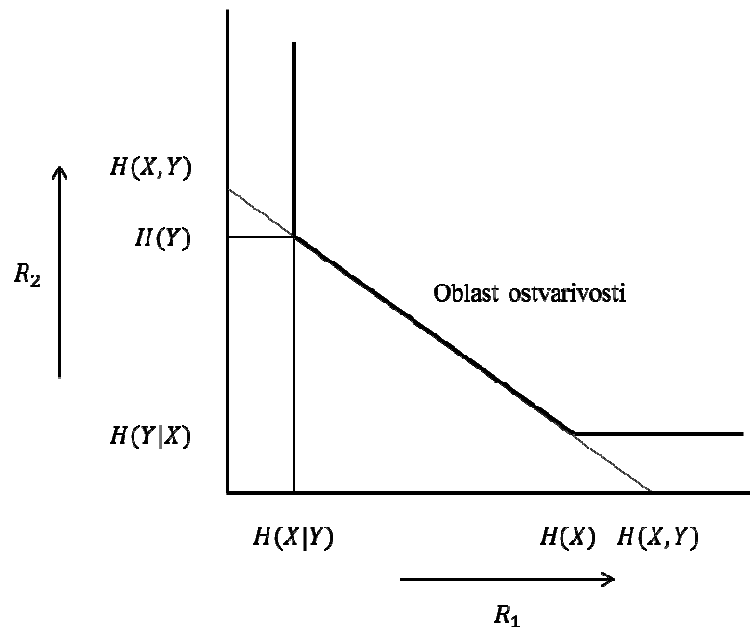
Zaključujemo da u slučaju poznatog Y , X ne može biti komprimovano iznad vrednosti $H(X|Y)$ i uvek je manje od $H(X)$. Analogno važi u slučaju poznatog X , Y ne može biti komprimovano iznad vrednosti $H(Y|X)$ i uvek je manje od $H(Y)$.

Značaj Slepian-Vulfove teorme [55] se može uočiti upoređivanjem entropije prilikom korišćenja jednog izvora podataka. Različiti enkoderi koji zanemaruju korelaciju mogu postići stope kodiranih bitova najviše do:

$$R_1 + R_2 \geq H(X + Y) \quad (120)$$

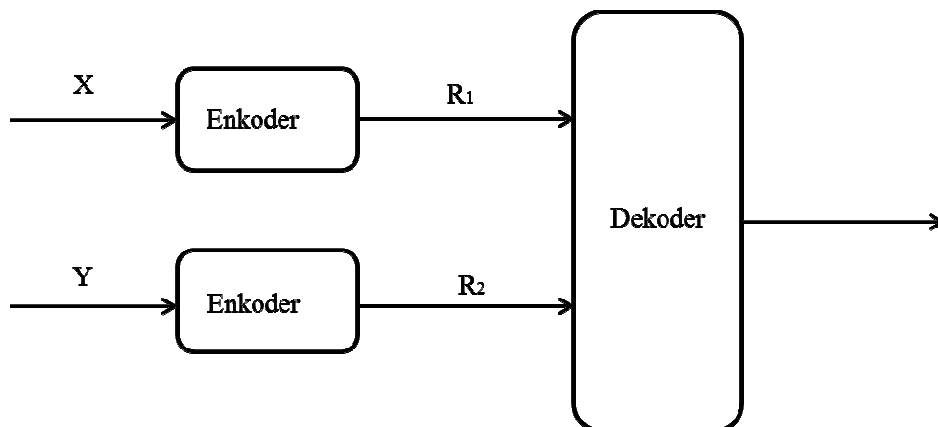
Odvojeni enkoderi koriste svoje saznanje o korelaciji između ulaznih veličina i pri tome mogu dostići istu stopu kodiranih bitova koja bi se mogla postići kao optimalna korišćenjem samo jednog enkodera ili:

$$R_1 + R_2 \geq H(X, Y) \quad (121)$$



Slika 37. Grafički prikaz oblasti ostvarivosti

Kao što je napomenuto, način kodiranja Slepian-Vulfovima se može javiti u vidu više varijacija. Ukoliko se radi o otisku prsta, što je predmet proučavanja i predloženog rešenja u ovom radu, kodiranje se može predstaviti na sledeći način:



Slika 38. Varijacija Slepian-Vulfovog kodiranja

Entropija

Sa X se označava blok od n uzastopnih, nezavisnih karaktera koji potiču od izvora informacija. Distribucija svakog od karaktera je $p_X(x)$. U tipično dugačkom bloku, znak 1 se pojavljuje $np_X(1)$ puta, znak 2 se pojavljuje $np_X(2)$ puta itd. Verovatnoća opisane tipično dugačke sekvence je:

$$p_T = p_X(1)^{np_X(1)} * p_X(2)^{np_X(2)} * ... * p_X(M)^{np_X(M)} = \quad (122)$$

$$= e^{(np_X(1)*\log_{p_X}(1))} * e^{(np_X(2)*\log_{p_X}(2))} * ... * e^{(np_X(M)*\log_{p_X}(M))} = \quad (123)$$

$$= e^{-nH(X)} \quad (124)$$

gde je:

$$H(X) = - \sum_{i=1}^M p_X(i) \log_{p_X}(i) \quad (125)$$

Entropija slučajne promenljive X .

U dalijim izlaganjima će biti predstavljena primena opisanog matematičkog alata za potrebe šifrovanja i zaštite otiska prsta kao biometrijskog podatka.

4. Kreiranje templejta i upoređivanje dva otiska prsta

Upoređivanje dva otiska prsta podrazumeva upoređivanje njihovih karakteristika. Na osnovu dobijenih rezultata, potrebno je izračunati stopu podudarnosti dva otiska prsta. Odluku o tome kolika je neophodna stopa podudarnosti da bi se dva otiska smatrala identičnim donosi sam korisnik procesa u zavisnosti od namene upoređivanja.

U ovom poglavlju će biti prikazana dva moguća načina kreiranja šablona i upoređivanja otisaka koji će kasnije biti iskorišćeni prilikom obrazlaganja predloženog rešenja u ovom radu.

4.1 Upoređivanje latentnih otisaka

Jedan od autora metode koja će biti razmotrena u narednom delu teksta je Anil Jain (*Anil K. Jain, 1948*), profesor na državnom Univerzitetu u Mičigenu [56].

Osnovni elementi, koji će biti korišćeni u ovom poglavlju su dati u poglavlju 2.1 ovog rada.

Slike otiska prsta se generalno mogu svrstati u tri kategorije:

Pun ili valjani otisak

Dobija se kada se prst rotira po skeneru tako da se dobiju svi detalji otiska posmatrano sa bilo koje strane prsta. Puni otisci se obično uzimaju pod nadzorom i najčešće su dobrog kvaliteta i sadrže mnoštvo informacija.

Običan otisak

se dobija kada se prst prsloni na skener pri čemu skener vidi centralni deo otiska prsta. Koristi se kod sistema za automatsku identifikaciju, za pristup računarskim resursima, ulasku u restriktivni prostor i slično.

Latentni otisak

predstavljaju slučajno ostavljeni otisak koji najčešće čine fragmenti punog otiska i lošeg kvaliteta. Obično se nalaze na mestima zločina. Zbog obično malog broja informacija veoma je teško i složeno njihovo uparivanje sa bazom podataka otisaka.

Proces identifikacije se uslovno može podeliti u četiri koraka:

Analiza

Ukoliko se radi o latentnom otisku najpre se vrši analiza da li otisak sadrži dovoljno informacija za upoređivanje. Prilikom analize se vrši pokušaj uočavanja i označavanja karakteristika. Analiza se često obavlja ručno pre automatske obrade.

Upoređivanje

Nakon analize se vrši upoređivanje uočenih i označenih karakteristika na posmatranom otisku sa karakteristikama otiska sa referentnim otiskom. Vrš se utvrđivanje njihove sličnosti i različitosti.

Evaluacija

Podrazumeva potvrđivanje identičnosti dva otiska, posmatranog i referentnog ili negaciju identičnosti dva otiska.

Verifikacija

Predstavlja proces preispitivanja otiska prsta od strane drugog ispitivača ili drugim algoritmom.

Iako je učinjen ogroman napredak u povećanju brzine i poboljšanju tačnost automatizovanog prepoznavanja otiska prsta, većina ovih sistema radi dobro samo u slučajevima kada se vrši upoređivanje punih ili običnih otisaka. Najprecizniji komercijalni sistemi postižu stopu prepoznavanja od 99,4% tačnosti prilikom upoređivanja dva otiska prsta. Ukoliko se upoređuju latentni otisci, stopa prepoznavanja je znatno niža. NIST je objavio da sistemi koje koriste postižu stopu prepoznavanja od svega 54% ukoliko se međusobno upoređuju samo latentni otisci. Ukoliko se latentni otisci upoređuju sa punim ili običnim otiscima stopu prepoznavanja se penje do 80%. Ova tačnost je i dalje značajno niža kada se uporedi sa upoređivanjem punih ili običnih otisaka.

Problemi koji se javljaju prilikom upoređivanja latentnih otisaka su:

- nekvalitetni i nejasni otisci
- mala površina latentnih u odnosu na pune otiske
- velika nelinearna izobličenja nastala usled varijacija pritiska

Tipičan pun otisak ima obično oko 60 minucija dok latentni otisak sadrži ne više od 15 upotrebljivih minucija. Postaje jasno da je upoređivanje latentnih otisaka složen postupak.

4.1.2 Izdvajanje minucija

Minucija se može opisati sa četiri atributa:

- x koordinata,
- y koordinata,
- definisanje tangente na papilu u tački minucija ili orijentacija,
- kvalitet utvrđene minucije koji se može označiti sa 0 ukoliko se radi o pozadini. Ukoliko je minucija pouzdana označava se sa 1 i ukoliko je minucija nepouzdana označava se oznakom 2.

Polje se deli na blokove 16 x 16 piksela. Blok ima dobar kvalitet sa jasnom strukturom papila tako što su sve minucije u tom bloku pouzdane. Na nepouzdanom bloku papile su nejasne i može se desiti da neka od minucija nije primećena. Orijentacija se ipak može proceniti i na nepouzdanom bloku na osnovu orijentacije minucija u okolnim blokovima [57].

Validacija minucija

Minucija se smatra lažnom ako je bliska pozadini bloka. Minucija se smatra nepouzdanom ako formira suprotni par sa drugom minucijom. Suprotni par je par minucija koje su blizu jedna drugoj ali imaju suprotne orijentacije. Preostale minucije se smatraju pouzdanim.

Validacija papila

Razmatraju se dva slučaja:

U prvom slučaju svaka papila je podeljena u više segmenata. Papila se identifikuje ukoliko postoji šest susednih piksela u nizu. Pouzdanost papile se zasniva na kranjim tačkama niza od šest piksela, odnosno potrebno je utvrditi da li one imaju susedne tačke koje nastavljaju sledeći niz piksela. Pored ovog slučaja papila se smatra pouzdanom ukoliko se na njenom jednom kraju nalazi minucija.

U drugom slučaju dve papile se smatraju kompatibilnim ako postoji više od 12 piksela na svakoj papili koji su susedi međusobno.

Konstrukcija polja kvaliteta minucija

Nakon ekstrakcije papila, svaki posmatrani blok sadrži ili papilu ili minuciju. Svaki piksel se označi stepenom kvaliteta, kako je to opisano u prethodnom delu teksta. Nepouzdanost papile i minucije se označavaju kao pozadina i to tako što blok čijih je osam susednih blokova nisu svi pouzdani se proglašava pozadinom.

Konstrukcija polja orijentacija papila

Za konstrukcija polja orijentacija papila, otisak prsta je podeljen u blokove veličine 16 x 16 piksela. Procena pravca svake papile se određuje na osnovu tačaka papile koji se javljaju u jednakim intervalima u dužini od 6 piksela [58].

Ako sa *dir* označimo pravac tangente na papilu u *i*-toj tački papile u tački sa koordinatama (*m, n*). Tada se napredovanje papile po *x* i *y* osi označava kao $DX(m, n)$ i $DY(m, n)$ respektivno. Dalje je:

$$DX(m, n) = DX(m, n) + \cos(2 * dir) \quad (126)$$

$$DY(m, n) = DY(m, n) + \sin(2 * dir) \quad (127)$$

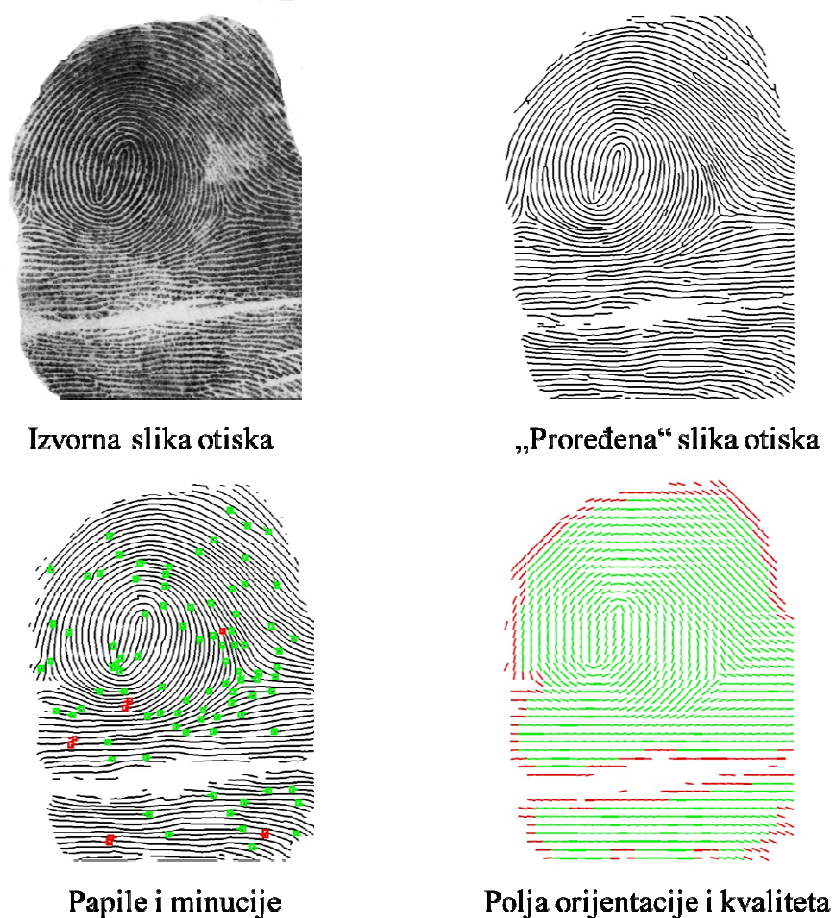
Nakon definisanja papila kao $\arctan\left(\frac{DY}{DX}\right)$ pri čemu se kao rezultat dobija ugao $\theta \in \{-\pi, +\pi\}$, pri čemu je:

$$\cos \theta = \frac{x}{\sqrt{x^2 + y^2}} \quad (128)$$

$$\sin \theta = \frac{y}{\sqrt{x^2 + y^2}} \quad (129)$$

Na taj način smo odredili ugao koji tangenta na papilu u posmatranoj tački zaklapa sa pozitivnim delom *x* ose.

Opisani postupak se nakon prevođenja u softverski kod koji vrši vizuelizaciju može prikazati na kao na slici 39. Zelenom bojom su prikazani pouzdani blokovi dok su crvenom bojom označeni nepouzdati blokovi.



Slika 39. Vizuelizacija procesa identifikacije minucija i procene pravca papila

Na prikazani način se formira šablon ili templejt otiska prsta. Očigledno je da se formiranje šablona vrši korišćenjem softvera i upotrebom računarske opreme. Ovako formirani šablon ćemo kasnije koristiti prilikom predlaganja rešenja zaštite otiska prsta kao i prenosa kriptografskog ključa.

4.1.3 Upoređivanje otisaka

Algoritam za upoređivanje datog, latentnog i punog otiska se sastoji iz tri osnovne faze:

- 1) Lokalno upoređivanje podrazumeva izračunavanje podudarnosti svake od minucija latentnog otiska sa svakom od minucija punog otiska
- 2) Globalno upoređivanje podrazumeva uočavanje pet parova minucija sa najvećom podudarnosti kao početnih parova i dalje traženje podudarnosti takozvanog „greedy“ algoritma. Pomenuti algoritam je algoritam koji koristi takvu metodu za rešavanje problema da u svakom stadijumu bira najbolje lokalno rešenje uz očekivanje da će tako pronaći globalni optimum.

- 3) Izračunavanje rezultata podudaranja se vrši za svaki skup parova minucija koje se podudaraju. Kao konačni rezultat podudaranja se uzima najbolji rezultat.

Lokalno upoređivanje

U ovom koraku, računa se podudarnost između svake od minucija latentnog otisaka prsta i svake od minucija punog otiska prsta se računa. Osim osnovnih osobinama minucija, kao što su: položaj, ugao pravca i tip uvodi se i obeležje kvaliteta minucije i pravca, kako je to opisano u prethodnom delu teksta.

Podudarnosti orijentacije papila za dve minucije koje se posmatraju izračunava se na sledeći način:

$$s_p = e^{\frac{-\varphi}{\pi/16}} \quad (130)$$

Gde je sa φ označen ugao između dve tangente na papile u tačkama minucija koje se upoređuju.

Susedne minucije za posmatranu minuciju su sve minucije u prečniku od 80 piksela. Neka su p i q dve minucije čiju podudarnost izračunavamo. Za svaku susednu minuciju p_i minucije p ispitujemo da li postoji slična susedna minucija za minuciju q koja ima slična svojstva minuciji p_i . Ako takva minucija postoji smatra se da su p i q dve podudarne minucije. Analogni proces se primenjuje i na minuciju q . Podudarnost dve minucije na osnovu analize susednih minucija se izračunava na sledeći način:

$$s_m = \frac{m_p + 1}{m_p + n_p + 3} * \frac{m_q + 1}{m_q + n_q + 3} \quad (131)$$

Gde m_p i m_q označavaju broj susednih minucija koje se podudaraju za minucije p i q respektivno, dok se sa n_p i n_q označavaju broj susednih minucija koje se ne podudaraju za minucije p i q respektivno. Potrebno je primetiti da se m_p može razlikovati od m_q imajući u vidu činjenicu da nije ustanovljeno preslikavanje jedan na jedan između minucija.

Globalno upoređivanje

Preslikavanje između minucija jedan na jedan se ustanovljava prilikom globalnog upoređivanja. Za pronalaženje podudarnih parova minucija koristi se „greedy“ algoritam i to po opadajućem stepenu podudarnosti. Posmatra se ne samo stepen podudarnosti između dve minucije već i stepen različitosti između posmatrane minucije i ostalih minucija. Normalizovana sličnost s_n se definiše na osnovu sličnosti s na sledeći način:

$$s_n = \frac{(M_L + M_R - 1) * s(i, j)}{\sum_{k=1}^{M_R} s(i, k) + \sum_{k=1}^{M_L} s(k, j) - s(i, j)} \quad (132)$$

gde je sa $s(i, j)$ označena sličnost između minucija i i j , dok M_L i M_R označavaju broj minucija u latentnom i punom otisku respektivno.

Izračunavanje stepena podudarnosti

Stepen podudarnosti između latentnog i punog otiska se izračunava na sledeći način:

$$S = W_m + S_m + (1 - W_m) * S_0 \quad (133)$$

gde su sa S_m i S_0 označene podudarnosti na osnovu minucija a zatim na osnovu orijentacije. Faktor W_m je dobijen empirijski i iznosi 0,8.

Dalje se vrši poravnavanje otisaka, ukoliko je prilikom uzimanja otiska došlo do rotacije. Neka su $\Delta x, \Delta y$ i $\Delta \theta$ parametri geometrijske transformacije između latentnog i punog otiska. Ugao rotacije $\Delta \theta_i$ se izračunava za svaki par podudarnih minucija gde je $i = 1, 2, \dots, M$, pri čemu M predstavlja broj uparenih minucija. Tada se srednji ugao rotacije izračunava na sledeći način:

$$\Delta \theta = \arctan(2 * (\sum_{i=1}^M (\sin(\Delta \theta_i)), \sum_{i=1}^M (\cos(\Delta \theta_i)))) \quad (134)$$

Dok se Δx i Δy izračunavaju na sledeći način:

$$\Delta x = \frac{1}{M} \sum_{i=1}^M (x_{R,i} - x_{L,i} \cos(\Delta \theta) - y_{L,i} \sin(\Delta \theta)) \quad (135)$$

$$\Delta y = \frac{1}{M} \sum_{i=1}^M (y_{R,i} - x_{L,i} \sin(\Delta \theta) - y_{L,i} \cos(\Delta \theta)) \quad (136)$$

gde x_L, y_L, x_R i y_R označavaju i -tog para uparenih minucija. Kao i do sada indeks L označava latentni otisak dok indeks R označava puni otisak.

Stopa podudarnosti na osnovu podudarnosti orijentacije S_o predstavlja proizvod kvantitativne S_{on} i kvalitativne S_{oq} stope podudarnosti. Kvantitativna stopa podudarnosti se izračunava kao:

$$S_{on} = \frac{N_b}{N_b + 100} \quad (137)$$

gde je N_b broj blokova u kojima je rezlika između orijentacija u punom i latentnom otisku manja od $\pi/8$. Kvalitativna stopa podudarnost se izračunava na sledeći način:

$$S_{oq} = \frac{1 - 2 * D_o}{\pi} \quad (138)$$

gde je D_o razlika u orijentaciji za sve blokove.

Stepen podudarnosti na osnovu podudarnosti minucija S_m takođe predstavlja proizvod

kvantitativne stope S_{mn} i kvalitativne stope S_{mq} . Kvantitativna stopa se izračunava kao:

$$S_{mn} = \frac{M}{M + 8} \quad (139)$$

dok se kvalitativna stopa izračunava na sledeći način:

$$S_{mq} = S_d * \frac{M}{M + N_L} * \frac{M}{M + N_R} \quad (140)$$

gde je S_d prosečna podudarnost svih uparenih minucija a N_L i N_R su brojevi neuparenih minucija.

Opisanim matematičkim aparatom moguće je izvršiti upoređivanje dva otiska prsta i utvrditi stepen njihove podudarnosti.

Jednu od opisanih modifikacija opisanog matematičkog modela je iskoristila američka kompanija „Digital Persona“ za razvoje veoma efikasnog softvera za kreiranje šablona i upoređivanje dva otiska prsta. Softver će u kasnijem delu teksta biti korišćen u tu svrhu radi eksperimentalnog ispitivanja predloženog rešenja. Pored toga postoje i mnogi drugi proizvođači kao što je proizvod kompanije „Aware“, koja snabdeva biometrijskim rešenjima FBI, kao i mnogi drugi proizvođači. Treba imati na umu činjenicu da su svi ovi alati zatvoreni i čuvani kao stroga poslovna tajna.

4.2 Deloneova triangulacija

Upoređivanje dva otiska se može vršiti isključivo upoređivanjem minucija, što predstavlja najrasprostranjeniji način upoređivanja. Za svaku od minucija se posmatraju tri karakteristike: tip, koordinate i orijentacija [59].

Očigledno je da je princip istovetan principu iznetom u prehodnom poglavlju. Razlika nastaje u načinu formiranja obrasca što će kasnije biti iskorišćeno za izlaganje rešenja koje se predlaže u ovom radu.

Osnovna ideja korišćenja Deloneove triangulacije (*Борис Николаевич Делоне, 1890. - 1980.*) je formiranje mreže na osnovu dostupnih tačaka, u našem slučaju dostupnih minucija. Upoređivanje dva otiska prsta se nakon toga svodi na upoređivanje dve mreže.

Kreiranje Deloneove mreže otiska

Uočene minucije na otisku prsta se mogu posmatrati kao rasute tačke u ravni. Triangulacija predstavlja efikasno sredstvo u konstruisanju konzistentnog skupa tačaka kojima se može manipulirati.

Za kreiranje Deloneove mreže se koristi algoritam koji se sastoji iz sledećih koraka:

Prethodna obrada podataka

Kreira se jedinstvena mreža koja pokriva ceo otisak, odnosno dostupan segment otiska. U svakom od kvadrata mreže može se naći jedna, ni jedna ili više minucija.

Određivanje početne minucije

Početna minucija mora da se nalazi u centru mreže, ili se uzima minucija koja je najbliža centru mreže. Nakon toga se traži minucija koja je najbliža početnoj minuciji. Kada se najbliža minucija odredi, dve minucije se spajaju jednom duži i na taj način se formira prva ivica.

Formiranje trouglova

Pretraga počinje od početne ivice u matematički negativnom smeru odnosno u smeru kazaljke na satu. Traži se tačka koja će spajanjem sa drugom tačkom obrazovati najveći ugao. Nakon spajanja treće tačke sa drugom, kao i treće tačke sa prvom formira se prvi trougao. Zatim se određuje kružnica koju definišu tri tačke dobijene na opisan način. Sledeći korak je izbor minucija u okviru nacrtanog kruga koja spojena sa nekom od tačaka definiše najveći ugao. Spajanjem tih tačaka se dobija sledeći trougao. Postupak se nastavlja sve dok ne ostanu ni jedan red ili kolona koji nisu obuhvaćeni.

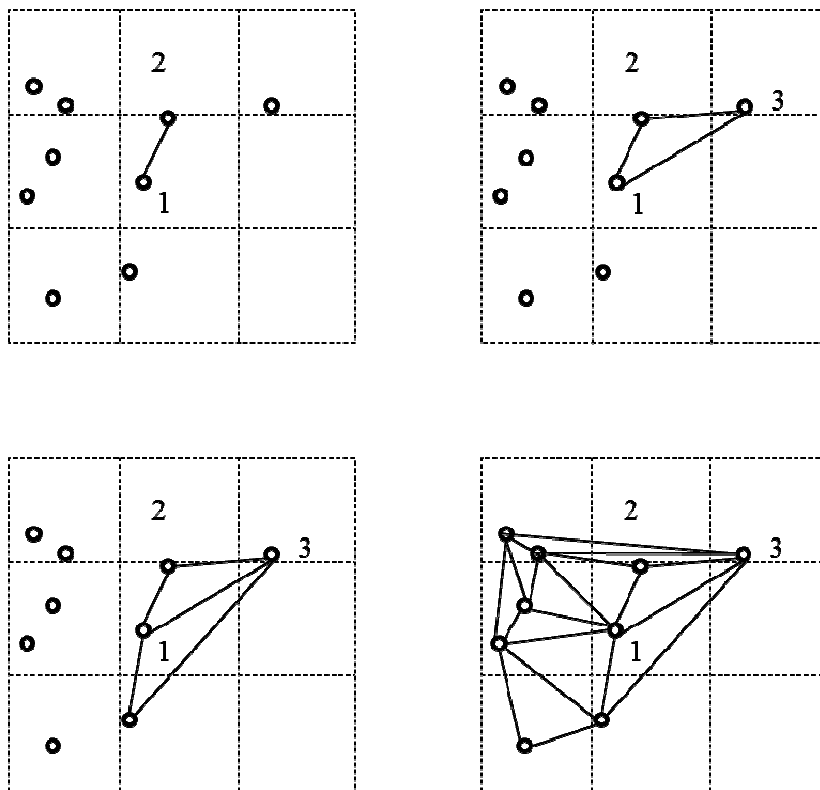
Sastavljanje trouglova

Dobijeni trouglovi se sastavljaju u jedinstvenu mrežu tako da više ne postoje propusti u spajanju minucija.

Dobijena mreža se naziva Deloneovom mrežom.

Prevođenjem Deloneove mreže u digitalni oblik kreira se drugi šablon koji će biti iskorišćen u predloženom rešenju.

Opisani koraci su prikazani na slici 40.



Slika 40. Kreiranje Deloneove mreže

Primena Deloneove mreže je prikazana na slici 41.



Slika 41. Deloneova mreža na otisku prsta

Postoje definisani načini za upoređivanje dva otiska prsta upoređivanjem odgovarajućih Deloneovih mreža.

5. Predloženo rešenje

U ovom delu rada biće razmotreni različiti aspekti predloženog rešenja zaštite biometrijskih podataka i kombinovanja otiska prsta sa kriptografskim ključem. Rešenje treba da doprinese bezbednijem čuvanju i prenošenju otiska prsta. Pored toga rešenje treba da doprinese pouzdanom prenosu kriptografskog ključa zajedno sa podacima o otisku prsta i na taj način omogući nedvosmisleni identifikaciju osobe. Detaljno će biti predstavljeni izabrani mehanizmi i matematički modeli. Biće objašnjeno kompletno rešenje, kako po segmentima tako i u celini. Do rešenja se došlo analizom dosadašnjih rešenja od kojih su neka u primeni dok su neka od njih još u fazi teorijskih razmatranja. U prethodnom delu rada date su osnovne stavke koje se koriste kao elementi za konstrukciju rešenja. Opisani matematički modeli koji su izabrani i opisani, koristiće se u predloženom rešenju.

5.1 Izbor algoritama za kreiranje šablona otiska prsta

Osnovna ideja rešenja koje je predstavljeno u radu je korišćenje dva nezavisna šablona otiska prsta. Šablon ili templejt otiska prsta predstavlja digitalizovanu sliku otiska prsta. Šablon nije jednostavno fotografija otiska prsta već predstavlja obrađen otisak prsta u smislu kako je to opisano u poglavljima 4.1.2 i 4.2. Model upoređivanja latentih otisaka u poglavlju 4.1.2 detaljno objašnjava kako se iz slike samog otiska izdvajaju karakteristične tačke otiska ili minucije. Takođe je opisano kako se određuju orijentacije papila u tačkama minucija. Ovakav pristup je jedan od uobičajenih i koriste ga mnogi istraživači. Navedeni model se koristi i u komercijalne svrhe. Jedna od kompanija za razvoj biometrijskog softvera koja koristi navedeni algoritam je kompanija „Digital persona“, osnovana 1996. godine u Kaliforniji, USA. Činjenica je da kompanija koristi modifikovan algoritam koji je opisan u poglavlju 4.1.2 i da pojedini detalji predstavljaju poslovnu tajnu. Ipak prikazani algoritam i metodologiju za kreiranje šablona čini osnovu sistema koji je kompanija razvila.

Kao algoritam za kreiranje drugog šablona izabrana je metoda Deloneove triangulacije što je detaljno opisano u poglavlju 4.2. Formiranje Deloneove mreže predstavlja razvijen postupak koji se može koristiti prilikom rešavanja različitih problema. Deloneova triangulacija predstavlja dobar model za obradu otiska prsta.

5.2 Izbor metode kodiranja

Postoji više različitih metoda za detekciju i korekciju greške, kako je to opisano u poglavlju 3. Tokom rada na ovom problemu testirano je više različitih kodova za korekciju greške. Između ostalog testirani su Rid-Solomonovi i Adamarovi kodovi, koji su opisani u poglavlju 3.1, i to u vidu uzastopnog kodiranja a zatim dekodiranja u obrnutom redosledu. Na kraju je odlučeno da se kao metoda za korekciju greške koriste Slepian-Vulfovi kodovi, opisani u poglavlju 3.4. Slepian-Vulfovi kodovi tretiraju dva ulazna parametra koji su u međusobnoj korelaciji. Kao izlazni rezultat Slepian-Vulfovog kodiranja mogu se dobiti jedan ili dva parametra, već u zavisnosti od izbora verzije algoritma. Kodovi su se pokazali kao primenljivi u rešenju koj je predloženo.

5.3 Generisanje kriptografskog ključa

Osnovni elementi kriptografije su dati u poglavlju 3. dok se više detalja može naći u navedenoj literaturi kao i drugoj literaturi koja je široko dostupna.

Kriptografski ključ predstavlja slučajni binarni niz sa uniformnom ili bar približno uniformnom raspodelom verovatnoća pojavljivanja svakog od elemenata. Pošto se radi o binarnom nizu potrebno je da je verovatnoća pojavljivanja nula i jedinica u binarnom nizu približno jednaka ili barem približno jednaka, odnosno:

$$p(0) \approx p(1) \approx 0,5 \quad (141)$$

Generator ključeva odnosno binarnog slučajnog niza može biti hardverski uređaj ili softver koji generiše slučajni bit. Ukoliko se radi o softverskom rešenju, gotovo je sigurno da se će se kao izlaz dobiti pseudoslučajni niz, koji je detaljnije opisan u Poglavlju 2.1.2. Za potrebe kriptografije se koriste HSM (*Hardware Security Module*) uređaji koji generišu slučajne binarne nizove.

Potrebno je naglasiti da se iz otiska prsta ili bilo koje druge biometrijske veličine, ne može direktno kreirati kriptografski ključ iz više razloga. Jedan od važnijih razloga za to je činjenica da otisak prsta nema ni približno ravnomernu raspodelu kada se prevede u binarni niz. Analogna tvrdnja važi i za druge biometrijske veličine.

Iz navedenih razloga u predloženom rešenju je izabrano da se kriptografski ključ generiše nezavisno od otiska prsta koji se obrađuje, odnosno uz poštovanje svih pravila koja važe za kriptografske ključeve.

5.4 Kombinovanje templejta otiska prsta sa kriptografskim ključem

Postojala je dilema kako od dva binarna niza od kojih jedan predstavlja šablon otiska prsta a drugi kriptografski ključ kreirati jedan binarni niz. Dodatni zahtev je bio da novi binarni niz u sebi sačuva informacije kako o šablonu otiska tako i o kriptografskom ključu, odnosno da se iz njega mogu rekonstruisati bilo kriptografski ključ bilo šablon otiska. Imajući u vidu

činjenice iznete u poglavlju 2.2.5, da ako možemo razbiti sistem za XOR možemo i za svaku drugu funkciju i obrnuto, tako da se ništa ne dobija upotrebom složenije funkcije. Naprotiv, ako je funkcija složenija produžava se vreme šifrovanja.

Iz navednog zaključujemo da je ekskluzivno OR ili XOR sasvim zadovoljavajuće ne samo zbog potrebe razmatranja u ovom radu već i šire.

5.5 Upoređivanje dva otiska

Upoređivanje dva otiska prsta se može izvršiti na više načina. Način koji se bira mora biti kompatibilan sa načinom kreiranja šablona. Ukoliko želimo da uporedimo dva otiska prsta neophodno je da sliku svakog otiska prsta obradimo istim algoritmom, recimo za izdvajanje minucija kako bi šabloni bili uporedivi.

Pošto proučavanje samog upoređivanja otisaka nije ključno u ovom radu, već se koristi kao pomoćni alat, iskorišćen je algoritam za upoređivanje koji je opisan u poglavlju 4.1.3 a koji koristi već pomenuta kompanija „Digital Persona“. Kako je navedeno način kreiranja šablona i način upoređivanja otisaka su u direktnoj vezi. Korišćenjem kompatibilnog rešenja omogućeno je da se vrši upoređivanje otisaka komercijalnim rešenjem. Istini za volju, komercijalno rešenje nije do kraja otvoreno, ali su poznate njegove osnovne pretpostavke.

Za potrebe razmatranje u ovom radu, izabrani način upoređivanja otisaka se pokazao kao zadovoljavajući.

5.6 Koncept dva templejta

Osnovna ideja rešenja je da se šablon otiska nikada ne prenosi u otvorenom obliku. Ukoliko bi došlo do kompromitovanja šablona *de facto* bi došlo i do kompromitovanja samog otiska prsta. Takav događaj bi predstavljao nenadoknativ gubitak.

Da bi se to sprečilo potrebno je da se šablon otiska uvek prenosi u zaštićenom obliku. Osnovna ideja zaštićenog oblika je postupak u kome se najpre generiše kriptografski ključ, nezavisno od otiska prsta. Zatim se otisak prsta transformiše u odgovarajući šablon. Sada imamo dva binarna niza. Odgovarajućim izborom parametara dobijaju se dva binarna niza jednake dužine. Sada se izvrši logičko sabiranje ova dva niza, XOR. Dobijeni niz se može „raspakovati“ jedino ukoliko je prisutan originalni otisak prsta iz koga se ponovo istim algoritmom formira isti šablon. Imajući u vidu svojstva logičkog sabiranja data u Tabeli 3, očigledno je da će se logičkim sabiranjem novonastalog niza i šablona otiska prsta ponovo dobiti identičan kriptografski ključ. Ukoliko dođe do kompromitovanja zaštićenog niza, jednostavno se generiše novi kriptografski ključ pomoću kojeg se dobija novi kombinovani niz uz korišćenje istog otiska prsta. Sam otisak ostaje zaštićen iako je kombinovani niz kompromitovan.

U prethodnom pasusu problem je donekle pojednostavljen pa se rešenje čini relativno jednostavnim.

Problem nastaje prilikom uzimanja otiska prsta. Gotovo nikada se ne može uzeti otisak prsta dva puta na isti način i to iz više razloga:

- prst se ne postavlja na čitač uvek na isti način,
- u zavisnosti od pritiska na skener dolazi do manje ili veće distorzije oblika samog otiska,
- ukoliko je prst manje ili više vlažan može doći do nejasnoća oblika papila i minucija,
- ...

Iz pomenutih razloga potrebno je koristiti matematičke alate za detekciju i korekciju greške. U suštini ovo alati dodaju bitove na određenim mestima i na određeni način koji naknadno omogućavaju da se binarni niz reprodukuje u svom originalnom obliku. Koji bitovi i na kojim mestima se dodaju zavisi od izabranog koda za korekciju greške. Što je kod za detekciju i korekciju greške bolji, greške su manje. Treba imati u vidu da bi bilo idealno da se greške svedu na nulu, ali to, za sada, ni jedan kod nije uspeo da postigne.

Sledeći momenat koji je potrebno imati na umu prilikom rešavanja problema je dozvoljeno odstupanje šablona prilikom različitih očitavanja otiska prsta. Izbor oblasti tolerancije ε je od velike važnosti. Ukoliko su tolerancije isuviše uske dva otiska se nikada neće proglasiti identičnim. U obrnutom slučaju ukoliko su tolerancije isuviše široke, za podudarne će se proglašavati i otisci koji nisu identični. Problem definisanja oblasti tolerancija je objašnjen u poglavlju 2.1.3. Prava mera tolerancija se određuje iskustveno, kroz eksperimente.

Naročito je bitno imati u vidu osobine kriptografskog ključa. Ukoliko se u kriptografskom ključu, zapravo binarnom nizu izmeni makar jedan bit, kriptografski ključ postaje potpuno neupotrebljiv iako su svi ostali bitovi ostali nepromenjeni. Ukoliko se sa, ma i najmanje izmenjenim kriptografskim ključem izvrše kriptografske operacije, rezultati će biti potpuno neupotrebljivi ili kriptografske operacije uopšte neće moći da budu izvršene.

Ideja predloženog rešenja u ovom radu se u osnovi sastoji u uvođenju dva različita šablona za isti otisak prsta. Kada se pominju različiti šabloni, misli se na dva različita algoritma za kreiranje šablona. Prvi šablon se koristi isključivo za identifikaciju. Ako se prilikom identifikacije, korišćenjem prvog šablona, pokaže da se radi o identičnim otiscima, pristupa se obradi drugog šablona. Drugi šablon se nalazi u zaštićenom obliku i predstavlja binarni niz XOR-ovan sa kriptografskim ključem. Sada se od prisutnog otiska prsta kreira drugi šablon koji se XOR-uje sa kombinovanim nizom. Kao rezultat se dobija kriptografski ključ.

Uvođenjem Slepian-Vulfovog kodiranja ceo postupak postaje nešto složeniji ali zato pouzdaniji. Pošlo se od pretpostvke da ograničenje računarskih resursa ne predstavlja problem, pri čemu je povećanje tačnosti veoma bitno.

5.7 Definisanje predloženog koncepta

Predloženi koncept se može definisati opisom pojedinačnih procesa kao i redosledom izvršavanja određenih koraka. Odvojeno će se razmatrati proces kodiranja i proces dekodiranja.

5.7.1 Proces kodiranja

1. Obezbeđivanje originalnog otiska prsta

Pre početka samog procesa neophodno je prisustvo originalnog otiska. Originalni otisak može biti prisutan i vidu osobe koja će skenirati svoj otisak prsta na skeneru ili već skenirana slika otiska koja se čuva u bazi podataka. Otisak sa kojim se započinje proces se smatra originalnim. Svi ostali otisci istog prsta koji u sebi sadrže distorzije mogu biti pozitivno ili negativno identifikovani, u zavisnosti od stepena distorzije.

2. Generisanje kriptografskog ključa - K

Predstavlja proces generisanja slučajnog binarnog niza određene dužine. Dužina kriptografskog ključa varira u zavisnosti od algoritma koji se koristi, ali je uvek eksponent broja dva koji se izračunava 2^n . Za simetrične šifarske sisteme su neke od uobičajenih dužina ključeva od 64 bita do 256 bita. Za asimetrične šifarske sisteme generiše se par ključeva, javni i tajni. Javni ključ nije potrebno štititi jer je po definiciji „javni“. Ključ koji je potrebno štititi je tajni ključ i njegova dužina se za RSA algoritam kreće u granicama od 512 bita do 4096 bita. Javni ključ je znatno duži.

3. Kreiranje šablona otiska prsta 1 - S_1

Šablon otiska prsta 1 - S_1 se kreira po metodi opisanoj u poglavlju 4.1.2, algoritam A_1 . Kao što je napomenuto, S_1 će se koristiti isključivo za identifikaciju i nema dodirnih tačaka sa kriptografskim ključem niti kodovima za korekciju greške. Pre smeštanja templateja S_1 na token vrši se njegovo šifrovanje privatnim ključem korišćenjem PKI sistema, kako je to opisano u poglavlju 2.2.4.

4. Kreiranje šablona otiska prsta 2 - S_2

Šablon otiska prsta 2 - S_2 se kreira po metodi opisanoj u poglavlju 4.2, algoritam A_2 , koristeći Deloneovu mrežu. S_2 se neće koristiti za identifikaciju već kao nosilac kriptografskog ključa. Očigledno je da su S_1 i S_2 u korelaciji imajući u vidu činjanicu da potiču od istog otiska prsta.

5. Kreiranje niza N

Niz N se kreira tako što se izvrši logičko sabiranje, XOR-ovanje S_2 i kriptografskog ključa - K . Novonastali niz N sadrži u sebi podatke kako o kriptografskom ključu tako i o šablonu 2 otiska prsta. Očigledno je da je niz N u korelaciji sa kriptografskim ključem jer u sebi sadrži i podatke o kriptografskom ključu. Razlog izbora logičkog sabiranja, XOR, prezentovan je u prethodnom tekstu ovog rada.

6. Ulazak ključa u enkoder 1- E_1 i izlaz niza R_1

Imajući u vidu šemu Slepian-Vulfovog kodiranja, koja je data na slici 36., očigledno je da sistem funkcioniše tako što postoje dva enkodera i jedan dekođer. U ovom koraku, kriptografski ključ se prosleđuje u enkoder 1- E_1 . Kao izlaz iz E_1 proizilazi binarni niz R_1 .

7. Ulazak niza N u enkoder 2 - E_2 i izlaz niza R_2

Takođe imajući u vidu šemu Slepian-Vulfovog kodiranja, koja je data na slici 36., u enkoder 2 - E_2 šalje se niz N . Kao izlaz iz E_2 proizilazi binarni niz R_2 .

Ovim je proces kodiranja završen. Da se podsetimo dobijene su tri veličine odnosno tri binarna niza koje je potrebno preneti na mesto dekodiranja, a to su: S_1 , R_1 i R_2 .

Ukoliko bi se prenos vršio nesigurnim komunikacionim kanalima ili komunikacionim kanalima sa šumom, otvorili bi se novi problemi koji bi bili rešavani posebnim metodama.

Iz tih razloga pretpostavićemo da se sve tri izlazne komponente dobijene u procesu kodiranja smeštaju na karticu sa mikrokontrolerom ili USB token. Nije preporučljivo koristiti standardne USB memorije jer kartica sa mikrokontrolerom ili USB tokeni imaju svoj sistem zaštite koji obezbeđuje dodatnu sigurnost u toku prenosa. Naime, za pristup podacima na čipu ili tokenu potrebno je znati odgovarajući PIN.

Pretpostavićemo da je vlasnik otiska ujedno i vlasnik koji poseduje karticu sa mikrokontrolerom ili USB token. Takođe ćemo pretpostaviti da vlasnik nosi karticu ili token sa sobom. Na mestu dekodiranja pojaviće se vlasnik sa karticom ili tokenom i svakako sa svojim otiskom prsta.

5.7.2 Proces dekodiranja

1. Obezbeđivanje originalnog otiska prsta

Da bi se izvršio proces dekodiranja neophodno je prisustvo originalnog otiska prsta, kako je to opisano u prethodnom poglavlju.

2. Kreiranje šablona otiska prsta - S_1'

Na osnovu prisutnog originalnog otiska prsta kreira se šablon otiska prsta S_1' kojise kreira po metodi opisanoj u poglavlju 4.1.2, algoritam A_1 . Iz ranije opisanih razloga, pretpostavka je da će se S_1' u određenoj meri razlikovati od S_1 .

3. Čitanje šablona otiska prsta S_1 sa nosioca podataka

Vrši se čitanje šablona S_1 sa nosioca podataka, prema proceduri koja važi za čip odnosno token. Pre dobijanja templejta S_1 u otvorenom obliku vrši se njegovo dešifrovanje javnim ključem PKI sistema, kako je to opisano u poglavlju 2.2.4. Sada je S_1 raspoloživ u sistemu za dekodiranje.

4. Upoređivanje S_1 sa S_1'

Vrši se upoređivanje S_1 koji je dobijen od prisutnog otiska sa S_1' koji je pročitani sa nosioca podataka. Nakon upoređivanja se potvrđuje istovetnost S_1 i S_1' , odnosno potvrđuje se identitet osobe.

Ukoliko se potvrdi podudarnost, proces dekodiranja se nastavlja.

Ukoliko se negira podudarnost, proces dekodiranja se prekida i proglašava se neuspešnim.

5. Kreiranje šablona S_2'

Ukoliko se proces nastavlja, u daljem postupku se iz prisutnog otiska prsta kreira šablon S_2' po metodi opisanoj u poglavlju 4.2, koristeći Deloneovu mrežu, algoritam A_2 . Iz ranije opisanih razloga, pretpostavka je da će se S_2' u određenoj meri razlikovati od S_2 .

6. Čitanje nizova R_1 i R_2 sa nosioca podataka

Sada se čitaju nizovi R_1 i R_2 sa čipa ili tokena prema proceduri koja važi za čip odnosno token. Sada je šablon S_2 raspoloživ u sistemu za dekodiranje.

7. Prosleđivanje nizova R_1 i R_2 Slepian-Vulfovom dekoderu - D

Nizovi R_1 i R_2 koji su pročitani sa nosioca podataka se prosleđuju Slepian-Vulfovom dekoderu - D , prema šemi datoj na slici 36.

8. Dekodiranje u Slepian-Vulfovom dekoderu - D

Nizovi R_1 i R_2 koji su pročitani sa nosioca podataka se prosleđuju Slepian-Vulfovom dekoderu - D, prema šemi datoj na slici 36. Kao izlaz iz D se dobijaju dve veličine: kriptografski ključ K' i niz N' .

9. Logičko sabiranje niza N' i šablona $2'$

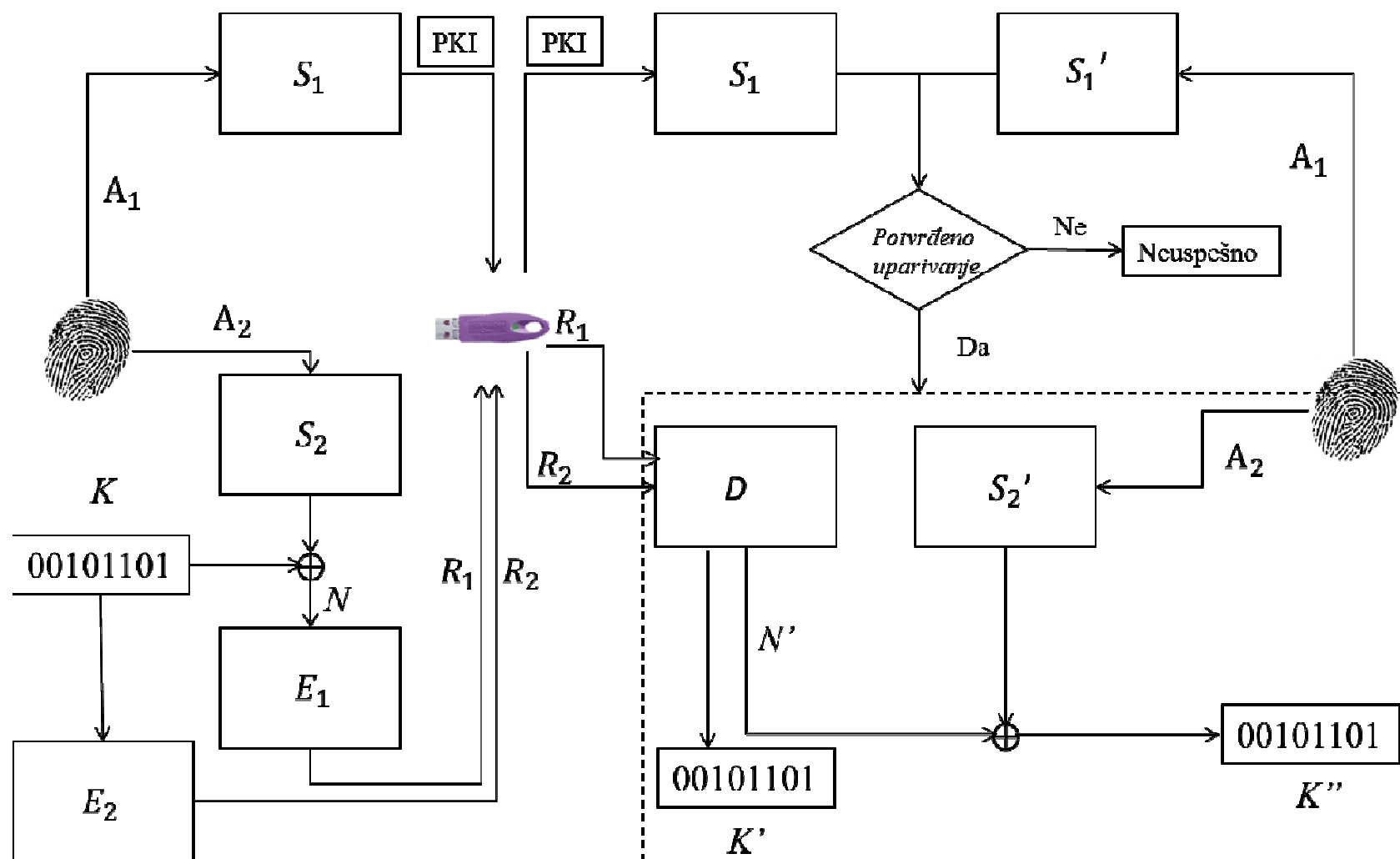
Vrši se logičko sabiranje, XOR-ovanje niza N' i šablona S_2' . Kao rezultat se dobija kriptografski ključ K'' .

10. Upoređivanje ključeva

U ovom koraku se vrši upoređivanje ključeva K' i K'' . Ukoliko su ključevi isti proces je uspešno okončan.

Neophodno je napomenuti da se konačna potvrda ispravnosti ključa vrši u procesu dešifrovanja šifrata koji je šifrovan originalnim ključem. Ukoliko se dešifrovanje izvrši korektno proces je uspešan. Ukoliko se dešifrovanje ne izvrši korektno proces nije uspešan.

Šematski prikaz procesa predloženog rešenja je dat na slici 42.



Slika 42. Šematski prikaz predloženog rešenja

6. Analiza eksperimentalnih rezultata

U cilju ispitivanja karakteristika predloženog rešenja u smislu njegove potvrde opravdanosti, ostvarivosti i pouzdanosti, izvršeno je više eksperimenata. Eksperimenti su imali za cilj da prikažu ponašanje sistema u radu sa realnim podacima. Osnovni cilj izvođenja eksperimenata je eksperimentalno pokazivanje tačnosti koju je moguće dostići predloženim modelom, odnosno ustanovljavanje efikasnosti sistema.

Pre nego što se sagledaju i analiziraju rezultati potrebno je imati na umu da su postojala određena ograničenja koja bi mogla uticati sam proces testiranja sistema.

Ograničenja su se ogledala kako u kvalitetu korišćene opreme, skenera otiska prsta, tako i u dostupnosti kvalitetne baze podataka. Dostupnost originalnog komercijalnog softvera za pomoćne operacije u otvorenom obliku takođe je predstavljala ograničavajući faktor.

6.1 Korišćeni resursi

Prilikom vršenja eksperimenata korišćeni su različiti resursi: baza podataka otiska prsta, hardver, softver.

6.1.1 Korišćena baza podataka

U eksperimentima je korišćena biometrijska baza podataka Centra za istraživanje bezbednosti biometrijskih podataka, Nacionalne laboratorija za prepoznavanje obrazaca, Instituta za automatizaciju, Kineske akademije nauka (CASIA).

Korišćena je baza podataka slika otiska prsta, verzija 5.0, CASIA - FingerprintV5. Baza podataka sadrži 20.000 slika otiska prsta od 500 prikupljenih od osoba. Slike otiska su dobijene korišćenjem skenera otiska prsta URU4000.

Otiske prstiju su dobrovoljno davali: diplomirani studenti, radnici, konobari ... itd. Svaka osoba je dala 40 otiska prsta svojih osam prstiju i to levi i desni: palac, srednji prst, domali i mali prst, odnosno 5 otisaka po prstu. Kao što se može videti otisak kažiprsta nije uziman iz razloga zaštite privatnosti i zaštite identiteta.

Prilikom uzimanja otisaka vršene su rotacije prstiju dok je pritisak na skener bio različitog intenziteta, kako bi se dobile različite varijacije istog otiska. Očigledno je da se ovde ne radi o punim otiscima već o običnim i latentnim otiscima.

Sve slike otiska prsta su sive dok je za opisivanje jednog piksela korišćeno 8 bita. Slike su bit mapirani fajlovi, rezolucije 328 x 356 piksela.

Slike su označene na sledeći način: ...\\YYY\\H\\YYY_HX_KKK.bmp, gde je:

YYY jedinstveni identifikator osobe

H je oznaka ruke sa „L“ je označena leva a sa „R“ desna ruka

HX označava koji prst je prikazan i to: 0 označava palac, 1 označava srednji prst, 2 označava domali prst i 3 označava mali prst

KKK predstavlja redni broj otiska posmatranog prsta

6.1.2 Korišćeni hardver

Korišćen je laptop računar Toshiba Satellite, M60-161, sa Intel Centrino procesorom, na taktu od 1,86 GHz sa 2 GB RAM memorije, i hard diskom kapaciteta 55,6 GB.

Na računaru je korišćen operativni sistem Windows XP Professional sa SP3.

Pored standardne računarske opreme korišćen je specijalizovani USB skener za skeniranje otiska prsta proizvođača firme „Digital Perosna“. Korišćen je model *U.aRE.U 4000 Optical Fingerprint Scanner*, slika 1. Osnovne karakteristike skenera su sledeće:

- Rezolucija skeniranja: 512 Dpi
- Tip senzora: optički
- Skenirani podaci: 8-Bit Grayscale, što znači 256 nijansi sive boje
- Oblast koja se skenira: elipsa sa osama 14,6 mm i 18,1 mm
- Kompatibilnost: USB 1.0, 1.1 and 2.0 (Full Speed)
- Standardi koje zadovoljava su: FCC Class B, CE, Ices, Bsmi, Mic, USB, Whql

6.1.3 Korišćeni softver

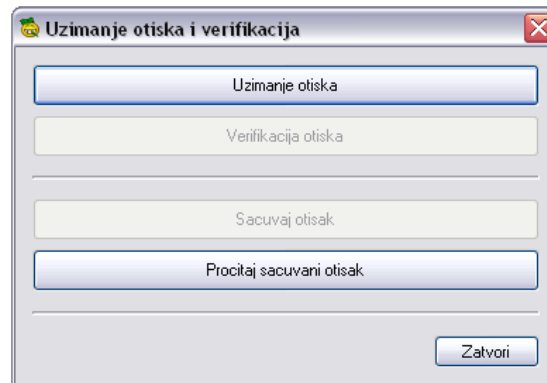
Korišćeni su SDK (*Software Development Kit*) alati firme „Digital Persona“ koji omogućava komunikaciju i razvoj aplikacija sa optičkim skenerom istog proizvođača. SDK alati sadrže API funkcije za pristup samom čitaču kao i okruženje za razvoj sopstvenih algoritama za kreiranje šablona i upoređivanje otisaka. Starije verzije su podržavale samo skeniranje otiska na skeneru dok novije verzije SDK podržavaju i učitavanje slika sa diska.

API (*Application programming interface*) funkcije podržavaju različite operativne sisteme: Windows, Linux, Windows CE, MacOS kao i Android. Pored toga, API funkcije su podržane u više programskih jezika i to: .NET, Java, C#, ActiveX/COM, C/C++, JPOS/OPOS.

Za razvoj koda vezanog za ovaj rad korišćen je Windows operativni sistem kao i API funkcije u C/C++ kao i C# programskim jezicima. Pored toga razvijan je softver koji nije u direktnoj vezi sa API funkcijama, Pri čemu su korišćena razvojna okruženja Microsoft Visual Studio 6 za C i C++ programske jezike kao i Microsoft Visual Studio 2010 u C# programski jezik za

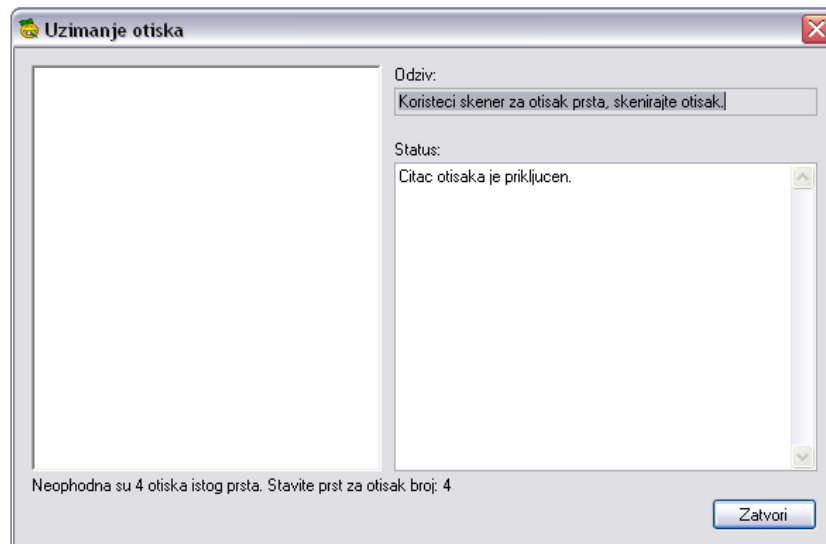
razvoj UI (User interface), korisničkog interfejsa. C i C++ su korišćeni za razvoj funkcija koje direktno pristupaju memorijskim lokacijama hardvera. Delovi koda koji su razvijeni u C i C++ programskim jezicima su u najvećem broju slučajeva kompajlirani u vidu dll (*dynamic link library*) fajlova. Dll fajlovi se pozivaju iz C# koda korišćenjem opcije „unmanaged code“.

Osnovni UI dela softverskog rešenja započinje kontrolnim panelom, kao što je to prikazano na slici 43.



Slika 43. Kontrolni panel osnovnog softverskog rešenja

U početnom stanju aktivne su opcije za uzimanje otiska ili čitanje otiska koji je već sačuvan na disku. Kao prva operacija, predviđeno je uzimanje otiska sa čitača, kao što je to prikazano na slici 44.



Slika 44. Uzimanje otiska sa čitača

Softver daje predefinisane poruke u zavisnosti od stanja sistema, kao o kratku instrukciju za korisnika. Predviđeno je uzimanje četiri otiska radi formiranja templejta. Pojednostavljeno

rečeno, kreira se oblast tolerancije u obliku četvorougla za svaku minuciju. Softver automatski detektuje da li je jagodica prsta prislonjena na skener i automatski vrši skeniranje. Nakon skeniranja generiše se poruka da je neophodno skloniti prst sa čitača i ponovo ga prisloniti, kao što je to prikazano na slici 45.



Slika 45. Skeniranje otiska prsta

Nakon uzimanja sva četiri predviđena otiska kreira se templejt o čemu se pojavljuje poruka za korisnika, kako je to prikazano na slici 46.



Slika 46. Faza završetka kreiranja templejta

Nakon kreiranja templejta, sam templejt se može smestiti na disk ili se direktno upoređivati sa drugim otiskom. Upoređivanje se može vršiti sa direktno prisutnim otiskom, ili otiskom koji je već mešten na disku računara. Analogno se mogu čitati i otisci koji se nalaze u bazi podataka ili se nalaze smešteni na nekom drugom medijumu kao što su, čip kartica, kartica sa laserskom memorijom ili USB token.

Sada su dostupne sve opcije na kontrolnom panelu i možemo vršiti upoređivanje otisaka. Pozitivna verifikacija je prikazana na slici 47.



Slika 47. Pozitivna verifikacija otiska

Negativna verifikacija je prikazana na slici 48.



Slika 48. Negativna verifikacija otiska

Na prethodnim slikama su prikazane elementarne funkcionalnosti razvijenog softvera.

Korišćenjem opisanih alata i programskih jezika su na optimalan način iskorišćene sve mogućnosti po pitanju razvoja softvera koje su u ovom trenutku na raspolaganju.

6.2 Analiza rezultata

Pre nego što se pređe na sagledavanje i komentarisanje rezultata, potrebno je napomenuti da algoritam za kreiranje šablona koji je korišćen kreira binarni niz uvek jednake dužine a to je 2048 bita. Algoritam je programiran tako da sva polja na kojima nije prepoznata minucija popunjavama nulama.

Eksperiment je izvođen tako što je uvek prvi otisak u nizu uziman kao referentni otisak dok su preostala četiri otiska tretirani kao testni. Drugim rečima, u fazi kodiranja se uvek uzimao prvi otisak u nizu, u CASIA bazi je to otisak sa indeksom nula.

U fazi dekodiranja je najpre uziman isti otisak sa indeksom nula da bi se proverilo da li će sistem prepoznati apsolutno isti otisak koji se uzeo u obzir u fazi kodiranja?

Može se konstatovati da je tačnost reprodukovanja kriptografskog ključa bila relativno visoka. Greške su se uglavnom dešavale kada softver, iz nekih razloga nije učitao kompletnu sliku, što se može pripisati problemima u relizaciji samog softvera.

U sledećem testiranju u fazi dekodiranja, uzimani su redom otisci sa indeksima 1, 2, 3 i 4. Prilikom ovog testiranja su dobijani primetno lošiji rezultati.

Uticaj kvaliteta podataka i primene prethodnog upoređivanja

U samoj bazi podataka se može uočiti da stepen efikasne realizacije sistema značajno opada sa padom kvaliteta slike otiska. Pod kvalitetom se podrazumeva ne kvalitet same slike otiska, pošto su sve slike otisaka u bazi relativno ujednačenog kvaliteta. Misli se na broj papila i minucija koje se mogu uočiti, odnosno na odgovarajuću veličinu regiona prsta koji je skeniran.

Isti eksperiment je kasnije ponavljan bez opcije prethodne verifikacije identiteta odnosno bez korišćenja opcije upoređivanja šablona S_1 . Tačnost rekonstrukcije kriptografskog ključa je značajno pala, opet u zavisnosti od kvaliteta slike, s tim što je pad znatno izraženiji nego što je to slučaj kada se koristi kompletno rešenje sa prethodnom proverom identiteta korišćenjem šablona S_1 .

Napravljen je deo softvera koji meri aktivnu skeniranu površinu, odnosno površinu na kojoj su vidljive papile i minucije. Nakon što izmeri površinu sa podacima izražava je u procentima u odnosu na ukupno skeniranu površinu. Ukoliko je skenirana površina u potpunosti popunjena kvalitet se ocenjuje jedinicom. U suprotnom, ukoliko na skeniranoj površini nema ni jednog podatka odnosno ni jedne papile niti minucije, kvalitet se ocenjuje nulom. Za sve međuvrednosti izračunava se udeo površine sa informacijam u celokupnoj skeniranoj površini i izražava razlomkom odnosno decimalnim brojem.

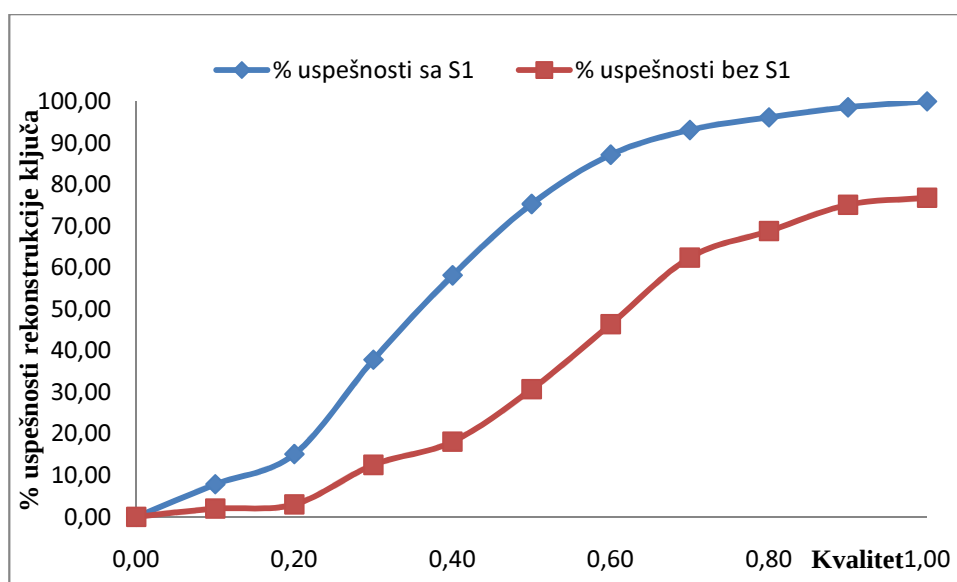
Pošto se radi o velikom broju otisaka rezultati testiranja svakog otiska kvaliteta otiska i procenat uspešne rekonstrukcije kriptografskog ključa su beleženi u posebne datoteke. Zbog velikog broja podataka a samim tim i merenja, uzete su presečne vrednosti kvaliteta slike otiska na deset karakterističnih mesta između nule i jedinice, kako bi se pregled mogao dati tabelarno. Nakon toga je izvršena aproksimacija jednom krivom koja sasvim sigurno oslikava trend pada procenta rekonstrukcije kriptografskog ključa sa padom kvaliteta slike.

Rezultati opisanog eksperimenta su dati numerički u sledećoj tabeli:

kvalitet	% uspešnosti sa S1	% uspešnosti bez S1
1,00	99,98	76,83
0,90	98,56	75,13
0,80	96,13	68,79
0,70	93,11	62,42
0,60	87,15	46,37
0,50	75,29	30,74
0,40	58,13	18,10
0,30	37,82	12,52
0,20	15,11	3,01
0,10	7,83	1,98
0,00	0,00	0,00

Tabela 5. Tabelarni prikaz eksperimentalno dobijenih vrednosti promene procenta uspešne rekonstrukcije kriptografskog ključa u zavisnosti od kvaliteta slike otiska

Ukoliko podatke prikažemo u vidu dijagrama, možemo uočiti njihov trend:



Slika 49. Grafički prikaz eksperimentalno dobijenih vrednosti promene procenta uspešne rekonstrukcije kriptografskog ključa u zavisnosti od kvaliteta slike otiska

Iz prikazanog se može zaključiti da uticaj prethodne provere podudarnosti šablona S_1 znatno

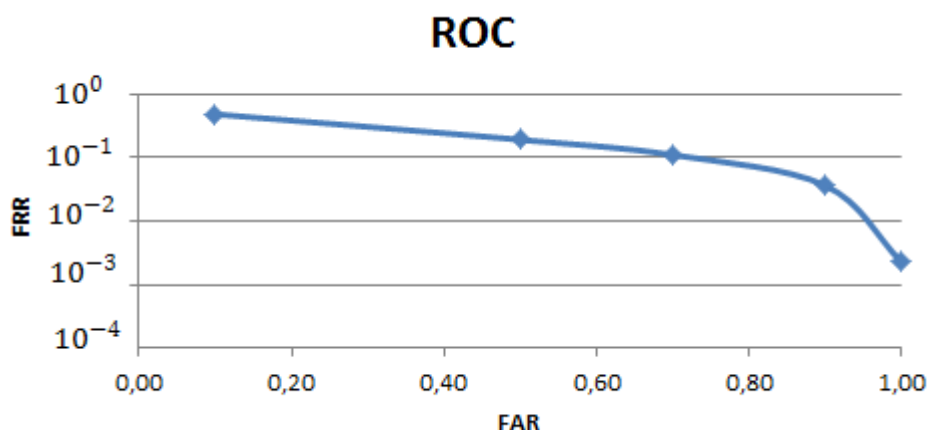
podizhe procenat uspešne rekonstrukcije kriptografskog ključa u predloženom rešenju. U slučaju kvaliteta slike sa oznakom 1, to povećanje iznosi čak 23,15%, što je veoma značajno povećanje. Čak iako se posmatra kriva koja prikazuje pad uspešnosti rekonstrukcije ključa, pri kvalitetu slike sa oznakom 1, postignuta vrednost od 76,38% nije zanemariva što znači da Slepian-Vulfovo kodiranje znatno uspeva da koriguje greške prilikom akvizicije otiska prsta. Ipak mora se konstatovati da je u praktičnoj primeni procenat uspešnost rekonstrukcije ključa od 76,38% gotovo neprimenjiv. To bi praktično značilo da se svaki četvrti ključ rekonstruiše kao neispravan.

ROC kriva

ROC (*Receiver Operating Characteristic*) ili kriva karakteristika rada prijemnika. U slučaju otiska prsta ne radi se o klasičnom prijemniku, već o distorziji otiska prsta prilikom skeniranja. Zadržan je naziv ROC, zbog svog porekla.

U narednom eksperimentu je zanemaren kvalitet slike. Posmatran je samo deo rešenja koji se bavi upoređivanjem otisaka. Iako se delimično radi o komercijalnom rešenju, izvršeno je praktična provera rada algoritma. Posmatran je odnos pogrešno uparenih otisaka (*Fault Acceptance Rate – FAR*) i otisaka koji su identični ali su proglašeni da nisu identični (*Fault Rejection Rate – FRR*).

Kompanija „Digital Persona“ objavila, algoritam koji koriste je algoritam opisan u poglavlju 4.1.3 sa modifikacijama koje kompanije nije javno objavila. Opisani algoritam je implementiran kroz softver koji je korišćen u ovom radu za upoređivanje dva otiska. Prema zvaničnim podacima koje je kompanija objavila njihov algoritam i softver daju izuzetne rezultate pri čemu FAR iznosi svega 0,01% dok je FRR tek 1,4%. Što su izvanredni rezultati. Ipak rezultati dobijeni eksperimentom ne pokazuju tako dobre rezultate. Uz ogradu da je softver napravljen za testiranje nesavršen, ROC kriva dobija sledeći izgled:



Slika 50. ROC kriva

Testiranje na velikim grupama ljudi tokom dužeg perioda je pokazalo da 80 % svih korisnika

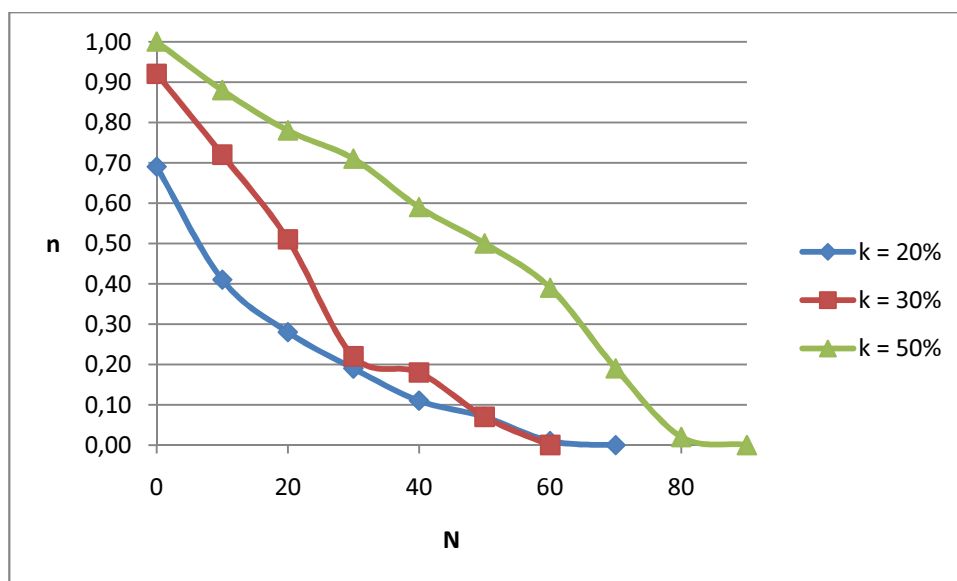
ima dovoljno karakteristika na otiscima prstiju i gotovo nikada se pogrešno ne prihvataju ili odbacuju. Od preostalih 20% , oko 15 % korisnika ima manje detalja u svojim otiscima. Oni moraju da stave prst na senzor barem dva puta da bi bili sigurni da je otisak uzet kvalitetno. Preostalih 5% populacije poseduje otiske prstiju koje je gotovo nemoguće skenirati. Najveći broj ljudi iz ove populacije je Azijskog porekla.

Analiza mogućeg napada

Kvantizacija korekcije greške kod definiše koliko grešaka može biti korigovano prilikom kodiranja. Ukoliko kvantizaciju označimo sa k i uzmemo pretpostavljene vrednosti: $k = 20\%$, $k = 30\%$ i $k = 50\%$. Za veće k sistem će korigovati više grešaka i biće pouzdaniji, ali će biti više izložen napadima pošto će informacija sadržati više tačnih bitova. Za manje vrednosti k biće manje tačnih bitova ali će ih zato biti teže rekonstruisati prilikom napada. Da bi se pronašlo najbolje moguće rešenje uzete su vrednost za k kako je i navedeno i vršena su testiranja tako što su korišćeni otisci istog otiska prsta različitog kvaliteta tako da je svaki otisak sadržao različiti broj informacija o minucijama. U cilju izvođenja eksperimenta broj informacija o minucijama je namerno smanjivan na definisane procenete kako bi se dobila jasna slika o uticaju broja informacija o minucijama, odnosno njihovoj izloženosti napadima.

Ukoliko ukupan mogući broj osobina obeležimo sa velikim N , a sa malim n obeležimo broj osobina uočen odnosno u eksperimentu dodeljen broj osobina pojedinom otisku. Nakon toga vršimo merenja za vrednost za k kako je navedeno. Nakon toga vrše se izračunavanja srednjih vrednosti za sve testirane otiske kak bi se sagledala celina.

Kao rezultat se dobija sledeći dijagram:



Slika 51. Izloženost podataka napadu

Može se zaključiti da je izloženost napadu najveća kada je broj korektnih bitova najveći. NA dijagramu je očigledno da je linija za $k = 50\%$ najduža, i na taj način i vizuelno postaje očigledno da je i najviše izložena potencijalnom napadu.

Eksperimentalni rezultati su pokazali neke od karakteristika predloženog rešenja od kojih je najbitniji prvi prikazani eksperiment koji potvrđuje opravdanost koncepta.

7 Zaključak

Sumirajući rezultate rada moguće je uočiti ostvaren cilj i potvrdu hipoteze kroz empirijsko istraživanje i formiranja originalnog rešenja u oblasti transformacije i zaštite biometrijskih podataka u kriptografskim operacijama

U radu je dat pregled izabranih svetskih iskustava iz oblasti zaštite biometrijskih podataka. Može se konstatovati da je mnogo pažnje posvećeno navedenoj oblasti u stručnoj javnosti. Pojedini istraživači su posvetili čitav svoj profesionalni vek izučavajući sisteme za zaštitu biometrijskih podataka. U prezentovanom radu je izvršen izbor pojedinih dela pojedinih autora i očigledno je da mnoga značajna razmatranja nisu obrađena u radu. Ipak izvršen je izbor, prema percepciji autora, najznačajnijih svetskih iskustava iz oblasti koja je tema ovog rada.

Dati su pregled i analiza postojećih softverskih alata za zaštitu biometrijskih podataka. Pored korišćenog komercijalnih alata kompanije „Digital Persona“ iz Redwood city-ja u Kaliforniji, koja je jedina pristala da otvori svoje SDK alate, ostale fime poput firme „Aware“ iz Bostona, svoje alate čuva kao najstrožu poslovnu tajnu. Treba imati na umu da kompanija softverski alati nastaju u različitim kompanijama ili od strane različitih pojedinaca. Ipak, prema zakonima tržišta i odnosa moći u svetu, sva dostignuća završavaju u institucijama kao što je NSA/CSS (*The National Security Agency/Central Security Service*). Iz tih razloga izbor slobodnih rešenja je relativno ograničen.

U radu je dat je pregled tehnologija koje se koriste za projektovanje i implementaciju sistema za testiranje i analizu matematičkih modela korišćenih funkcija. Veliku količinu već gotovog koda je moguće pronaći na različitim Internet lokacijama. Ipak, uvek se ispostavljalo da je za rešavanje konkretnog problema neophodno doraditi kod kako bi izvršavao željenu funkciju. Često je efikasnije izraditi ceo kod od početka nego ulaziti u tuđi kod koji je rađen prema viziji autora. Može se desiti da je programski kod konfuzan i nerazumljiv za sve druge programere osim za svog autora.

Opisano je korišćenje standardnih metodologija i alata za softversko modelovanje radi proširivanja funkcionalnosti postojećih aplikacija za generisanje binarnog niza. Takođe su pomenute hardverske metode za generisanje slučajnog niza i napravljena je distinkcija između hardverski i softverski generisanog niza. Prikazan je elementarni matematički aparat koji definiše pseudoslučajni niz.

Rezime doprinosa

- Izvršena je kritička analiza dosadašnjih istraživanja u oblasti transformacije biometrijskih podataka u binarni niz i dostignuća u oblasti primene kodova za korekciju greške na binarnim nizovima dobijenim na osnovu biometrijskih podataka. Sagledana su aktuelna predložena rešenja u oblasti zaštite biometrijskih podataka, s posebnim osvrtom na zaštitu otiska prsta. Analizirani su i testirani postojeći softverski alati za zaštitu biometrijskih podataka;
- Eksperimentalno je potvrđena transformacija biometrijskih podataka i njihova primena u kriptografskim operacijama. Korišćene su standardne metodologije i alati za softversko modelovanje radi proširivanja funkcionalnosti postojećih aplikacija za obradu biometrijskih podataka;
- Projektovano je i razvijeno softversko rešenje koje je omogućilo testiranje i analizu izabranih matematičkih modela;
- Kreirane su sopstvene klase u C++ i C# programskim jezicima u svrhu transformacije i zaštite biometrijskih podataka;
- Izvršena je sinteza rezultata i dat je predlog smernica za razvoj i primenu biometrijskih podataka u kriptografskim operacijama.

Originalni softver

Za potrebe testiranja predloženog rešenja korišćene su gotove softverske komponente u meri u kojoj je to bilo moguće, odnosno u meri u kojoj su zadovoljavala potrebe rada. Istini za volju, najčešće je trebalo vršiti određene korekcije. Velik i deo softvera je originalno razvijen i pojedine komponente se mogu operativno koristiti. Ceo softver nije integrisan u jednu celinu već se sastoji od modula koji rade autonomno.

Ne postavlja se pitanje mogućnosti integracije softvera u jednu celinu, ali se postavlja pitanje svrsishodnosti tog projekta? Moguće je integrisati sve module i jedan programski paket ali bi taj paket radio striktno jedan posao a to je podrška iznetim idejama u ovom radu, što ne bi bilo od velike koristi za ostale korisnike. Modularni pristup omogućava nezavisnim korisnicima da koriste softverske module nezavisno jedan od drugog, već prema vlastitoj potrebi.

Izabrani delovi softverskog koda su dati u prilogu. Kao što je napomenuto, kod je pisan i programskim jezicima C/C++ i C#. Jezici C i C++ su pisani za slučajeve kada je bilo neophodno obezbediti direktan pristup hardveru, dok je C# uglavnom bio korišćen za pisanje korisničkog interfejsa, mada je bilo i izuzetaka.

Kompjuterski jezici su izabrani kao kompajlerski jezici koji su znatno stabilniji od interpreterskih i nezavisni su od platforme na kojoj se izvršavaju.

Predlog smernica za dalji rad

U radu koji je predstavljen kao i u iznetim idejama neke teme su samo započete a njihova rešenja su naznačena u obrisima. Razrada pojedinih komponenti otvara neograničene mogućnosti razvoja. Pogotovo imajući u vidu izuzetno brz napredak u ovoj oblasti kao i sve veće dobijanje zaštite biometrijskih podataka na značaju, kao i zaštite podataka uopšte.

Imajući u vidu široku dostupnost informacija i revoluciju u razmeni znanja koju je omogućio Internet, u savremenom svetu je svako novo istraživanje ili novi rezultat gotovo trenutno vidljiv svim zainteresovanima koji se određenom oblasti bave.

Na osnovu ovog rada, kao i mnogobrojnih radova na koje se ova teza naslanja, potrebno je pre svega obratiti pažnju na način akvizicije i obrade otiska prsta. Iako je otisak prsta jedna od najstarijih biometrijskih metoda identifikacije, nije izgubila na svom značaju, naprotiv. Daljim istraživanjem se mogu definisati nove metode obrade otisaka prsta i njegovog prevođenja u templejt. Moguća su i razmišljanja u smeru napuštanja minucija kao osnove sadašnjeg prepoznavanja otiska kao i otvaranje nekih novih poglavlja u oblasti zaštite biometrije otiska prsta.

Naučni izazov koji se ispoljio kao neograničeno polje za istraživanje su kodovi za korekciju greške. U ovoj tezi su korišćeni Slepian Vulfovi kodovi kao jedno od mogućih rešenja. Sigurno je da postoje još mnoga druga, znatno bolja rešenja do kojih se može doći kombinovanjem različitih kodova za korekciju greške ili značajnijim prodorom u ovoj oblasti, može doći i do kreiranja novih vrsta kodova za korekciju greške.

Rad koji je predstavljen je dotakao određene probleme uz izložena moguća rešenja kao podsticaj i oslonac za dalja istraživanja transformacije biometrijskih podataka i njihovu primenu u kriptografskim operacijama.

Literatura

[1] Philadelphia Convention; *United States Constitution*; National Archives; Washington, D.C., 1787.

[2] Jain, A.K. ; Dept. of Comput. Sci. & Eng., Michigan State Univ., USA ; Ross, A. ; Prabhakar, S.; *An Introduction to Biometric Recognition*; Pages: 4 - 20; ISSN: 1051-8215; INSPEC Accession Number: 7952000; Digital Object Identifier: 10.1109/TCSVT.2003.818349; IEEE Circuits and Systems Society; Januar 2004.

[3] Sanchez-Reillo, R. ; ETSI Telecommun.-Dept., Ciudad Univ., Madrid, Spain ; Sanchez-Avila, C. ; Gonzalez-Marcos, A., IEEE; *Biometric identification through hand geometry measurements*; Pages: 1168 – 1171; ISSN: 0162-8828; INSPEC Accession Number: 6766377; Digital Object Identifier: 10.1109/34.879796; Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:22 , Issue: 10); Oktobar 2000.

[4] Peter Komarinski, *Automated Fingerprint Identification Systems (AFIS)*, Academic Press, 2005.

[5] Yi Chen, Sarat Dass, Arun Ross, and Anil Jain, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48823; *Fingerprint Deformation Models Using Minutiae Locations and Orientations*; Appeared in Proc. of IEEE Workshop on Applications of Computer Vision (WACV), (Breckenridge, Colorado), pp. 150-155, January 2005.

[6] Mladen Veinović, Goran Kunjadić, *Appliance of multimodal biometrics systems on the optical memory cards with chip*, e-Smart conference, 2008, Nice, France

[7] Mladen Veinović, Goran Kunjadić, *Generating Sketch on the Chip Card and Online Data Matching*, e-Smart conference 2009, Nice, France

[8] Chaur-Chin Chen and Yaw-Yi Wang, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 30043; Palmerston North, *An AFIS Using Fingerprint Classification*; New Zeland, November 2003

[9] Uludag, U. ; Dept. of Comput. Sci. & Eng., Michigan State Univ., USA ; Pankanti, S. ; Prabhakar, S. ; Jain, A.K.; *Biometric cryptosystems: issues and challenges*, Volume:92 Issue:6; Pages: 948 - 960; ISSN: 0018-9219; INSPEC Accession Number: 8021899; Digital Object Identifier: 10.1109/JPROC.2004.827372; Jun 2004.

- [10] Xudong Jiang ; Centre for Signal Process., Nanyang Technol. Inst., Singapore ; Wei-Yun Yau, IEEE; *Fingerprint minutiae matching based on the local and global structures*; Pages: 1038 - 1041 vol.2; ISSN: 1051-4651; Print ISBN: 0-7695-0750-6; INSPEC Accession Number: 6894486; Digital Object Identifier: 10.1109/ICPR.2000.906252; Barcelona, Septembar 2000.
- [11] Vlastimir Vučić, *Osnovna merenja u fizici*, Naučna knjiga, 1986, Beograd, Jugoslavija
- [12] Anil K. Jain, Michigan State University ;Jianjiang Feng, Tsinghua University, Beijing; Karthik Nandakumar, Institute for Infocomm Research; *Fingerprint matching*; 0018-9162/10 © 2010 IEEE
- [13] Jianjiang Feng and Anil K. Jain Department of Computer Science and Engineering Michigan State University; *FM Model Based Fingerprint Reconstruction from Minutiae Template*; Proc. ICB 2009, Alghero, Italy: 544-553; 2009.
- [14] Abinandhan Chandrasekaran Dr.Bhavani Thuraisingham, Data and Applications Security Laboratory, Department of Computer Science, University of Texas at Dallas; *Fingerprint Matching Algorithm Based on Tree Comparison using Ratios of Relational Distances*; Vienna, Austria, 2007.
- [15] J.A Bouwstra, P.L Honeywell-Nguyen; *Skin structure and mode of action of vesicle*; Advanced Drug Delivery Reviews; Volume 54, Supplement, Pages S41–S55; The Netherlands, Novembar 2002.
- [16] Yi Chen, East Lansing, Dass, S., Ross, A., Jain, A.; *Fingerprint deformation models using minutiae locations and orientations*; Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops; Pages: 150 - 155; Print ISBN: 0-7695-2271-8; INSPEC Accession Number: 10265639; Digital Object Identifier: 10.1109/ACVMOT.2005.54; Breckenridge, Januar 2005.
- [17] Susan Standring; *Gray's Anatomy, the anatomical basis of clinical practice, expert consult*; ISBN: 978-0-443-06684-9; Amsterdam, 2009.
- [18] Andrey S. Krylov, Elena A. Pavelyeva; *Iris data parametrization by Hermite projection method*; Faculty of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia, 2007.
- [19] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller; *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*; University of Massachusetts, Amherst, Technical Report 07-49, 2007.
- [20] Volker Blanz, Thomas Vette; *Face recognition based on fitting a 3D morphable model*; IEEE Transactions on pattern analysis and machine intelligence, Vol. 25, No. 9, Septembar 2003.
- [21] Sabine Delaitre; *DNA: future biometric identifier*; Smart University, Biometrics Track, , European Commision, 2008.

- [22] Yuk Fai Leung, Duccio Cavalieri; *Fundamentals of cDNA microarray data analysis*; TRENDS in Genetics Vol.19 No.11 Novembar 2003.
- [23] Marie Roch; *Gaussian-selection-based non-optimal search for speaker identification*; San Diego State University, 5500 Campanile Drive, San Diego, CA 92182-7720, USA Jun 2005.
- [24] Darren R. Gitelmana, Todd B. Parrisha, Karl J. Fristone, M.-Marsel Mesulama; *Functional Anatomy of Visual Search: Regional Segregations within the Frontal Eye Fields and Effective Connectivity of the Superior Colliculus*; Elsevier Science, Volume 15, Issue 4, April 2002, Pages 970–982; USA, 2002.
- [25] Ravi Das; *Retinal recognition Biometric technology in practice*; Keesing Journal of Documents & Identity, issue 22, 2007.
- [26] Sang-Kyun Im, Hyung-Man Park, Young-Woo Kim, Sang-Chan Han, Soo-Won Kim and Chul-Hee Kang; *An Biometric Identificaton System by Extracting Hand Vein Patterns*; Journal of the Korean Physical Society, Vol. 38, No. 3; Seoul, South Korea; Mart 2001.
- [27] Ishani Sarkar, Farkhod Alisherov, Tai-hoon Kim, and Debnath Bhattacharyya; *Palm Vein Authentication System:A Review*; International Journal of Control and Automation Vol. 3, No. 1; Seoul, South Korea; Mart, 2010.
- [28] Plamondon R., Srihari S.N.; *Online and off-line handwriting recognition: a comprehensive survey*; Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:22 , Issue: 1);Pages: 63 - 84, ISSN:0162-8828, INSPEC Accession Number:6525227, Digital Object Identifier: 10.1109/34.824821; 2002.
- [29] Haiyun Xu, Raymond N.J. Veldhuis ; *Binary Representations of Fingerprint Spectral Minutiae Features*; Department of Electrical Engineering, University of Twente; The Netherlands, 2010.
- [30] Abinandhan Chandrasekaran, Bhavani Thuraisingham; *Fingerprint Matching Algorithm Based on Tree Comparison using Ratios of Relational Distances*; University of Texas at Dallas, ISBN: 0-7695-2775-2; Vienna, Austria, 2013.
- [31] Asker M. Bazen, Sabih H. Gerez; *Segmentation of Fingerprint Images*; University of Twente, Department of Electrical Engineering, Laboratory of Signals and Systems, The Netherlands, 2001.
- [32] L.A. Zadeh; *Fuzzy Sets*; Department of Electrical Engineering and Electronics Research Labaratory, University of California, Barkeley; California, USA, 1965
- [33] Lior Cohen, Gil Avrahami, Mark Last; *Incremental Info-Fuzzy Algorithm for Real Time Data Mining of Non-Stationary Data Streams*; Ben-Gurion University of the Negev; Israel, 2008.
- [34] Stephan Sigg; *Secure communication based on noisy input data Fuzzy extractors and identity based encryption* ;Technische Universitat Braunschweig, 2011

- [35] Ahmad Shahi, Rodziah Binti Atan, Nasir Sulaiman; *Detecting Effectiveness of Outliers and Noisy Data on Fuzzy System Using FCM*; European Journal of Scientific Research, ISSN 1450-216X Vol.36 No.4 (2009), pp.627-638; 2009.
- [36] Yevgeniy Dodisy Bhavana Kanukurthiz Jonathan Katzx, Leonid Reyzin, Adam Smith; *Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets*; USA, 201.
- [37] Milan M. Milosavljević , Gojko Grubor,; *Osnove zaštite informacija*; ISBN: 978-86-7912-313-8; Univerzitet Singidunum, Beograd, 2010.
- [38] A Menezes, P van Oorschot, S Vanstone; *Handbook of Applied Cryptography*; CRC Press, ISBN: 0-8493-8523-7; 1996.
- [39] Thomas Markwig; *Algebraic Structures*; Fachbereich Mathematik, Technische Universität Kaiserslautern; Deutschland, 2009.
- [40] Veronique Cortier, Steve Kremer, Bogdan Warinschi; *A Survey of Symbolic Methods in Computational Analysis of Cryptographic Systems*; Journal of Automated Reasoning, Volume 46, Issue 3-4, pp 225-259; Springer, April 2011.
- [41] Ayushi; *A Symmetric Key Cryptographic Algorithm*; Hindu College of Engineering, International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 15; 2010.
- [42] H. Prafullchandra, J. Schaad; *Diffie-Hellman Proof-of-Possession Algorithms*; Working Group; USA, 2000.
- [43] Andrew Nash, William Duane, Celia Joseph; *PKI: Implementing and Managing E-Security*; McGraw-Hill, Inc., ISBN:1590610008; New York, NY, USA, 2001.
- [44] Ilya Mironov; *Hash functions: Theory, attacks, and applications*; Microsoft Research, Silicon Valley Campus; 2005.
- [45] Ramachandran Ganesan, Mohan Gobi, Kanniappan Vivekanandan; *A Novel Digital Envelope Approach for A Secure E-Commerce Channel*; International Journal of Network Security, Vol.11, No.3; 2010.
- [46] Shimon Even, Oded Goldreich, Silvio Micali; *On-Line/Off-Line Digital Signatures*; Online ISBN 978-0-387-34805-6Springer, New York, 1990.
- [47] Gao Zheng-xian,Tu Ya-qing,Li Zhong-xue; *Design and Implementation of PKI and RBAC-authorized Digital Certificate*; Department of Logistics Information Engineering, Logistics Engineering University, Chongqing; 2008.
- [48] Nicolas T. Courtois, Josef Pieprzyk ; *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*; Advances in Cryptology - ASIACRYPT, 2002.
- [49] Robin Schriebman; *Error Correcting Code*; Google; USA, 2006.

- [50] Michael Mitzenmacher, Eli Upfal; *Probability and Computing – randomized algorithms and probabilistic analysis*; ISBN 0 521 83540 2; Cambridge, 2005.
- [51] Milan M. Milosavljević, Saša Adamović; *Osnovi teorije informacija i kodovanje*; ISBN: 978-86-7912-506-4; Univerzitet Singidunum, Beograd, 2013.
- [52] Bahl, L., Cocke, J., Jelinek, F., Raviv, J.; *Optimal decoding of linear codes for minimizing symbol error rate*; ISSN: 0018-9448; Digital Object Identifier: 10.1109/TIT.1974.1055186; IEEE Information Theory Society, 2003.
- [53] Richard W. Hamming ; *Coding and information theory*; ISBN:0-13-139072-4; Naval Postgraduate Schoool, Prentice-Hall, Inc.,1986,
- [54] Jun Chen, Dake He, Ashish Jagmohan ; *Slepian-Wolf Code Design via Source-Channel Correspondence*; Urbana, IL, USA, 2006.
- [55] Jagadish Venkataraman; *Analysis of slepian wolf coding*; University of Notredame, USA, 2006.
- [56] Jianjiang Feng, Anil K. Jain; *Filtering Large Fingerprint Database for Latent Matching*; Department of Computer Science and Engineering, Michigan State University; 2008.
- [57] Chengfeng Wang, Marina Gavrilova, Yuan Luo and Jon Rokne; *An efficient algorithm for fingerprint matching*; Conference on Pattern Recognition (ICPR'06) 0-7695-2521-0/06; IEEE; University of Calgary, Canada, 2006.
- [58] Anil K. Jain, Jianjiang Feng, Abhishek Nagar, Karthik Nandakumar; *On Matching Latent Fingerprints*; Department of Computer Science and Engineering, Michigan State University and Institute for Infocomm Research Fusionopolis, Singapore; 2008.
- [59] Net Ning Liu, Yilong Yin, Hongwei Zhang ; *A Fingerprint Matching Algorithm Based On Delaunay Triangulation*; Computer Department, Shandong University, Jinan, 250100, China, 2005.

Spisak slika

1. Skener otiska prsta korišćen u istraživanju	16
2. Šematski prikaz čitanja biometrijskih podataka i njihovo upisivanje u bazu podataka	21
3. Šematski prikaz verifikacije identiteta korisnika	21
4. Šematski prikaz identifikacije korisnika	22
5. Primer skupa minucija jednog otiska prsta	29
6. Označavanje koordinata izabrane minucije	30
7. Uporedni prikaz raspodele sličnosti za pravi i lažni otisak	33
8. ROC (Receiver Operating Curve) kriva	35
9. Nabori na jagodici prsta su predmet istraživanja	37
10. Globalne karakteristike otiska prsta	38
11. Snimak oka	41
12. Uočavanje lokacije irisa	41
13. Pravougaona - normalizovana slika irisa	41
14. Pravougaona oblast koja se koristi za upoređivanje	41
15. Karakteristične tačke koje se koriste za prepoznavanje lika u 2D tehnologiji	43
16. 3D mreža lika	44
17. Pojednostavljeni grafički prikaz DNK	45
18. Upoređivanje segmenata DNK	46
19. Grafički prikaz upoređivanja dva glasovna templejta	47
20. Pojednostavljeni prikaz građe oka	48
21. Pojednostavljena šema akvizicije rasporeda vena	50

22. Definisanje međusobnog položaja minucija	53
23. Grafički prikaz zona tolerancije	55
24. Šematski prikaz procesa upoređivanja dva otiska prsta	57
25. Prenosna funkcija	60
26. Grafička interpretacija brzine promene greške	62
27. Karakteristike funkcije pripadnosti	65
28. Grafička interpretacija fazi težišta	66
29. Šematski prikaz bezbednosnog skeča	68
30. Šematski prikaz fazi ekstraktora	69
31. Šematski prikaz procesa razmene šifrovanih podataka	75
32. Tjuringova mašina	77
33. Šematski prikaz procesa šifrovanja	95
34. Šematski prikaz prenosa poruke kroz kanal sa šumom	102
35. Struktura bloka podataka kod Rid-Solomonovog kodiranja	106
36. Jedna varijacija Slepian-Vulfovog kodiranja	108
37. Grafički prikaz oblasti ostvarivosti	109
38. Varijacija Slepian-Vulfovog kodiranja	109
39. Vizuelizacija procesa identifikacije minucija i procene pravca papila	115
40. Kreiranje Deloneove mreže	120
41. Deloneova mreža na otisku prsta	120
42. Šematski prikaz predloženog rešenja	129
43. Kontrolni panel osnovnog softverskog rešenja	133
44. Uzimanje otiska sa čitača	133
45. Skeniranje otiska prsta	134
46. Faza završetka kreiranja templejta	134
47. Pozitivna verifikacija otiska	135
48. Negativna verifikacija otiska	135

49. Grafički prikaz eksperimentalno dobijenih vrednosti promene procenta uspešne rekonstrukcije kriptografskog ključa u zavisnosti od kvaliteta slike otiska	137
50. ROC kriva	138
51. Izloženost podataka napadu	139

Spisak tabela

1. Način određivanja karakterističnih tačaka – minucija	39
2. Osnovni oblik matrice pravila	63
3. Autokorelaciona funkcija – primer	94
4. Sabiranje i množenje nad konačnim poljem $\mathbb{F}_2 = \{0, 1\}$	104
5. Tabela prikaz eksperimentalno dobijenih vrednosti promene procenta uspešne rekonstrukcije kriptografskog ključa u zavisnosti od kvaliteta slike otiska	137

Dodatak 1

Komunikacija sa skenerom za otisak prsta i kreiranje binarnog i heksadecimalnog stringa (templejta)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Enrollment
{
    /* napomena: ova forma je osnova za EnrollmentForm and the VerificationForm,
    sve promene na CaptureForm ce se odraziti na izvedene forme.
    */
    public partial class CaptureForm : Form, DPFP.Capture.EventHandler
    {
        public CaptureForm()
        {
            InitializeComponent();
        }

        protected virtual void Init()
        {
            try
            {
                // kreira operaciju skeniranja
                Capturer = new DPFP.Capture.Capture();
                // zapocinje dogadjaj kreiranja skeniranja
                if ( null != Capturer )
                    Capturer.EventHandler = this;
                else
                    SetPrompt("Ne mogu da inicijalizujem skeniranje!");
            }
            catch
            {
                MessageBox.Show("Ne mogu da zapocnem skeniranje!", "Greska", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
        }

        protected virtual void Process(DPFP.Sample Sample)
        {
            // crta sliku oriska prsta
            DrawPicture(ConvertSampleToBitmap(Sample));
        }

        protected void Start()
        {
            if (null != Capturer)
            {
                try
                {
                    Capturer.StartCapture();
                    SetPrompt("Koristeci skener za otisak prsta, skenirajte otisak");
                }
            }
        }
    }
}
```

```

    }
    catch
    {
        SetPrompt("Ne mogu da inicijalizujem citanje otiska!");
    }
}

protected void Stop()
{
    if (null != Capturer)
    {
        try
        {
            Capturer.StopCapture();
        }
        catch
        {
            SetPrompt("Ne mogu da zaustavim skeniranje!");
        }
    }
}

#region Form Event Handlers:

private void CaptureForm_Load(object sender, EventArgs e)
{
    Init();
    Start(); // zapocinjanje prihvatanja slike
}

private void CaptureForm_FormClosed(object sender, FormClosedEventArgs e)
{
    Stop();
}

#endregion

#region EventHandler Members:

public void OnComplete(object Capture, string ReaderSerialNumber, DPFP.Sample Sample)
{
    MakeReport("Otisak je uzet");
    SetPrompt("Skenirajte prst");
    Process(Sample);
}

public void OnFingerGone(object Capture, string ReaderSerialNumber)
{
    MakeReport("Sklonite prst sa citaca");
}

public void OnFingerTouch(object Capture, string ReaderSerialNumber)
{
    MakeReport("Prst je na citacu");
}

public void OnReaderConnect(object Capture, string ReaderSerialNumber)
{
    MakeReport("Citac otisaka je prikljucen");
}

public void OnReaderDisconnect(object Capture, string ReaderSerialNumber)
{
    MakeReport("Citac otisaka nije prikljucen");
}

public void OnSampleQuality(object Capture, string ReaderSerialNumber,
DPFP.Capture.CaptureFeedback CaptureFeedback)
{
    if (CaptureFeedback == DPFP.Capture.CaptureFeedback.Good)
        MakeReport("Kvalitet otiska je dobar");
}

```

```

        else
            MakeReport("Kvalitet otiska je los");
    }
#endregion

protected Bitmap ConvertSampleToBitmap(DPFP.Sample Sample)

// kreiranje konvertora
DPFP.Capture.SampleConversion Convertor = new DPFP.Capture.SampleConversion();
// Nije bitna velicina
Bitmap bitmap = null;

// vraca bitmap kao rezultat
Convertor.ConvertToPicture(Sample, ref bitmap);

int niz_bin = Sample.Size;

byte[] vr = Sample.Bytes;

string hex = BitConverter.ToString(vr);
hex = hex.Replace("-", "");

// kreiranje upisa heksadecimalnog niza u fajl
TextWriter tw = new StreamWriter("C:\\hex.txt");

// upisivanje u fajl
tw.WriteLine(hex);

// zatvaranje stream-a
tw.Close();

// pripremanje binarnog stringa
string bin = "";
bin = Convert.ToString(Convert.ToInt32(hex, 16), 2);
bin = Convert.ToString(Convert.ToUInt64(hex, 16), 2);
Convert.ToUInt64

// kreiranje upisa binarnog niza u fajl
TextWriter tw1 = new StreamWriter("C:\\bin.txt");

// upisivanje u fajl
tw1.WriteLine(bin);

// zatvaranje stream-a
tw1.Close();

// kreiranje hesh vrednosti
int hesh = Sample.GetHashCode();

string nizz = Convert.ToString(niz_bin);

MessageBox.Show (nizz);

return bitmap;
}

Purpose) protected DPFP.FeatureSet ExtractFeatures(DPFP.Sample Sample, DPFP.Processing.DataPurpose
{
    // kreiranje ekstraktora karakteristika
    DPFP.Processing.FeatureExtraction Extractor = new DPFP.Processing.FeatureExtraction();

    DPFP.Capture.CaptureFeedback feedback = DPFP.Capture.CaptureFeedback.None;
    DPFP.FeatureSet features = new DPFP.FeatureSet();
    // vraca karakteristike kao rezultat
    Extractor.CreateFeatureSet(Sample, Purpose, ref feedback, ref features);

    if (feedback == DPFP.Capture.CaptureFeedback.Good)
    {
        return features;
    }
}

```

```
else
    return null;
}

protected void SetStatus(string status)
{
    this.Invoke(new Function(delegate()
    {
        StatusLine.Text = status;
    }));
}

protected void SetPrompt(string prompt)
{
    this.Invoke(new Function(delegate()
    {
        Prompt.Text = prompt;
    }));
}

protected void MakeReport(string message)
{
    this.Invoke(new Function(delegate() {
        StatusText.AppendText(message + "\r\n");
    }));
}

private void DrawPicture(Bitmap bitmap)
{
    this.Invoke(new Function(delegate()
    {
        // podesavanje sike na velicinu okvira u formi
        Picture.Image = new Bitmap(bitmap, Picture.Size);
        Picture.Image.Save("C:\\slika.bmp");
    }));
}

private DPFPCapture.Capture Capturer;
}
}
```

Dodatak 2

Verifikacija otiska

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Enrollment
{
    /* Napomena: Ova forma je nasledjena iz forme CaptureForm,
       moze da se desi da je VisualStudio Form Designer ne ucita korektno
       (bar dok se ne uradi build projekta).
       Ako zelite da izmenite u dizajnu forme - ucinite to u osnovnoj formi - CaptureForm.
       Sve promene u CaptureForm ce se odraziti na izvedene forma
       (npr. na EnrollmentForm i na VerificationForm)
    */
    public class VerificationForm : CaptureForm
    {
        public void Verify(DPFP.Template template)
        {
            Template = template;
            ShowDialog();
        }

        protected override void Init()
        {
            base.Init();
            base.Text = "Verifikacija otiska";
            // kreirara template verifikatora otiska
            Verificator = new DPFP.Verification.Verification();
            UpdateStatus(0);
        }

        protected override void Process(DPFP.Sample Sample)
        {
            base.Process(Sample);

            // obradjuje otisak i kreira skup karakteristika za verifikaciju otiska.
            DPFP.FeatureSet features = ExtractFeatures(Sample,
            DPFP.Processing.DataPurpose.Verification);

            // Proverava kvalitet otiska i zapocinje verifikaciju ako je otisak dobar
            if (features != null)
            {
                // uporedjuje skup karakteristika sa nasim templejtom
                DPFP.Verification.Verification.Result result = new
                DPFP.Verification.Verification.Result();
                Verificator.Verify(features, Template, ref result);
                UpdateStatus(result.FARAchieved);
                if (result.Verified)
                    MakeReport("Otisak je VERIFIKOVAN!");
                else
                    MakeReport("Otisak NIJE VERIFIKOVAN!");
            }
        }

        private void UpdateStatus(int FAR)
        {
            // pokazuje vrednost "False accept rate"
        }
    }
}
```

```
        SetStatus(String.Format("Stopa pogresne verifikacije (Fault Acceptance Rate - FAR)
je = {0}", FAR));
    }

    private DFPF.Template Template;
    private DFPF.Verification.Verification Verificator;

    private void InitializeComponent()
    {
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(VerificationForm));
        this.SuspendLayout();
        //
        // forma za verifikaciju
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.ClientSize = new System.Drawing.Size(581, 354);
        this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
        this.Name = "VerificationForm";
        this.ResumeLayout(false);
        this.PerformLayout();
    }
}
```

Dodatak 3

Izdvajanje minucija i kreiranje templejta Deloneovom triangulacijom

```
// Izdvajanje minucija i uporedjivanje

using System;
using System.Drawing;
using System.Drawing.Drawing2D;

namespace BiometricsSDK.OtisakPrsta
{
    /// <summary>
    /// Summary opis klase
    /// </summary>

    public class COTisakPrsta
    {
        //public int FP_SIRINA_SLIKE = 500;
        //public int FP_VISINA_SLIKE = 500;
        public int FP_SIRINA_SLIKE = 323;
        public int FP_VISINA_SLIKE = 352;
        public int FP_MAX_VELICINA_TEMPLEJTA = 601;
        public int FP_DOZVOLJENO_ODSTUPANJE = 10;
        // rotacija
        public int FP_ROTACIJA = 10;//10;
        // prag
        public int FP_PRAG = 55;
        // klasifikacija
        public const int FP_VRTLOG = 1 ;
        // tip otiska
        public const int FP_LEVA_PETLJA = 2;
        public const int FP_DESNA_PETLJA = 3;
        public const int FP_LUK = 4;
        public const int FP_LUK_SA_KROVOM = 5;
        // parametri za kreiranje templejta
        private const int FP_VELICINA_TEMPLEJTA = 0;
        private const int FP_POCETAK_TEMPLEJTA_X = 1;
        private const int FP_POCETAK_TEMPLEJTA_Y = 2;
        private const int FP_VELICINA_KARAKTERISTIKE_TEMPLEJTA = 6;
        private const int FP_RADIUS_TEMPLEJTA = 1;

        public byte [,] P = new byte[500,500];

        public COTisakPrsta()
        {
        }

        public COTisakPrsta(int width , int height )
        {
            FP_SIRINA_SLIKE = width;
            FP_VISINA_SLIKE = height;
            P = new byte[width,height];
        }

        public COTisakPrsta(int width, int height, int MatchPointDistanceMovement, int
        MatchPointRotationMovment, int MatchThreshold)
        {
            FP_SIRINA_SLIKE = width;
            FP_VISINA_SLIKE = height;
            FP_DOZVOLJENO_ODSTUPANJE = MatchPointDistanceMovement;
        }
    }
}
```

```

FP_ROTACIJA = MatchPointRotationMovment;
FP_PRAG = MatchThreshold;
}

public void setFingerPrintImage(Bitmap m_image)
{
try
{
    for (int i = 0; i <= FP_SIRINA_SLIKE - 1; i++)
    {
        for (int j = 0; j <= FP_VISINA_SLIKE - 1; j++)
        {
            if ((m_image.GetPixel(i, j).B <= 127) && (m_image.GetPixel(i, j).R <= 127) &&
(m_image.GetPixel(i, j).G <= 127))
            {
                P[i, j] = 1;
            }
            else
            {
                P[i, j] = 0;
            }
        }
    }
    // podesavanje ivica na nulu
    for (int i = 0; i <= FP_SIRINA_SLIKE - 1; i++)
    {
        P[i, 0] = 0;
        P[i, FP_VISINA_SLIKE - 1] = 0;
    }
    for (int j = 0; j <= FP_VISINA_SLIKE - 1; j++)
    {
        P[0, j] = 0;
        P[FP_SIRINA_SLIKE - 1, j] = 0;
    }
}
catch (Exception ex)
{
    throw ex;
}
}

public Bitmap getFingerPrintImage()
{
    Bitmap imageBuffer = new Bitmap(FP_SIRINA_SLIKE, FP_VISINA_SLIKE );
    for (int i = 0; i <= FP_SIRINA_SLIKE - 1; i++)
    {
        for (int j = 0; j <= FP_VISINA_SLIKE - 1; j++)
        {
            if (P[i, j] == 1)
            {
                imageBuffer.SetPixel(i, j, Color.Blue);
            }
            else
            {
                imageBuffer.SetPixel(i, j, Color.White);
            }
        }
    }
    return imageBuffer;
}

public Bitmap getDetaljiSlikeOtiskaPrsta()
{
    // posesavanje slike otiska
    Bitmap imageBuffer = new Bitmap(FP_SIRINA_SLIKE, FP_VISINA_SLIKE );
    for (int i = 0; i <= FP_SIRINA_SLIKE - 1; i++)
    {
        for (int j = 0; j <= FP_VISINA_SLIKE - 1; j++)
        {

```

```

        if (P[i,j] == 1)
            imageBuffer.SetPixel(i,j,Color.Blue);
        else
            imageBuffer.SetPixel(i,j,Color.White );
    }
}

// kreiranje karakteristika
double[] m_arr = this.getTemplejtOtiskaPrsta();
int linelength = 5;
// crtanje linija
Pen penBlue = new Pen(Color.Blue, 1);
Pen penRed = new Pen(Color.Red, 1);
Pen penGreen = new Pen(Color.Green, 1);
Pen penGray = new Pen(Color.Gray, 1);
Graphics gf = Graphics.FromImage(Image.FromHbitmap(m_ImageBuffer.GetHbitmap() ));
Graphics gf = Graphics.FromImage(imageBuffer);
gf.CompositingMode = CompositingMode.SourceOver;

for(int i=7 ; i<=m_arr[0]-1;i=i+6)
{
    if(m_arr[i+4]>1)
    {
        gf.DrawRectangle(penRed,(int)m_arr[i]+(int)m_arr[1]-
3,(int)m_arr[i+1]+(int)m_arr[2]-2,5,5);
    }
    else if(m_arr[i+4]==1)
    {
        gf.DrawEllipse(penGreen,(int)m_arr[i]+(int)m_arr[1]-
3,(int)m_arr[i+1]+(int)m_arr[2]-2,5,5);
    }

}

// crtanje pocetnih tacaka
gf.DrawLine(penGray,(int)m_arr[1]-5,(int)m_arr[2],(int)m_arr[1]+5,(int)m_arr[2]);
gf.DrawLine(penGray,(int)m_arr[1],(int)m_arr[2]-5,(int)m_arr[1],(int)m_arr[2]+5);
return imageBuffer;
}

public void ObradaPapila()
{
    int change = 1;
    bool mbool = true;
    while (change != 0)
    {
        change = 0;
        for(int i = 2; i <= FP_SIRINA_SLIKE - 2;i++)
        {
            for(int j = 2; j <= FP_VISINA_SLIKE - 2;j++)
            {
                if (P[i,j] == 1)
                {
                    short c = 0;
                    // izracunavanje okolnih tacaka
                    if (P[i,j+1] == 1) { c++;}
                    if (P[i+1,j+1] == 1) { c++;}
                    if (P[i+1,j] == 1) { c++;}
                    if (P[i+1,j-1] == 1) { c++;}
                    if (P[i,j-1] == 1) { c++;}
                    if (P[i-1,j-1] == 1) { c++;}
                    if (P[i-1,j] == 1) { c++;}
                    if (P[i-1,j+1] == 1) { c++;}

                    if ((c >= 2) && (c <= 6 ))
                    {
                        c = 0;
                        // izracunavanja u smeru kazaljke na satu od 2 do 9
                        if ((P[i-1,j+1] == 0) && (P[i,j+1] == 1)) { c++;}
                        if ((P[i,j+1] == 0) && (P[i+1,j+1] == 1)) { c++;}
                        if ((P[i+1,j+1] == 0) && (P[i+1,j] == 1)) { c++;}

```

```

        if ((P[i+1,j] == 0) && (P[i+1,j-1] == 1)) { c++;}
        if ((P[i + 1,j-1] == 0) && (P[i,j-1] == 1)) { c++;}
        if ((P[i,j-1] == 0) && (P[i-1,j-1] == 1)) { c++;}
        if ((P[i-1,j-1] == 0) && (P[i-1,j] == 1)) { c++;}
        if ((P[i - 1,j] == 0) && (P[i-1,j+1] == 1)) { c++;}

        if (c == 1 )
        {
            c = 0;
            if (mbool == true)
            {
                if ((P[i,j+1] * P[i+1,j] * P[i+1,j-1])
                {
                    if ((P[i+1,j] * P[i+1,j-1] *
                    {
                        P[i,j] = 0;
                        change++;
                    }
                }
                mbool = false;
            }
            else
            {
                if ((P[i,j+1] * P[i+1,j-1] * P[i-1,j])
                {
                    if ((P[i,j+1] * P[i+1,j] * P[i-
                    {
                        P[i,j] = 0;
                        change++;
                    }
                }
                mbool = true;
            }
        }
    }
}
}
}
}
}
}
}
}
}

public void ProveraObradaPapila()
{
    int c = 1;
    while (c != 0)
    {
        c = 0;
        for(int i = 1;i<=FP_SIRINA_SLIKE - 1;i++)
        {
            for(int j = 1;j<=FP_VISINA_SLIKE - 1;j++)
            {
                if ((P[i,j] == 1) && (i != 0) && (j != FP_VISINA_SLIKE - 1) &&
                (j != 0) && (i != FP_SIRINA_SLIKE - 1))
                {
                    if ((P[i-1,j-1] == 1) && (P[i,j-1] == 1) && (P[i+1,j-1]
                    == 1) && (P[i-1,j+1] == 0) && (P[i,j+1] == 0) && (P[i+1,j+1] == 0))
                    {
                        P[i,j] = 0;
                        c++;
                    }
                    else if ((P[i-1,j+1] == 1) && (P[i,j+1] == 1) &&
                    (P[i+1,j+1] == 1) && (P[i-1,j-1] == 0) && (P[i,j-1] == 0) && (P[i+1,j-1] == 0))
                    {
                        P[i,j] = 0;
                        c++;
                    }
                    else if ((P[i-1,j] == 1) && (P[i-1,j - 1] == 1) && (P[i-

```

```

1,j+1] == 1) && (P[i+1,j] == 0) && (P[i+1,j+1] == 0) && (P[i+1,j-1] == 0))
{
    P[i,j] = 0;
    c++;
}
else if ((P[i+1,j] == 1) && (P[i+1,j-1] == 1) &&
(P[i+1,j+1] == 1) && (P[i-1,j] == 0) && (P[i-1,j+1] == 0) && (P[i-1,j-1] == 0) )
{
    P[i,j] = 0;
    c++;
}
else if ((P[i-1,j] == 1) && (P[i,j-1] == 1) && (P[i,j+1]
== 0) && (P[i+1,j+1] == 0) && (P[i + 1,j] == 0))
{
    P[i,j] = 0;
    c++;
}
else if ((P[i-1,j] == 1) && (P[i,j+1] == 1) && (P[i,j-1]
== 0) && (P[i+1,j-1] == 0) && (P[i+1,j] == 0))
{
    P[i,j] = 0;
    c++;
}
else if ((P[i,j+1] == 1) && (P[i+1,j] == 1) && (P[i-1,j]
== 0) && (P[i-1,j-1] == 0) && (P[i,j-1] == 0))
{
    P[i,j] = 0;
    c++;
}
else if ((P[i,j-1] == 1) && (P[i+1,j] == 1) && (P[i-1,j]
== 0) && (P[i-1,j+1] == 0) && (P[i,j+1] == 0) )
{
    P[i,j] = 0;
    c++;
}
}
}
}
}

public void AlgoritamUlanacavanja(int ChainLinkDistance)
{
    for(int i = 1;i<= FP_SIRINA_SLIKE - 1;i++)
    {
        for (int j = 1 ; j <= FP_VISINA_SLIKE - 1;j++)
        {
            if ((P[i,j] == 1) && (i != FP_SIRINA_SLIKE - 1) && (i != 0) && (j !=
FP_VISINA_SLIKE - 1) && (j != 0))
            {
                if (P[i + 1,j] == 0)
                {
                    short countX = 0;
                    while (((i + countX) <= FP_SIRINA_SLIKE - 1) && (countX
<= ChainLinkDistance))
                    {
                        if (((i + countX + 1) <= FP_SIRINA_SLIKE - 1) &&
((countX + 1) <= ChainLinkDistance))
                        {
                            if (P[i + countX + 1,j] == 0)
                            {
                                countX++;
                            }
                            else
                            {
                                break;
                            }
                        }
                    }
                    else
                    {
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    if ((countX != 0) && ((countX + 1) <= ChainLinkDistance))
    {
        for (int temp = 0; temp <= countX; temp++)
        {
            P[i+temp,j] = 1;
        }
    }
}

if ((P[i,j] == 1) && (i != FP_SIRINA_SLIKE - 1) && (i != 0) && (j !=
FP_VISINA_SLIKE - 1) && (j != 0))
{
    if (P[i,j+1] == 0 )
    {
        short countY = 0;
        while (((j + countY) <= FP_VISINA_SLIKE-1) && (countY <=
ChainLinkDistance))
        {
            if (((j + countY + 1) <= FP_VISINA_SLIKE-1) &&
((countY + 1) <= ChainLinkDistance) )
            {
                if (P[i,j + countY + 1] == 0)
                {
                    countY++;
                }
                else
                {
                    break;
                }
            }
            else
            {
                break;
            }
        }
        if ((countY != 0) && (countY + 1 <= ChainLinkDistance))
        {
            for(int temp = 0; temp <= countY; temp++)
            {
                P[i,j + temp] = 1;
            }
        }
    }
}

if ((P[i,j] == 1) && (i != FP_SIRINA_SLIKE - 1) && (i != 0) && (j !=
FP_VISINA_SLIKE - 1) && (j != 0))
{
    if (P[i + 1,j + 1] == 0)
    {
        short countYX = 0;

        while ((j + countYX <= FP_VISINA_SLIKE-1) && (i + countYX
<= FP_SIRINA_SLIKE-1) && (countYX <= ChainLinkDistance))
        {
            if (((j + countYX + 1) <= FP_VISINA_SLIKE-1) && ((i
+ countYX + 1) <= FP_SIRINA_SLIKE-1) && ((countYX + 1) <= ChainLinkDistance))
            if (P[i+countYX+1,j + countYX + 1] == 0)
            {
                countYX++;
            }
            else
            {
                break;
            }
        }
        else
    }
}

```

```

        {
            break;
        }
    }

    if ((countYX != 0) && (countYX + 1 <= ChainLinkDistance))
    {
        for(int temp = 0 ; temp <= countYX ; temp++)
        {
            P[i+temp,j+temp] = 1;
        }
    }
}

}

}

}

private Point getIzvorniOtkisak()
{
    Point m_Point = new Point();
    double gradcur=0;
    double gradprev=0;
    double gradchangebig=0;
    double gradchange=0;

    double graddistancebig=0;
    double graddistance=0;

    double prevx=0;
    double prevy=0;

    for(int j =50;j<=FP_VISINA_SLIKE-50;j++)
    {
        for(int i =50;i<=FP_SIRINA_SLIKE-50;i++)
        {
            if(P[i,j]==1)
            {
                // izracunavanje okolnih piksela
                int tc=0;
                int x1=0;
                int y1=0;
                int x2=0;
                int y2=0;

                for (int m = -
1*FP_RADIJUS_TEMPLEJTA;m<=FP_RADIJUS_TEMPLEJTA;m++)
                {
                    for (int n = -
1*FP_RADIJUS_TEMPLEJTA;n<=FP_RADIJUS_TEMPLEJTA;n++)
                    {
                        if ((m==FP_RADIJUS_TEMPLEJTA)|| (m==(-
1)*FP_RADIJUS_TEMPLEJTA)|| (n==FP_RADIJUS_TEMPLEJTA)|| (n==(-1)*FP_RADIJUS_TEMPLEJTA))
                        {
                            if(P[i+m,j+n] == 1 )
                            {
                                tc++;
                                if (tc==1)
                                {
                                    x1=i+m;
                                    y1=j+n;
                                }
                                if (tc==2)
                                {
                                    x2=i+m;
                                    y2=j+n;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    if(tc==2)
    {
        if ((x2-x1)>0)
        {
            gradcur = (y2-y1)/(x2-x1);

            if((gradcur>0)&&(gradprev<0))
            {
                gradchange = Math.Abs(gradcur) +
                graddistance = Math.Abs(i) -
                if(gradchangebig<gradchange)
                {
                    if (graddistancebig<graddistance)
                    {
                        gradchangebig=gradchange;
                        graddistancebig=graddistance;
                        m_Point.X=i;
                        m_Point.Y=j;
                    }
                }
                break;
            }
            // resetovanje promenljivih
            gradprev=gradcur;
            gradcur=0;
            prevx=i;
            prevy=j;
        }
    }
}

return m_Point;
}

private int getKlasifikacijaOtiskaPrstan()
{
    Point m_Point = this.getIzvorniOtisak();
    double gradcur=0;

    double gradlt=0;
    double gradrt=0;
    double gradlb=0;
    double gradrb=0;

    double cgradlt=0;
    double cgradrt=0;
    double cgradlb=0;
    double cgradrb=0;

    for(int j =50;j<=FP_VISINA_SLIKE-50;j++)
    {
        for(int i =50;i<=FP_SIRINA_SLIKE-50;i++)
        {
            if(P[i,j]==1)
            {
                // izracunavanje okolnih piksela
                int tc=0;
                int x1=0;
                int y1=0;
                int x2=0;
                int y2=0;

                for (int m = -
1*FP_RADIJUS_TEMPLEJTA;m<=FP_RADIJUS_TEMPLEJTA;m++)
                {

```

```

for (int n = -
1*FP_RADIJUS_TEMPLEJTA; n<=FP_RADIJUS_TEMPLEJTA; n++)
{
    if ((m==FP_RADIJUS_TEMPLEJTA) || (m==(-
1)*FP_RADIJUS_TEMPLEJTA) || (n==FP_RADIJUS_TEMPLEJTA) || (n==(-1)*FP_RADIJUS_TEMPLEJTA))
    {
        if (P[i+m,j+n] == 1 )
        {
            tc++;
            if (tc==1)
            {
                x1=i+m;
                y1=j+n;
            }
            if (tc==2)
            {
                x2=i+m;
                y2=j+n;
            }
        }
    }
}

if(tc==2)
{
    if ((x2-x1)>0)
    {
        gradcur = (y2-y1)/(x2-x1);

        if ((x2<m_Point.X)&&(y2>m_Point.Y))
        {
            gradlt=gradlt+gradcur;
            gradlt++;
        }
        else if ((x2>m_Point.X)&&(y2>m_Point.Y))
        {
            gradrt=gradrt+gradcur;
            gradrt++;
        }
        else if ((x2<m_Point.X)&&(y2<m_Point.Y))
        {
            gradlb=gradlb+gradcur;
            gradlb++;
        }
        else if ((x2>m_Point.X)&&(y2<m_Point.Y))
        {
            gradrb=gradrb+gradcur;
            gradrb++;
        }
    }
}

}

}

// prosečna stopa za 4 ugla
gradlb=gradlb/cgradlb;
gradrb=gradrb/cgradrb;
gradlt=gradlt/cgradlt;
gradrt=gradrt/cgradrt;

if ((gradlt>0)&&(gradrt>0)&&(gradlb>0)&&(gradrb>0))
{
    return FP_VRTLOG;
}
else if ((gradlt>0)&&(gradrt>0)&&(gradlb>0)&&(gradrb>0))
{
    return FP_LEVA_PETLJA;
}
else if ((gradlt>0)&&(gradrt>0)&&(gradlb>0)&&(gradrb>0))

```

```

        {
            return FP_DESNA_PETLJA;
        }
        else if ((gradlt>0)&&(gradrt>0)&&(gradlb>0)&&(gradrb>0))
        {
            return FP_LUK;
        }
        else if ((gradlt>0)&&(gradrt>0)&&(gradlb>0)&&(gradrb>0))
        {
            return FP_LUK_SA_KROVOM;
        }
        else
        {
            return 1;
        }
    }
}

/// <summery>
/// Kreiranje templejta
/// </summery>

public double[] getTemplejtOtiskaPrsta()
{
    double x=0;
    double y=0;
    double r=0;
    double d=0;
    double[] m_arr = new double[FP_MAX_VELICINA_TEMPLEJTA];

    this.ObradaPapila();
    this.ProveraObradaPapila();
    this.ObradaPapila();
    this.ProveraObradaPapila();

    Point origin = this.getIzvorniOtisak();
    m_arr[1]=origin.X;
    m_arr[2]=origin.Y;

    int c = 7 ;
    int previ=0;
    int prevj=0;

    bool first = true;

    for(int j = 5 ;j<= FP_VISINA_SLIKE - 6;j++)
    {
        first = true;
        for( int i =5 ;i<= FP_SIRINA_SLIKE - 6;i++)
        {
            if ((c<FP_MAX_VELICINA_TEMPLEJTA)&&(P[i,j] == 1) && (i !=
FP_SIRINA_SLIKE - 1) && (i != 0) && (j != FP_VISINA_SLIKE - 1) && (j != 0) )
            {
                if(first == true)
                {
                    first = false;

                    if( (c>7)&&((m_arr[c-6]+origin.X)==previ)&&((m_arr[c-
5]+origin.Y)==prevj) )
                    {
                        m_arr[c--]=0;
                        m_arr[c--]=0;
                        m_arr[c--]=0;
                        m_arr[c--]=0;
                        m_arr[c--]=0;
                        m_arr[c--]=0;
                    }
                }
            }
            else

```

```

{
    int tc = 0;
    for (int m = -
1*FP_RADIJUS_TEMPLEJTA; m<=FP_RADIJUS_TEMPLEJTA; m++)
    {
        for (int n = -
1*FP_RADIJUS_TEMPLEJTA; n<=FP_RADIJUS_TEMPLEJTA; n++)
        {
            if ((m==FP_RADIJUS_TEMPLEJTA) || (m==(-
1)*FP_RADIJUS_TEMPLEJTA) || (n==FP_RADIJUS_TEMPLEJTA) || (n==(-1)*FP_RADIJUS_TEMPLEJTA))
            {
                if (P[i+m, j+n] == 1 )
                {
                    tc++;
                }
            }
        }
    }

    // izracunavanje parametara za keiranje templejta
    if ((tc==1) || (tc==3))
    {
        x = i- origin.X;
        y = j- origin.Y;
        r = Math.Sqrt(x*x + y*y);
        if ((x>0)&&(y>0))
        {
            d =Math.Atan(y/x);
        }
        else if ((x<0)&&(y>0))
        {
            d =Math.Atan(y/x) - Math.PI ;
        }
        else if ((x<0)&&(y<0))
        {
            d =Math.PI + Math.Atan(y/x);
        }
        else if ((x>0)&&(y<0))
        {
            d =2*Math.PI + Math.Atan(y/x);
        }
    }

    // provera uocениh minucija
    bool foundx = false;
    bool foundy = false;
    for (int m=7; m<=c; m=m+6)
    {
        if(m_arr[m+4]==3)
        {
            if(Math.Abs(Math.Abs((int)m_arr[m])-
Math.Abs(x))<4)
            {
                foundx=true;
            }
            if(Math.Abs(Math.Abs((int)m_arr[m+1])-
Math.Abs(y))<4)
            {
                foundy=true;
            }
        }
    }

    if ((tc==1) && (c <= FP_MAX_VELICINA_TEMPLEJTA-6) &&
(x!=0) && (y!=0) && ((foundx==false) || (foundy==false)) )
    {
        if (P[i-1, j+1] == 1)
        {
            m_arr[c++] = x;

```

```

        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 135 ;
    }
    else if (P[i,j+1] == 1)
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 90 ;
    }
    else if (P[i+1,j+1] == 1)
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 45 ;
    }
    else if (P[i+1,j] == 1)
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 0 ;
    }
    else if (P[i+1,j-1] == 1)
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 315 ;
    }
    else if (P[i,j-1] == 1)
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 270 ;
    }
    else if (P[i-1,j-1] == 1)
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 225 ;
    }
    else if (P[i-1, j] == 1 )
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 1 ;
        m_arr[c++] = 180 ;
    }
}
else if ((tc>=3)&&(c <= FP_MAX_VELICINA_TEMPLEJTA -

```

```

6)&&(x!=0)&&(y!=0)&&((foundx==false)|| (foundy==false)) )
    {
        m_arr[c++] = x;
        m_arr[c++] = y;
        m_arr[c++] = r;
        m_arr[c++] = d;
        m_arr[c++] = 3;
        m_arr[c++] = 0;
    }
    }
    previ=i;
    prevj=j;

    if (((i- origin.X)>=(306 - 4))&&((i- origin.Y)>=(269-4)))
    {
        if (((i- origin.X)<=(306+4))&&((i- origin.Y)<=(269+4)))
        {
            JOptionPane.showMessageDialog
(
null,Double.toString(c)+";"+Integer.toString(i)+";"+Integer.toString(j),"My
Point",JOptionPane.PLAIN_MESSAGE);
        }
    }
}

}

m_arr[0]=c;
return m_arr;
}

public String KonvertovanjeOtiskaPrstaUString(double[] finger)
{
    String temp="";
    for (int i=0;i<=finger.Length-1 ; i++)
    {
        temp = temp + finger[i].ToString() +";";
    }
    return temp;
}

public double [] KonvertovanjeTemplejtauDoble(String finger)
{
    double[] m_finger = new double[FP_MAX_VELICINA_TEMPLEJTA];
    int c=-1;
    String m_double = "";
    String temp = "";
    for (int i=0;i<=finger.Length-1 ; i++)
    {
        char[] tempch = finger.ToCharArray();
        temp = temp[i].ToString();
        if (temp=="")
        {
            m_finger[c++] =Double.Parse(m_double);
        }
        else
        {
            m_double = m_double + temp;
        }
    }
    return m_finger;
}

/// <summery>
/// Uporedjivanje
/// </summery>

// cross-corelation algoritam
public int Match(double[] finger1,double[] finger2 , int threshold , bool fastmatch)
{
    // uporedjivanje matrica

```

```

JOptionPane.showMessageDialog
(null, Double.toString(finger1[0])+";"+Double.toString(finger1[1])+";"+Double.toString(finger2[3]), "Mat
ch", JOptionPane.PLAIN_MESSAGE);
double matchcount = 0;
double matchcounttotal = (finger1[0]-6)/6;
double bestmatch = 0;
double radian = Math.PI/180;
bool foundpoint;

for (int k = -1 * FP_ROTACIJA; k <= FP_ROTACIJA; k++)
{
    for (int i = 7; i <= finger1[0]-5; i=i+6)
    {
        foundpoint = false;
        for (int j = 7; j <= finger2[0]-5; j=j+6)
        {
            if(foundpoint==false)
            {
                // uporedjivanje dve tacke
                int resx=0;
                int resy=0;
                double x1=0;
                double y1=0;
                double x2=0;
                double y2=0;
                double r=0;
                double d=0;

                r =finger2[j+2];
                d = finger2[j+3];
                x2=finger1[i];
                y2=finger1[i+1];

                x1 = r*Math.Cos(d+(k*radian));
                resx = Math.Abs((int)x2 + (int)(-1*x1));

                y1 = r*Math.Sin(d+(k*radian));
                resy = Math.Abs((int)y2 + (int)(-1*y1));

                if((FP_DOZVOLJENO_ODSTUPANJE > resx) &&
(FP_DOZVOLJENO_ODSTUPANJE > resy))
                {
                    if(finger1[i+4] == finger2[j+4])
                    {
                        if(((finger1[i+5]-
finger2[j+5])<=46)||((finger1[i+5]==0)&&(finger2[j+5]==315))||((finger1[i+5]==0)&&(finger2[j+5]==45)))
                        {
                            matchcount++;
                            foundpoint = true;
                            break;
                        }
                    }
                }
            }
        }
    }

    if (((matchcount/matchcounttotal)*100)>=threshold)&& (fastmatch == true))
    {
        // pronalazenje stepena poklapanja
        return (int)((matchcount/matchcounttotal)*100);
    }
    else
    {
        // nije pronadjeno poklapanje
        if(matchcount>bestmatch)
        {
            bestmatch = matchcount;
        }
        //resetovanje brojaca
    }
}

```

```
        matchcount=0;
    }
}
return (int)((bestmatch/matchcounttotal)*100);
}
} //end class
} //end namespace
```