Višefaktorska autentifikacija putem dijeljenja tajni

Sabolek, Iva

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

Diplomski studij informatike – nastavni smjer

IVA SABOLEK

VIŠEFAKTORSKA AUTENTIFIKACIJA PUTEM DIJELJENJA TAJNI

Diplomski rad

Pula, 2021.

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

Diplomski studij informatike – nastavni smjer

VIŠEFAKTORSKA AUTENTIFIKACIJA PUTEM DIJELJENJA TAJNI

Diplomski rad

JMBAG: 0303068581, redovna studentica

Studijski smjer: Informatika – nastavni smjer

Predmet: Kriptografija

Znanstveni područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Siniša Miličić

Pula, rujan 2021.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Iva Sabolek, kandidat za magistra edukacije informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljeni način, odnosno daje prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

_____

Student

U Puli, _____

IZJAVA O KORIŠTENJU
AUTORSKOG DJELA

Ja, Iva Sabolek dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom Višefaktorska autentifikacija putem dijeljenja tajni

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____

Potpis

_____

SADRŽAJ

1. SAŽETAK I KLJUČNE RIJEČI

SAŽETAK

Neka vrsta autentifikacija nas okružuje gdje god da se okrenemo, autentifikacija nam je potrebna na internetu, pametnim telefonima, bankomatima i mnogim drugim mjestima. No, kako bi se svaka osoba još bolje zaštitila od krađa identiteta može se koristiti i šifriranje različitih faktora autentifikacije. Dijeljenje tajni funkcionira na način da se svaki faktor autentifikacije šifrira na nekoliko određenih dionica koje se kasnije koriste za autentifikaciju te korištenjem njih nije potrebno koristiti prave podatke.

Ključne riječi: autentifikacija, jednostavna autentifikacija, dvofaktorska autentifikacija, višefaktorska autentifikacija, biometrijski faktori, lozinka, token, dijeljenje tajni, Shamirova shema dijeljenja tajni

ABSTRACT

Some kind of authentication surrounds us wherever we go, we need authentication on the internet, smartphones, ATMS and many other places. However, in order to protect better each person from identity theft we can use encryption of various authentication factors. Secret sharing works by encrypting each authentication factor into several specific shares that are later used for authentication, and using them does not require the use of real data.

Keywords: authentication, single-factor authentication, two-factor authentication, multi-factor authentication, biometrics factor, password, token, secret sharing, Shamir's secret sharing

2. UVOD

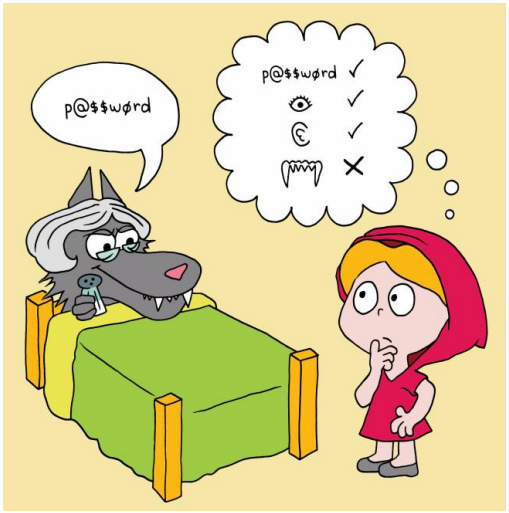
Autentifikacija ili autentikacija postupak je provjere nečijeg identiteta nekome, odnosno jesu li korisnici baš oni kojima se i predstavljaju. U realnom svijetu postupak provjere autentifikacije vrši se svakodnevno i to potpuno nesvjesno, ljudi se prepoznaju po izgledu, po načinu kako razgovaraju ili po boji glasa, isto tako, zna se dogoditi da se ljudi prepoznaju i po načinu kako netko zvuči dok hoda. Puno je lakše napraviti provjeru autentifikacije u realnom svijetu nego što je to unutar digitalnog svijeta, odnosno Internet mreže. Prije samo desetak godina, bilo je možda i nemoguće zamisliti da će današnje tehnologije imati toliko mogućnosti samog otključavanja mobitela, uz standardne pinove i lozinke, imamo oznake (engl. *patterns*), prepoznavanje lica, otiska prsta. Najvažnija autentifikacija reklo bi se da je ona na Internetu, kako u prošlosti tako i sada postoje česta probijanja sustava, krađe identiteta, osobnih podataka te je važno da se svaka osoba dobro zaštiti na internetu.

Obzirom na tehnologiju s kojom danas živimo, i na koji način se sve podaci mogu slati, velika sigurnost zaštite podataka bi trebala postojati. Dijeljenje tajni prvi put se spominje 1979. godine te označava metodu kojom se mogu zaštiti bilo koji vrlo osjetljivi podaci (oni mogu biti kriptografski ključevi, osobni podaci i slično). Ova metoda bavi se dijeljenjem tajne koju želimo zaštititi na više manjih dijelova, odnosno dionica koje djelitelj može poslati grupi ljudi kojima vjeruje. Odlična stvar samog dijeljenja tajne jest u tome što je svaka dionica sama za sebe beskorisna i tek kada se sve dionice skupe imat će smisla, u ovakvim slučajevima, neprijatelj ne može ništa ako je uspio probiti neku dionicu.

U prvom poglavlju opisuje se što je to autentifikacija te kojih to šest sigurnosnih ciljeva postoji. Nakon toga radi se usporedba autentifikacije i autorizacije. U drugom poglavlju spominju se tri vrste autentifikacije, jednostavna autentifikacija, dvofaktorska autentifikacija i višefaktorska autentifikacija te se sve tri pobliže objašnjavaju. U trećem poglavlju fokus je samo na višefaktorskoj autentifikaciji jer je ona i glavna tema ovog diplomskog rada, za nju se opisuju i izazovi glavnih operacija same višefaktorske autentifikacije. U četvrtom poglavlju nabrajaju se i opisuju koje sve vrste faktora autentifikacije postoje, pobliže se opisuju lozinka, token, prepoznavanje glasa, prepoznavanje lica, šarenica oka, otisak prsta te prepoznavanje vena. U petom

poglavlju opisuje se što je u kriptografiji dijeljenje tajni te koja su njena svojstva. U šestom poglavlju opisuju se pristupne strukture dijeljenja tajne te u sedmom poglavlju se opisuje Shamirova shema dijeljenja tajni kao jedan od mogućnosti na koji se način može dijeliti tajna. Unutar ovog poglavlja opisana su svojstva Shamirove sheme dijeljenja tajni, njeni nedostaci te primjer kako funkcionira Shamirova shema dijeljenja tajni. I na samom kraju nalazi se implementacija programa koji za prijavu korisnika koristi višefaktorsku autentifikaciju putem dijeljenja tajni.

3. AUTENTIFIKACIJA



Slika 1 Ilustrativni prikaz autentifikacije
Izvor: Bonneau et al, 2015: 79

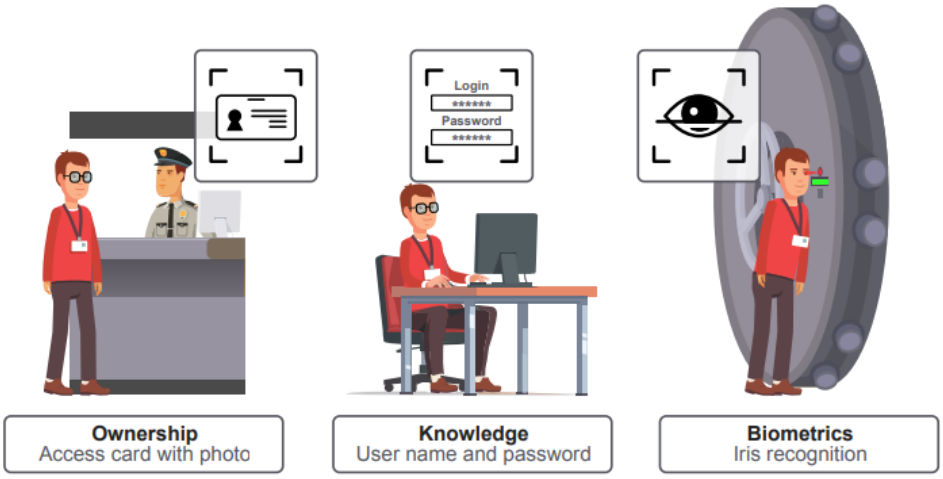
Kako bi sigurnost podataka bila što bolje očuvana, važna je autorizacija i provjera autentičnosti korisnika koji žele pristupiti privatnim i osjetljivim informacijama. Autentifikacija (engl. *autenthication*) postupak je koji potvrđuje identitet korisnika. Prvo, identificira se tko je korisnik, a drugo, pokušava se provjeriti je li korisnik taj za koga on tvrdi da jest (Mohsin, et al., 2017: 3). Potvrda nečijeg identiteta na internetu izuzetno je važna, zbog mnogih pokušaja prevara i krađa identiteta. Po Jainu (2017.) postoje dobri razlozi zašto je dobro koristiti autentifikaciju, odnosno raditi potvrdu identiteta:

- radi kontrole pristupa sustavu ili aplikaciji,
- radi povezivanja nekih privatnih podataka s pojedincem,
- radi uspostavljanja povjerenja između više strana kako bi se stvorila neka interakcija s njima,
- radi osiguravanja istinitosti podataka.

Još jedna definicija *online* autentifikacije glasi da je autentifikacija proces gdje korisnik identificira sebe slanjem x sustavu; sustav autentificira njegov identitet pomoću računanja $F(x)$ i provjerava je li to jednako spremljenoj vrijednosti y (Ometov, et al., 2018: 1). Što znači da prilikom unosa nekakve sigurnosne lozinke sustav provjerava je li baš ta lozinka vezana za nečije korisničko ime ili email, ukoliko jest, onda se

korisnika prebacuje u unutrašnjost same aplikacije. Na primjer, ukoliko se korisnik pokušava prijaviti na Gmail – mora upisati svoj email te svoju lozinku. Nakon što je korisnik provjeren i odobren, korisnika se automatski usmjerava na naslovnu stranu poruka. Autentifikacija predstavlja temeljni sigurnosni blok i čini sastavni dio kontrole pristupa sigurnosnoj politici (Mohsin, et al., 2017: 3). Drži korisnika odgovornim za svoje postupke prilikom pristupanja resursima, a istovremeno zadržava integritet i autentičnost podataka tijekom procesa komunikacije (Mohsin, et al., 2017: 3). U slučaju korisnika koji želi pristupiti određenim podacima, mora dokazati da je on baš taj za kojeg tvrdi da je.

Postoje tri skupine faktora prema kojima se mogu povezati osobe s utvrđenim vjerodajnicama: faktor znanja (engl. *knowledge factor*), faktor vlasništva (engl. *ownership factor*) i biometrijski faktor (engl. *biometric factor*).



Slika 2 Tri skupine faktora autentifikacije
Izvor: Ometov et al., 2018: 2

Na slici 2 možemo vidjeti da se faktor vlasništva (na slici prikazano kao engl. *Ownership*) odnosi na nešto što korisnik ima, kao što je to kartica sa osobnom slikom (osobna iskaznica, vozačka dozvola, iksica), mobitel, token. Faktor znanja (na slici prikazan kao engl. *Knowledge*) odnosi se na nešto što korisnik zna, kao na primjer lozinka, pin. I zadnji je biometrijski faktor (na slici prikazan kao engl. *Biometrics*) koji se odnosi na ono što korisnik posjeduje, odnosno ono što korisnik zapravo je, na primjer njegov otisak prsta, prepoznavanje lica, šarenice oka i slično.

3.1. Sigurnosni ciljevi

Kako se autentifikacija razvijala, usprkos tome, razvijala se naravno i sama sigurnost sustava, Benarous i drugi (2017.) tvrde da su na početku postojala samo tri načela sigurnosti: dostupnost (engl. *availability*), povjerljivost (engl. *the confidentiality*) i integritet (engl. *the integrity*), ali s vremenom su dodana još tri nova načela sigurnosti kao što su: autentičnost (engl. *authenticity*), nepobijanje (engl. *non-repudiation*) i auditivnost (engl. *auditability*). Benarous i drugi (2017.) su zatim opisali ovih spomenutih šest načela sigurnosti:

a) Dostupnost (engl. *availability*) – omogućuje ovlaštenim korisnicima pristup sustavu i potrebnim podacima za koje se prijavljuje. Jedna od najopasnijih prijetnji ovom načelu sigurnosti je Napad uskraćivanja usluge (engl. *Denial of Service attack*) koji onemogućava ovlaštenim korisnicima pristup sustavu (resursu) i tako ih sprječava u korištenju njihovih usluga (Benarous et al., 2017: 375).
b) Povjerljivost ili privatnost (engl. *the confidentiality*) – omogućuje da samo ovlaštena osoba može i smije pristupiti imovini, osobnim podacima ili resursima koje sustav ima. Jedna od glavnih prijetnji ovom načelu sigurnosti je izlaganje podataka (Benarous et al., 2017: 375).
c) Integritet (engl. *the integrity*) – omogućuje da samo ovlaštena osoba može promijeniti neke podatke te se tako dokazuje kako je sadržaj ostao autentičan te da ga nije promijenila treća strana već ovlaštena osoba.
d) Autentičnost (engl. *authenticity*) – omogućuje sustavu potvrđivanje identiteta pošiljatelja (korisnika) ako se već bio identificirao na određenom sustavu, a nakon toga se određuje je li mu pristup u sustav odobren ili ne.
e) Nepobijanje (engl. *non-repudiation*) – omogućava da korisnik ne može poreći da je poslao neku poruku. Ovo je načelo dodano zato što se zna dogoditi da korisnici poriču slanje poruke ako su pokrenute neke legalne provjere, posebno u slučajevima kada poruka sadrži ilegalni sadržaj, prijetnje ili u internetskoj trgovini korisnikovo poricanje kupnje (Benarous et al., 2017: 376).
f) Auditivnost (engl. *auditability*) – omogućuje sustavu da prati sve radnje koje su povezane s određenom imovinom. To znači da sustav bilježi sve važne događaje koji se događaju, kao što su slanje i primanje poruka, IP adrese ili bilo

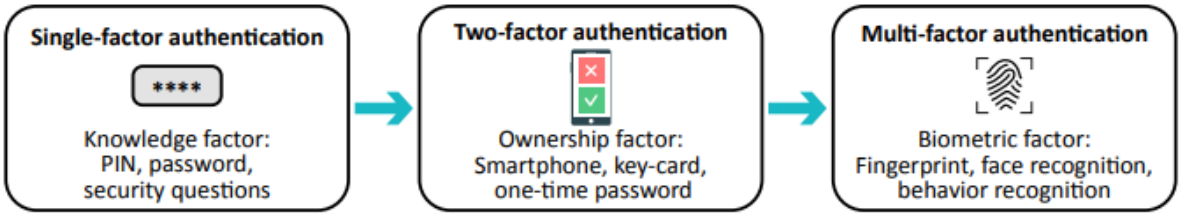
koje druge informacije koje bi pomogle u otkrivanju sigurnosnih problema i grešaka, praćenju uljeza i rješavanju i dokumentiranju problema (Benarous et al., 2017: 376).

3.2. Autentifikacija i autorizacija

Dok autentifikacija identificira tko je korisnik, autorizacija odobrava što taj korisnik smije napraviti. Autorizacija omogućuje nekakva dopuštenja korisniku koji se identificirao. Provodi se uporabom mehanizama kontrole pristupa odobravanjem ili odbijanjem pristupa resursu prema skupu kriterija (Benarous et al., 2017: 384). Na sljedećem primjeru može se jasnije objasniti autorizacija. Korisnici koji koriste online kupovinu, kada naprave svoj korisnički račun na nekoj internetskoj trgovini, imaju pristup postavkama računa, pojedinostima plaćanja, povijestima narudžbi i tako dalje. Dok administratori internetske trgovine imaju uz navedene mogućnosti i još neka druga dopuštenja, kao što su pregled robe, unos novih usluga, proizvoda, izmjena cjenika i drugo.

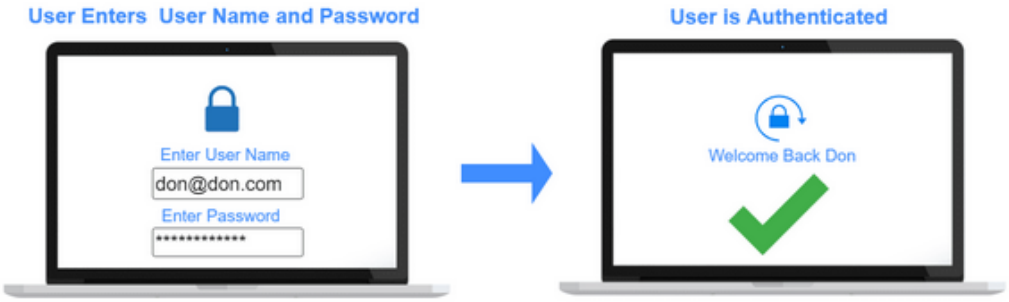
Slika 3 Autorizacija i autentifikacija
Izvor: https://www.ssl2buy.com/wiki/authentication-vs-authorization-whats-the-difference, zadnji pristup 18.05.2021

4. VRSTE AUTENTIFIKACIJA



Slika 4 Evolucija autentifikacije od jednostavne do višefaktorske
(Izvor: Ometov et al., 2018: 3)

4.1. Jednostavna autentifikacija



Slika 5 Prikaz jednostavne autentifikacije
Izvor: https://doubleoctopus.com/security-wiki/authentication/single-factor-authentication/, zadnji pristup 10.06.2021

Jednostavna autentifikacija (engl. *Basic authentication* ili *Single-factor authentication*, SFA) poznata je svim ljudima. Najjednostavnije bazirana na jednoj lozinki. Lozinka može biti bilo koja informacija koja se koristi za verificiranje identiteta osobe. Najčešći primjeri koji spadaju u takvu kategoriju su: uobičajena lozinka, imena domaćina (engl. *host*) i sustava, imena aplikacija, numerički ID (Jain, 2017: 385). Autentifikacija podrazumijeva provjeru valjanosti korištenjem jedne kombinacije za verificiranje - identitet korisnika i njegove lozinke. Proces provjere autentičnosti obično uspoređuje upisanu lozinku s onom koja je pohranjena u bazi podataka za provjeru autentičnosti. Ova se usporedba često vrši kao usporedba s običnim tekstom gdje se navedena

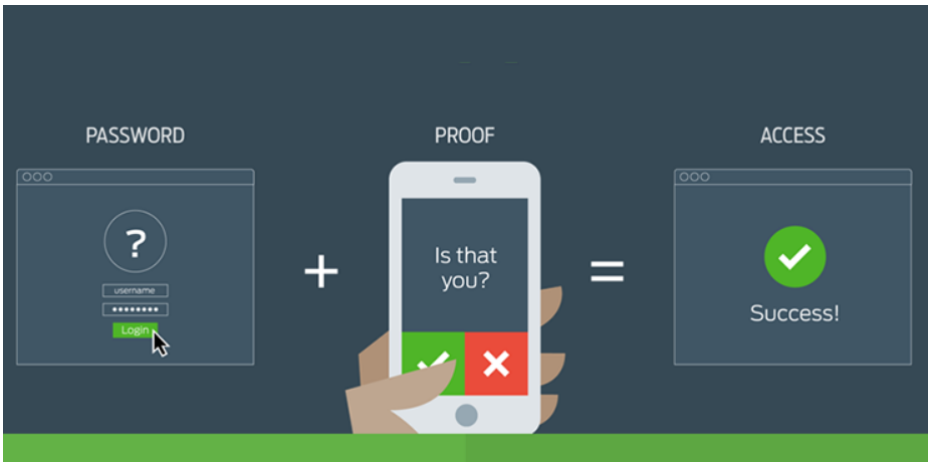
lozinka točno podudara s očekivanom lozinkom ili s nekom funkcijom permutacije kod koje lozinka prvo prolazi izmjenu poput šifriranja, a dobiveni podaci se zatim uspoređuju. Pohrana lozinke sljedeći je dio koji je također često u otvorenom tekstu ili se lozinka šifrira pa je u obliku šifrata.

Prednosti koje se dobivaju korištenjem jednostavne autentifikacije su:

- jednostavnost upravljanja unutar aplikacije,
- jednostavan za primjenu u svim aplikacijama,
- jednostavan za krajnje korisnike (Jain, 2017: 386).

Jednostavna autentifikacija nije previše sigurna te vrlo lako može biti ukradena zato što je tehnologija hakiranja postala sve više raznolika i naprednija te se sigurnost i provjera autentičnosti ne mogu pouzdati samo na autentifikaciju na temelju ID-a i lozinke (Kim i Hong, 2011: 187).

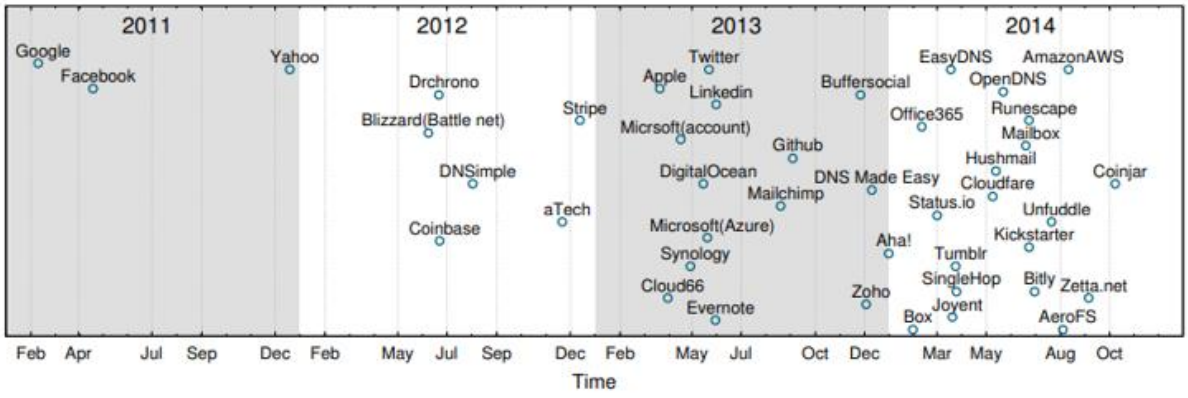
4.2. Dvofaktorska autentifikacija



Slika 6 Prikaz dvofaktorske autentifikacije
Izvor: https://www.dignited.com/30668/configure-two-factor-authentication-before-you-get-locked-out-of-your-own-account/, zadnji pristup 10.06.2021.

Dvofaktorska autentifikacija (engl. *Two-factor authentication,* 2FA) malo je složenija nego jednostavna autentifikacija jer koristi dvije kombinacije za verificiranje korisnikovog identiteta. Ona zahtjeva od korisnika čiji se identitet želi verificirati da priloži nešto što korisnik zna kao što su PIN ili lozinka i još nešto što korisnik posjeduje

kao na primjer bankovna kartica. Zato je najčešći primjer dvofaktorske autentifikacije automatizirani bankomat koji zahtjeva bankovnu karticu i osobni identifikacijski broj (PIN) koji služi kao lozinka. Prednost dvofaktorske autentifikacije naspram jednostavne autentifikacije jest u tome da ako napadač ukrade korisnikovu lozinku, on i dalje ne može pristupiti nekakvom računu, zato što mu treba i druga identifikacija koju ne može pridobiti. Današnji softveri najviše koriste mobilne telefone kao drugu komponentu autentifikacije, zbog toga što svi ljudi uvijek imaju mobilni telefon kod sebe. Dvofaktorska autentifikacija putem mobilnog telefona funkcionira na način da web aplikacija šalje jednokratno valjanu pristupnu šifru na mobitel korisnika (na primjer putem SMS-a, e-pošte ili namjenske aplikacije), koja se zatim mora unijeti zajedno s korisnikovim podacima kako bi se dovršila provjera autentičnosti (Konoth et al., 2016: 406). Iz tog razloga postoje brojne poznate tvrtke koje koriste mobilne telefone unutar dvofaktorske autentifikacije, jedne od njih su Facebook, Apple, Google, Microsoft, Dropbox, Github i mnogi drugi. Istraživanje koje su proveli Petsas i drugi (2015.) pokazalo je koliko je tvrtki počelo koristiti dvofaktorsku autentifikaciju na internetu. Na temelju rezultata, može se pretpostaviti da je brojka tvrtki koja koristi dvofaktorsku autentifikaciju sve brojnija i brojnija. Zbog toga što se može vidjeti kako 2011. godine samo troje pružatelja usluga koristi dvofaktorsku autentifikacije, a već 2014. godine broj pružatelja usluga koje koriste dvofaktorsku autentifikaciju skočio je čak na 19. Prema ovom pokazatelju, može se zaključiti kako danas u 2021. godini broj tvrtki koje koriste dvofaktorsku autentifikaciju puno veći.



Slika 7 Vremenski period početka korištenja dvofaktorske autentifikacije na web stranicama i web uslugama
Izvor: Petsas et al., 2015: 3

4.3. Višefaktorska autentifikacija

Slika 8 Prikaz višefaktorske autentifikacije
Izvor: https://www.securid.com/en-us/blog/the-language-of-cybersecurity/what-is-mfa, zadnji pristup 10.06.2021.

Višefaktorska autentifikacija (engl. *Multi-factor authentication*, MFA) koristi više kombinacija za verificiranje identiteta korisnika. Najčešće korišteni opis višefaktorske provjere autentičnosti jest upotreba podataka koji su poznati samo osobi, u kombinaciji s nečim u njezinom posjedu (Jain, 2017: 389-390). To su najčešće korisničko ime i lozinke ili token. Token je hardverska komponenta koja se koristi tijekom postupka provjere autentičnosti; obično pruža još jedan podatak koji se ne može utvrditi bez fizičke kontrole tokena (Jain, 2017: 390). Jain (2017.) isto tako navodi da postoje različite vrste tokena koji se koriste u višefaktorskoj autentifikaciji su:

- pametne kartice,
- jednokratna lozinka,
- PIN-ovi za jednokratnu upotrebu ili pseudo-slučajni brojevi,
- biometrijske informacije.

Višefaktorska autentifikacija pruža sljedeće dodatne prednosti:

- teško je prevariti i lažno se predstavljati,
- jednostavan za korištenje (Jain, 2017: 390).

Kako su sigurnosne komponente slojevite, složenost same autentifikacije također raste. Nedostaci koji se mogu dogoditi ukoliko se koriste tokeni za višefaktorsku autentifikaciju su ti da upravljanje tokenima može biti izazov, posebno u slučaju

izgubljenih ili ukradenih tokena (Jain, 2017: 390). Jedna od najčešće korištenih tokena današnjice su biometrijski faktori. Biometrijski faktor fizička je ili karakteristika ponašanja koja je jedinstvena za korisnika za provjeru autentičnosti (van Tilborg i Jajodia, 2011: 1287). U biometrijske faktore spadaju otisak prsta, šarenice oka, prepoznavanje lica, prepoznavanje glasa, geometrija ruku. Biometrijski faktori jednostavni su za korištenje jer već na pametnim telefonima postoji značajka otiska prsta ili prepoznavanje lica. Međutim, postoji i druga strana, a to je da pristup provjere autentičnosti uključuje visoke troškove postavljanja i održavanja. Također sustavi za provjeru autentičnosti koji koriste ovu vrstu mogu odbiti korisnika jer se nešto možda promijenilo. Iz tog razloga, puno puta se dogodi kako pametni telefon javi kako lice nije prepoznato ili otisak prsta nije točan iz razloga što se možda osoba našminkala pa više nema neke crte lica koje ima kada nije našminkana ili nije dobro prislonjen prst te se otisak nije mogao cijeli skenirati. Dakle, to znači da dva uzorka iste osobe nikada nisu potpuno ista. Ako su dva uzorka šifrirana iz sigurnosnih razloga, treba ih dešifrirati prije nego što se mogu podudarati (Bhargav-Spantzel et al., 2007: 530).

5. VIŠEFAKTORSKA AUTENTIFIKACIJA

Višefaktorska autentifikacija danas je važan faktor za mnoge provjere valjanosti kao što je:

- provjeravanje identiteta korisnika i elektroničkog uređaja (ili njegovog sustava),
- provjeravanje valjanosti infrastrukturne veze,
- provjeravanje valjanosti međusobno povezanih uređaja Internet stvari (engl. *Internet of Things*, IoT) (Ometov et al., 2018: 3), kao na primjer mobiteli, tableti ili neki drugi tokeni.

Postupak višefaktorske autentifikacije treba biti što jednostavniji kako se korisnici ne bi bunili te kako bi im bilo olakšano korištenje neke aplikacije. Ometov i drugi (2018.) objasnili su kako bi olakšana provjera identiteta korištenjem višefaktorske autentifikacije zapravo trebala izgledati ovako:

1. Korisnici se prvo registriraju i ovjeravaju kod pružatelja usluga kako bi aktivirali i upravljali uslugama kojima su spremni pristupiti;
2. Nakon pristupanja usluzi, korisnik mora proslijediti jednostavnu autentifikaciju s otiskom prsta koji je unaprijed potpisao pružatelj usluga;
3. Jednom kada ga sustav prihvati, korisnikov identitet se provjerava korisničkim imenom i lozinkom koje je morao podesiti na tom korisničkom portalu ili društvenoj prijavi;
4. Sekundarna provjera autentičnosti događa se automatski na temelju biometrijskog faktora, ili bi se tražio još dodatni kôd (Ometov et al., 2018: 4).

5.1. Izazovi glavnih operacija višefaktorske autentifikacije

5.1.1. Upotrebljivost

Upotrebljivost (engl. *usability*) odnosi se na to koliko je neki faktor autentifikacije upotrebljiv i treba li ga se koristiti. Ometov i drugi (2018.) prikazuju glavne izazove upotrebljivosti koji se mogu proučavati pomoću tri perspektive:

1. Efikasnost (učinkovitost) zadatka (engl. *task efficiency*) – odnosi se na vrijeme koje je potrebno za registraciju i autentifikaciju u samom sustavu (Ometov et al., 2018: 9);
2. Efektivnost (djelotvornost) zadatka (engl. *task effectiveness*) – odnosi se na broj prijavljenih pokušaja autentifikacije u sustavu (Ometov et al., 2018: 9);
3. Korisničke postavke (engl. *user preference*) – odnosi se na shemu provjere autentičnosti koju korisnik preferira u odnosu na neku drugu provjeru autentičnosti (Ometov et al., 2018: 9).

Stvar je u tome da se danas većina internetskih usluga provjere autentičnosti temelji na znanju, a to znači da se najviše koriste kombinacije korisničkog imena i lozinke. Složeniji sustavi zahtijevaju od korisnika interakciju s dodatnim tokenima (jednokratne lozinke, generatori koda, telefoni, itd.). Dopunjavajući tradicionalne strategije provjere autentičnosti, višefaktorska autentifikacija nije moguća bez provjere biometrijskih parametara. Poznato je kako današnji pametni telefoni sve više koriste prepoznavanje lica ili otiska prsta nego PIN-a ili uzorka.

5.1.2. Integracija

Drugi izazov višefaktorske autentifikacije jest u tome da se većina potrošačkih rješenja za višefaktorsku autentifikaciju i dalje temelji na hardveru. Općenito, integriranje fizičke i informatičke sigurnosti može uroditi značajnim prednostima za organizaciju, uključujući povećanu učinkovitost i usklađenost, plus poboljšanu sigurnost (Ometov et al., 2018: 10).

5.1.3. Sigurnost i privatnost

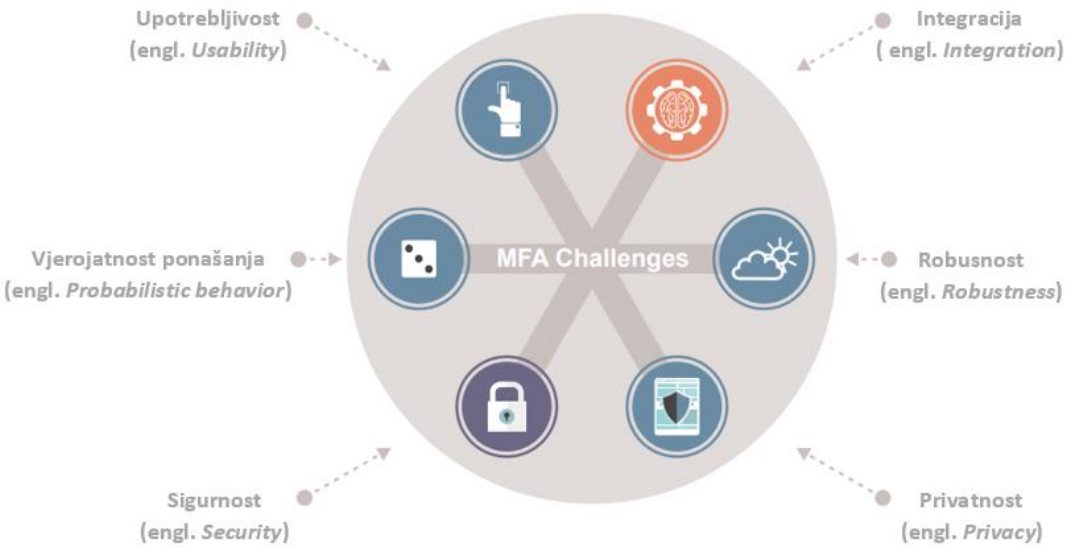
Bilo koji okvir višefaktorske autentifikacije digitalni je sustav koji se sastoji od kritičnih komponenti, poput senzora, uređaja za pohranu podataka, uređaja za obradu i komunikacijskih kanala (Ometov et al., 2018: 10). Svi faktori mogu se raniti raznim napadima na potpuno različitim razinama i to u rasponu od pokušaja ponavljanja do napada protivnika. Zato nam je sigurnost jako neophodna za omogućavanje i

održavanje privatnosti podataka te sigurnosti korisnika i aplikacija. Jedan od ključnih rizika napada na nečije podatke povezan je s lažnim predstavljanjem koje bi višefaktorska autentifikacija uspješno prihvatila. Značajno je da, zbog biometrije koju koriste različiti okviri višefaktorske autentifikacije, sjajna prilika za napadača je da analizira i tehnologiju koja se bazira na osnovi senzora i sam senzor rezultira otkrivanjem najprikladnijih lažnih materijala. (Ometov et al., 2018: 10). Glavni cilj sustava je zapravo osigurati sigurnosno okruženje ili da se unaprijed razmotri koje su to moguće prevare koje bi se mogle dogoditi. Sigurnosni okvir višefaktorske autentifikacije također bi trebao podržati panel za testiranje probijanja sustava kako bi se procijenile njegove potencijalne slabosti. U današnje vrijeme, programeri najčešće provode vanjsku auditivnost kako bi procijenili rizike i na temelju takve procjene djeluju za pažljivije planiranje (Ometov et al., 2018: 11).

5.1.4. Robusnost u radnom okruženju

Čak i ako su aspekti sigurnosti i privatnosti u potpunosti riješeni i zadovoljeni, biometrijski sustavi i dalje nisu baš ispunili zahtjev robusnosti. Razlog tom je uglavnom zbog operativnih ispitivanja koja su se provodila u laboratorijskom okruženju umjesto operativnih ispitivanja na terenu, odnosno u stvarnom okruženju. Jedan od takvih primjera je prepoznavanje glasa, koje je bilo vrlo pouzdano u tihoj sobi, ali nije uspjelo potvrditi korisnika u urbanim krajolicima (Ometov et al., 2018: 11). Biometrijske provjere autentičnosti temelje se na polju podudaranja uzoraka (Ometov et al., 2018: 11). Približno podudaranje važno je u bilo kojem sustavu višefaktorske autentifikacije, jer bi razlika između korisnika mogla biti presudna zbog različitih faktora i nesigurnosti, na koje sam korisnik ne može utjecati, a mogli su utjecati programeri koji su sastavljali i provodili ispitivanje biometrijskih faktora. Isto tako, kao što je već u radu spomenuto, ponekad se može dogoditi da otisak prsta nije prepoznat iako ga pravi korisnik daje. Zato što je slika snimljena tijekom skeniranja otiska prsta bila drugačija možda zbog kuta prsta, pritiska ili senzora koji ima neke smetnje. Postoje četiri važne stope pogrešaka koje se koriste kod biometrijskih sustava za provjeru autentičnosti, a one su FAR, FRR, EER i vrijeme provjere. FAR je stopa lažnog prihvaćanja (engl. *False Acceptance Rate*) koja prikazuje vjerojatnost da će sustav prihvatiti krivog korisnika (varalicu) kao pravog korisnika (Ometov et al., 2018: 11). FRR je stopa lažnog

odbijanja (engl. *False Rejection Rate*) koji prikazuje vjerojatnost da će sustav odbiti pravog korisnika jer će ga zamijeniti s varalicom, a koji se definira kao omjer broja lažnih odbijanja i ukupnog broja istinskih pokušaja podudaranja (Ometov et al., 2018: 11). EER je jednaka stopa pokušaja (engl. *Equal Error Rate*) koji prikazuje vjerojatnost pogreške kada su parametri sustava (poput praga odluke, engl. *the decision threshold*) postavljeni tako da su FRR i FAR jednaki (Jorgensen et al., 2011: 477). Što znači da je sustav precizniji što je EER niži. I na kraju postoji vrijeme provjere (engl. *Verification time*) koji prikazuje koje je vrijeme potrebno sustavu za prikupljanje dovoljno podataka o ponašanju za donošenje autentifikacije (Jorgensen et al., 2011: 477).



Slika 9 Prikaz izazova višefaktorske autentifikacije
Izvor: Ometov et al., 2018: 9

6. VRSTE FAKTORA ZA AUTENTIFIKACIJU

Tablica 1 Tipovi autentifikacije

Autentifikacija	Tipovi
Faktor znanja (Nešto što znaš?)	Lozinke, PIN
Faktor vlasništva (Nešto što imaš?)	Pametne kartice, tokeni, bankovna kartica
Biometrijski faktor (Nešto što jesi?)	Otisak prsta, geometrija ruke, prepoznavanje lica, glasa, šarenice oka

6.1. Lozinka

U svom radu Ometov i drugi (2018) prikazuju kako su PIN i lozinka najuobičajeniji načini autentifikacije te da je za takvu autentifikaciju potreban samo jednostavan uređaj u koji se unosi PIN ili lozinka. Ono što je već ranije spomenuto i vidljivo iz tablice 1, lozinka i PIN su primjeri faktora znanja, jer je to nešto što korisnik zna. Svakako, prednost korištenja sustava lozinki su niska cijena i jednostavnost implementacije nad određenim sustavima. Lozinka bi trebala biti tajna informacija koju zna samo ovlaštena osoba. No, ovdje se događa problem, a to je taj što je lako izgubiti kontrolu nad njima. Nažalost, ljudi nisu svjesni što sve može poći po krivu ako podijele s nekime svoju lozinku te se zbog toga događa da ljudi zapisuju svoje lozinke, šalju ih e-mailovima koje drugi ljudi mogu s lakoćom pročitati ili presresti u e-mailu. Lozinke se jako lagano mogu pogoditi, a kad se nešto od toga dogodi, lozinka više ne funkcionira kao token za provjeru autentičnosti jer nikada ne možete biti sigurni tko tu lozinku upisuje (Schneier, 2005: 136). Slabosti uključuju nebrojene načine presretanja i probijanja lozinki. Uz to, teško je otkriti kompromis sustava razbijanjem lozinke dok se ne napravi šteta. (van Tilborg i Jajodia, 2011: 1287).

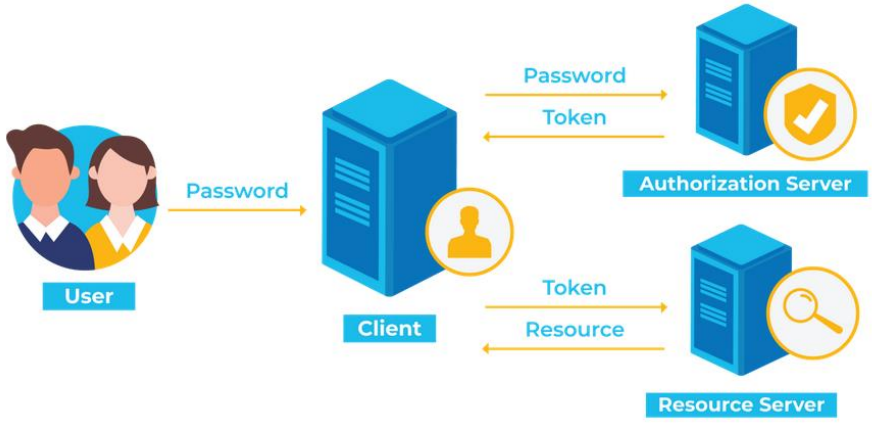
	Password Meter	How Secure Is My Password?	My1Login
Abc123456	Very Strong	Very Weak (4 days to crack)	Very Weak (0.12 seconds to crack)
Qwerty1234!	Very Strong	Good (400 years to crack)	Very Weak (0.01 seconds to crack)
Steven123!	Very Strong	Weak (6 years to crack)	Very Weak (0.04 seconds to crack)

Slika 10 Lozinka
Izvor: https://swoopnow.com/password-authentication/, zadnji pristup 15.07.2021.

Na slici 10 prikazano je koliko je neka od probnih lozinki jaka te koliko dugo je potrebno kako bi se odgonetnula, odnosno ukrala jedna lozinka. Na slici se vidi kako su sve lozinke prepoznate kao jake lozinke, ali već prvom prijavom u neki program, hakerima je potreban samo djelić sekunde kako bi ih odgonetnuli.

6.2. Token

Osim lozinke, u trećem poglavlju spomenut je i token koji se može koristiti za provjeru autentičnosti. Token je fizička komponenta koja na primjer može biti kartica koju korisnik posjeduje te token pripada faktoru vlasništva. Iz hardverske perspektive, korisnik može predstaviti pametnu karticu, mobilni telefon ili nosivi uređaj koje je teže prenijeti (Ometov, 2018: 5). Ako se to dogodi, sustav bi trebao imati sučelje koje bi omogućilo dvosmjernu komunikaciju između sustava i tokena. Osim nabrojanih komponenti tokena, najpoznatiji token je jednokratna lozinka (engl. *one time password*). No, problem koji predstavlja korištenje jednokratne lozinke jest nekontrolirano dupliciranje koje se može dogoditi. Prednosti same autentifikacije putem tokena jest u tome što je dosta njih u fizičkom obliku te se teže mogu izgubiti, a ako se to i dogodi možemo brzo intervenirati, isto tako korisniku je teško podijeliti token s drugim ljudima jer je u njegovom vlasništvu. Nedostaci korištenja tokena za autentifikaciju su veći troškovi postavljanja i održavanja takvih sustava, hardver je podložan kvaru ili se može izgubiti (van Tilborg i Jajodia, 2011: 1287).



Slika 11 Token
Izvor: https://www.okta.com/identity-101/what-is-token-based-authentication/, zadnji pristup 15.07.2021.

Na slici 11 prikazan je način na koji token funkcionira, recimo da se radi o kupnji autobusne karte preko interneta. Nakon unosa svih podataka o kartici korisnika koji želi kupiti kartu, internetska stranica obavještava korisnika da ne može procesuirati plaćanje jer je potreban unos mTokena. mToken u ovom slučaju je jednokratna lozinka koja se može dobiti korištenjem aplikacije MyWay, za korištenje aplikacije i dobivanje tokena potrebno je upisati PIN te nakon toga aplikacija daje token koji se može iskoristiti unutar 60 sekundi. Zatim dobiveni token upisuje se unutar polja na internetskoj stranici za kupnju te nakon toga događa se plaćanje i kupnja karte.

6.3. Prepoznavanje glasa

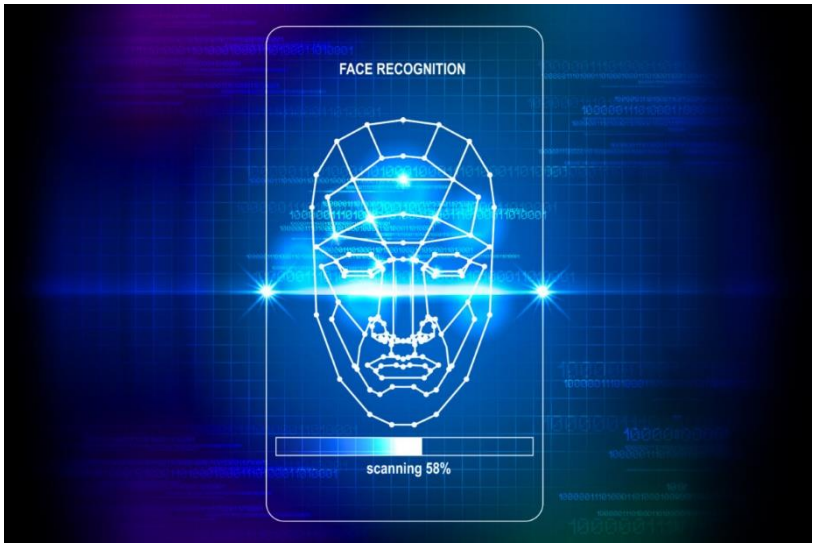
Prepoznavanje glasa davno je uveden pojam unutar elektroničkih uređaja, što zbog toga što postoji Siri na pametnim telefonima, što zbog toga što se koristi za višefaktorsku autentifikaciju. Prepoznavanje glasa prema tablici 1 pripada trećoj vrsti faktora, a ona jest biometrijski faktori, odnosno ono što čovjek jest. Ometov et al. (2018) navode kao jedan od glavnih i ozbiljnih nedostataka korištenja prepoznavanja glasa za autentičnost jest u tome što tehnologija sve više napreduje te se može dogoditi da specijalne agencije ne samo da budu mogle prepoznavati govornike, već će moći i oponašati njihov glas, intonaciju, ton i slično.

Slika 12 Prepoznavanje glasa
Izvor: https://speechpro-usa.com/product/voice_authentication/voicekey-webaccess, zadnji pristup 15.07.2021.

Na slici 12 prikazan je postupak provjere autentičnosti putem prepoznavanja glasa. Prijenosna računala, mobiteli imaju u sebi mikrofon te je vrlo lako raditi provjeru prepoznavanja glasa za neku internetsku stranicu, aplikaciju ili sustav.

6.4. Prepoznavanje lica

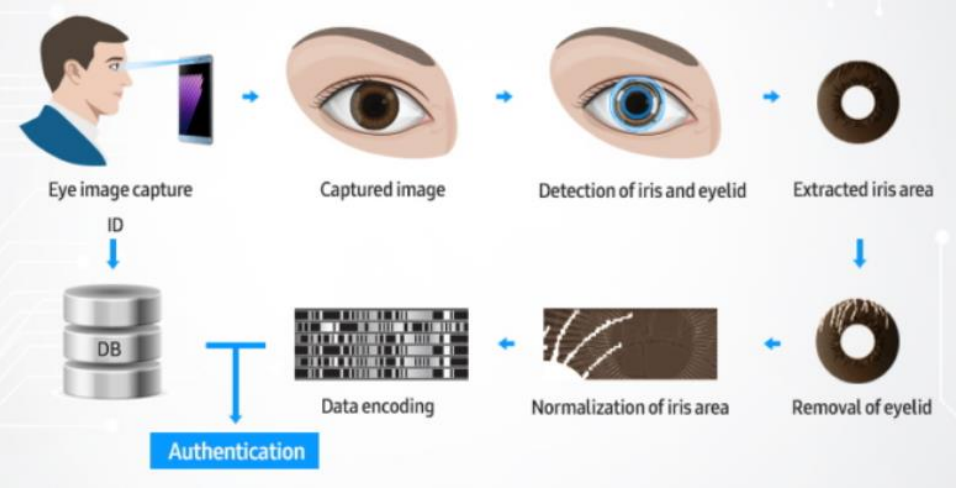
Na početku razvoja, tehnologija se temeljila na analizi orijentacijske slike, koju je bilo relativno jednostavno ponoviti isporukom sustava s fotografijom (Ometov, 2018: 5). Pošto to nije bila baš najbolja solucija za prepoznavanje lica, sljedeći korak dalje koji su znanstvenici napravili bilo je trodimenzionalno prepoznavanje lica. Trodimenzionalno prepoznavanje lica tražilo je od korisnika da tijekom postupka provjere autentičnosti zapravo pomiče glavu (lijevo-desno, gore-dolje ili u krug). Današnji pametni telefoni koriste baš takvu vrstu prepoznavanja lica. Pametni telefoni imaju mogućnost otključavanja licem, a prije toga potrebno je unutar postavku otključavanja telefona snimiti svoje lice tako što se popunjavaju dijelovi lica na samom zaslonu te se može vidjeti kako treba sve okrenuti glavu.



Slika 13 Prepoznavanje lica
Izvor: https://www.infoworld.com/article/3573069/what-is-face-recognition-ai-for-big-brother.html, zadnji pristup 15.07.2021.

6.5. Šarenica oka

Prepoznavanje šarenice oka na tržištu je dugi niz godina, što možemo vidjeti i po filmovima u kojima se traži da se ovlaštena osoba autentificira pomoću skeniranja šarenice oka. Osim šarenice oka, još jedna tehnika koja se može koristiti za provjeru autentičnosti je i analiza mrežnice. U tom slučaju se hvata i analizira tanko tkivo sastavljeno od živčanih stanica koje se nalaze u stražnjem dijelu oka (Ometov, 2018: 5). Pošto je struktura kapilara jako složena, mrežnica svake osobe je jedinstvena. Nedostatak tehnike prepoznavanja šarenice oka je u tome što je potreban visokokvalitetan uređaj za skeniranje i analizu slike.

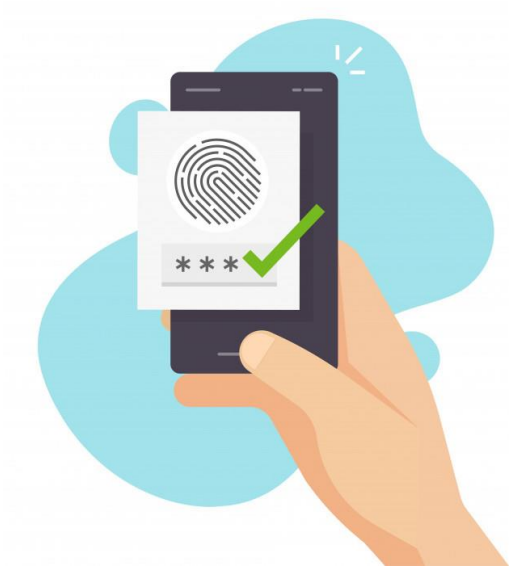


Slika 14 Prepoznavanje pomoću šarenice oka
Izvor: https://news.samsung.com/global/in-depth-look-keeping-an-eye-on-security-the-iris-scanner-of-the-galaxy-note7, zadnji pristup 15.07.2021.

Na slici 14 prikazan je postupak korištenja šarenice oka za autentifikaciju. Ono što možemo iščitati iz slike jest u tome da se skenira oko te se dobiva slika oka i iz nje se zatim detektira što je šarenica unutar njega. Zatim je fokus samo na šarenici oka iz koje se kasnije mogu pročitati podatci i koji se uspoređuju s podacima u bazi podataka.

6.6. Otisak prsta

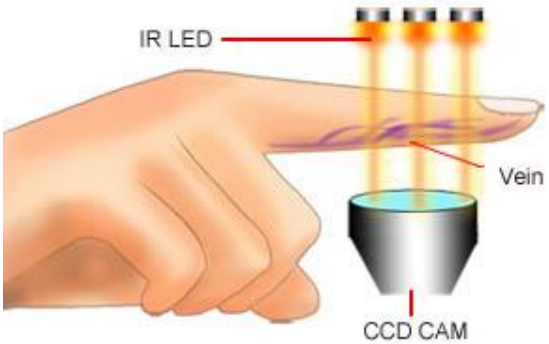
Tehnika prepoznavanja otiska prsta ili prstiju najkorištenija je tehnika za provjeru autentičnosti na pametnim telefonima. Danas, svaka druga osoba ima pametni telefon koji ima značajku korištenja otiska prsta za otključavanje – kao što je prikazano na slici 15. A radi na isti princip kao i prepoznavanje lica; unutar postavki otključavanja zaslona, korisnik mora otisnuti prst, ali isto tako ga malo i pomicati kako bi otisak prsta bio snimljen u potpunosti. Iako je jako dobar za korištenje, problem otiska prsta je taj što ga je jednostavno za iskopirati jer se naši otisci prstiju nalaze na svemu što dotaknemo. Iz tog razloga, ne preporučuje se da se koristi kao samostalni pristup provjere autentičnosti (Ometov et al., 2018: 6)



Slika 155 Otisak prsta
Izvor: https://www.freepik.com/premium-vector/fingerprint-security-identification-via-digital-biometric-sensor-online-mobile-phone-smartphone-finger-print-secure-authentication-authorization_7741085.htm, zadnji pristup 15.07.2021.

6.7. Prepoznavanje vena

Osim tehnike prepoznavanje otiska prsta, napredak u skenerima otiska prsta nudi priliku za prikupljanje i venske slike prsta (Ometov et al., 2018: 6). U trenutnoj fazi razvoja, tehnika prepoznavanja vena još uvijek je ranjiva na napadaje podvale (engl. *spoofing attacks*).



Slika 16 Prepoznavanje vena
Izvor: https://directory.ifsecglobal.com/rac82mfv-finger-vein-access-controller-prod1160626.html, zadnji pristup 15.07.2021.

Na slici 16 prikazana je tehnika prepoznavanja vena koja radi na principu CCD kamere preko koje korisnik stavlja svoj prst te kamera ima unutar sebe infracrveno led svijetlo koje prodire kroz kožu prsta te skenira vene.

Dolje u tablici 2 prikazana je usporedba glavnih pokazatelja za gore spomenute faktore autentifikacije. Prema istraživanju Ometova i drugih (2018.) prikazano je kako se faktori procjenjuju na temelju sljedećih parametara:

- univerzalnost (engl. *universality)* predstavlja prisutnost faktora u svakoj osobi,
- jedinstvenost (engl. *uniqueness*) pokazuje koliko dobro faktor razlikuje jednu osobu od druge,
- mjera naplativosti (engl. *collectability*) mjeri koliko je lako dobiti podatke za obradu,
- performanse (engl. *performance*) ukazuju na dostižnu točnost, brzinu i robusnost,
- prihvatljivost (engl. *acceptability*) označava stupanj prihvaćanja tehnologije od strane ljudi u njihovom svakodnevnom životu,
- lažno predstavljanje (engl. *spoofing*) označava razinu poteškoća u hvatanju i lažiranju uzorka.

U tablici „H“ označava *high*, odnosno visoku prikladnost faktora za višefaktorsku autentifikaciju, „M“ označava *medium*, odnosno srednju prikladnost, „L“ označava *low*, odnosno nisku prikladnost te „-„ označava da nije prikladno.

Tablica 2 Usporedba prikladnih faktora za višefaktorsku autentifikaciju
Izvor: Ometov et al., 2018: 8

Faktori	Univerzalnost	Jedinstvenost	Naplativost	Učinkovito	Prihvatljivost	Lažiranje
Lozinka	-	L	H	H	H	H
Token	-	M	H	H	H	H
Glas	M	L	M	L	H	H
Lice	H	L	M	L	H	M
Šarenica oka	H	H	M	M	L	H
Otisak prsta	M	H	M	H	M	H
Vena	M	M	M	M	M	M

Prema tablici 2 mogli bismo po svakom stupcu odrediti koji je najbolji faktor s obzirom na parametre prihvatljivosti. U stupcu Univerzalnosti može se primijetiti kako su najprikladniji faktori prepoznavanja lica i šarenice oka, dok kod jedinstvenosti faktor prepoznavanja lica je potpuno neprikladan, ali su zato prepoznavanje šarenice oka i otiska prsta najprikladniji. Kod parametra naplativosti, odnosno provjere koliko je lako dobiti nečije podatke, visoku prikladnost imaju faktori lozinke i tokena, vjerojatno jer se lako mogu probiti, kao što i prikazuje posljednji stupac parametra Lažiranja. Za najučinkovitije faktore izdvojeni su lozinka, token i otisak prsta jer se brzo mogu potvrditi. Brže je na pametnom telefonu upisati lozinku, otisnuti prst na senzor nego prepoznavanje glasa. Osim šarenice oka, koja još uvijek nije toliko popularna – svi ostali faktori su dobro prihvaćeni u svakodnevnom životu, najviše zahvaljujući pametnim telefonima. Iz ovoga svega može se zaključiti kako bi najbolje za višefaktorsku autentifikaciju bilo dobro koristiti lozinku ili token uz neke od ovih biometrijskih faktora kako bi svaki korisnik bio najbolje zaštićen.

7. DIJELJENJE TAJNI

Dijeljenje tajni (engl. *Secret sharing*) u kriptografiji označava način zaštite vrlo osjetljivih informacija poput privatnih kriptografskih ključeva ili biometrijskih podataka. Dijeljenje tajni djeluje na način da privatne podatke koje želimo zaštiti podijelimo na manje dijelove i zatim te podijeljene dijelove šaljemo unutar grupe ili mreže. Svaki taj podijeljeni dio, sam za sebe je beskoristan te ga nitko ne može iskoristiti, ali kada su svi ti dijelovi na jednom mjestu tada obnavljaju tu tajnu koju smo htjeli zaštititi. Ono označava da samo ovlašteni podskupovi koji su dobili dio tajne mogu rekonstruirati samu tajnu. Metoda dijeljenja tajni važan je alat u kriptografiji jer se može koristiti za izgradnju mnogih sigurnosnih protokola, kao što su na primjer opći protokol za višestranačko računanje, bizantski sporazum, *threshold* kriptografija (o kojoj će biti riječ u idućem poglavlju), kontrola pristupa i mnoge druge (Beimel, 2014: 2). Dijeljenje tajni je hijerarhijska shema distribucije, vlasnik koji želi tajno dijeliti tajnu ima mogućnost izbora kome će poslati dijelove informacija. No, naravno, postoji još i dodani sloj šifriranja kako bi se dodatno još osigurala privatnost i sigurnost.

Prema A beginner's guide to Shamir's Secret Sharing (2020) napravljen je primjer u kojem je tajna riječ KRIPTO. Tu tajnu riječ možemo podijeliti na šest dijelova i svaki dio sadrži po jedno slovo:

K_____, _R____, __I___, ___P__, ____T_, _____O,

ali, ako ju želimo dodatno osigurati, možemo je riječ šifrirati jednostavnom zamjenom slova tajne riječi u brojeve i to neka budu brojevi ASCII koda, svako slovo ima svoj broj unutar ASCII koda. Time bismo dobili da nam je K=75, R=82, I=73, P=80, T=84 i O=79, što bi vizualno prikazano izgledalo:

75____, _82____, __73___, ___80__, ____84_, _____79,

ovim načinom zaštitili smo privatni podatak od organiziranih napada, mogućnost napada na takve sustave je vrlo vjerojatno moguć, zato što sve osobe kojima smo poslali dijelove tajne mogu nas izdati i pokušati rekonstruirati tajnu riječ. U prvom slučaju bi otkrili koji je to tajni podatak, ali u drugom slučaju ne bi mogli odgonetnuti.

Najprikladnija vrsta podataka za algoritam dijeljenja tajni su podaci koji moraju biti apsolutno privatni, ali isto tako moraju biti sigurni i nikada ih se ne smije izgubiti. Cilj

ovakvog dijeljenja tajne je zapravo proširiti ključ s jednog mjesta na više, jer u slučaju da netko pokuša napasti sustav, prvo bi morali napasti sve uređaje na različitim lokacijama prije nego bi došli do tajnog ključa.

Dijeljenje tajni prvi put se spominje 1979. godine u radovima dvojice kriptografa koji su radili zasebno, oni su A. Shamir i G. R. Blakley. Obojica su objasnila svoju teoriju dijeljenja tajni te obojica rade na principu engl. *threshold* sheme, odnosno sheme praga (t, n). U takvoj shemi praga tajna K dijeli se na n dijelova među sudionicima, a t je prag koji je potreban za rekonstruiranje tajne. Tajnu odabire sudionik D koji se zove djelitelj (engl. *dealer*) (Stinson, 1999: 13).. Kada D želi podijeliti tajnu K s n sudionika, on onda svakome sudioniku daje djelomičnu informaciju koja se naziva dionica (engl. *share*) (Stinson, 1999: 13).. Naravno, svaka dionica bi se trebala podijeliti na siguran način, a pravilan siguran način je takav da niti jedan sudionik ne zna što su ostali sudionici dobili (Stinson, 1999: 13).

Sudionici mogu odgonetnuti tajnu K samo onda ako se udruže, njihovu skupinu nazvat ćemo B. U tom slučaju, ako je $|B| \geq t$, odnosno ako je broj sudionika veći od praga t koji je zadan za rekonstruiranje tajne K, onda oni mogu izračunati vrijednost K iz dionica koje kolektivno posjeduju (Stinson, 1999: 13).. U slučaju, ako je $|B| \leq t$, odnosno ako je broj sudionika koji su se udružili manji od broja praga t , oni neće moći izračunati vrijednost tajne K iz dionica jer ih nema dovoljan broj, ali isto tako neće moći doći ni do bilo kakvih informacija o tajni K (Stinson, 1999: 13).

7.1. Osnovni primjer tajnog dijeljenja

U djelu Stinsona (1999.) opisan je binarni niz duljine n, zato što je riječ o binarnom nizu imamo dvije dionice. Pretpostavimo da je djelitelj D odabrao tajnu $K = (k1, \dots, kn)$. D će konstruirati dvije dionice na sljedeći način:

- prva dionica odabrana je kao slučajni binarni niz duljine n, $s_1 = (x_1, \dots, x_n)$
- druga dionica konstruirana je na kao $s_2 = (y_1, \dots y_n)$

Obzirom da postoje samo dvije dionice s_1 i s_2, tajna K izračunava se uzimajući modulo 2 zbroja niza dionica s_1 i s_2:

$$K = (x_1 + x_2 \; mod \; 2, y_1 + y_2 \; mod \; 2)$$

Pretpostavimo da je:

- duljina ključa $n = 2$,
- prva dionica $s_1 = (0, 1)$,
- druga dionica $s_2 = (1, 1)$,

računanje tajne bi izgledalo:

$$K = (0 + 1 \; mod \; 2, 1 + 1 \; mod \; 2)$$

$$K = (1, 0),$$

tablično objašnjenje zašto je odgovor $(1, 0)$:

Tablica 3 XOR

A	***B***	***A+B***
0	0	***0***
1	0	***1***
0	1	***1***
1	1	***0***

Međutim, u slučaju da znamo samo dionicu $s_2 = (1, 1)$, onda imamo četiri moguća rješenja, odnosno četiri moguće vrijednosti tajne K:

- ako je $s_1 = (0, 0)$, onda je vrijednost $K = (1, 1)$,
- ako je $s_1 = (1, 0)$, onda je vrijednost $K = (0, 1)$,
- ako je $s_1 = (0, 1)$, onda je vrijednost $K = (1, 0)$,
- ako je $s_1 = (1, 1)$, onda je vrijednost $K = (0, 0)$.

7.2. Svojstva dijeljenja tajne

Iako se svaka metoda dijeljenja tajni razlikuje po načinu na koji se klasificira, sve metode i dalje imaju jednaka svojstva. Milutinović (2016.) u svome radu navodi dva osnovna svojstva dijeljenja tajni su:

1. Privatnost

2. Mogućnost otkrivanja tajne

Kod privatnosti, otkrivanje same tajne se uvijek mora zaštiti od neautoriziranog sudionika, a kod mogućnosti otkrivanja tajne, tajna je uvijek otkrivena uz pomoć pojedinih dijelova autoriziranog sudionika. To bi značilo da kod dijeljenja tajni, možemo imati više dionica koje smo podijelili, ali samo neke od tih dionica su nam potrebne za otkrivanje tajne. U poznatoj shemi praga (t, n), neprijatelj mora svakakvim pokušajima doći do točnog broja praga t, jer dok nema toliko dionica koje su mu potrebne, ne može nikako rekonstruirati tajnu. Naravno, to znači da neprijatelju treba jako puno vremena kako bi odgonetnuo tajnu. No, ono što bi se trebalo raditi s tajnom jest to da nakon nekog vremena se neke dionice promijene, a da tajna ostane nepromijenjena ili da se i cijela tajna promijeni – time dobivamo maksimalnu sigurnost. Ako mijenjamo dionice svaki put nakon određenog vremena, odnosno stvaramo nove dionice tajne, to neprijatelju skraćuje vrijeme djelovanja, zbog toga što može otkriti dionice koje su po novom zastarjele i neupotrebljive – što ga vraća na sam početak. Milutinović (2016.) navodi da ovo svojstvo stvaranja novih dijelova ili promijene tajne naziva se proaktivno svojstvo. Osim vanjskog neprijatelja, mogu postojati i „pouzdani“ neprijatelji, odnosno sudionici kojima je dodijeljen neka dionica tajne, ali žele zloupotrijebiti svoj položaj te i oni znaju raditi na prikupljanju ostalih dionica kako bi se sami uspjeli rekonstruirati tajnu. Zbog toga, postoji svojstvo kojom možemo provjeriti jesu li svi članovi bili pošteni, a to svojstvo naziva se provjerljivost (Milutinović, 2016: 5).

8. PRISTUPNE STRUKTURE

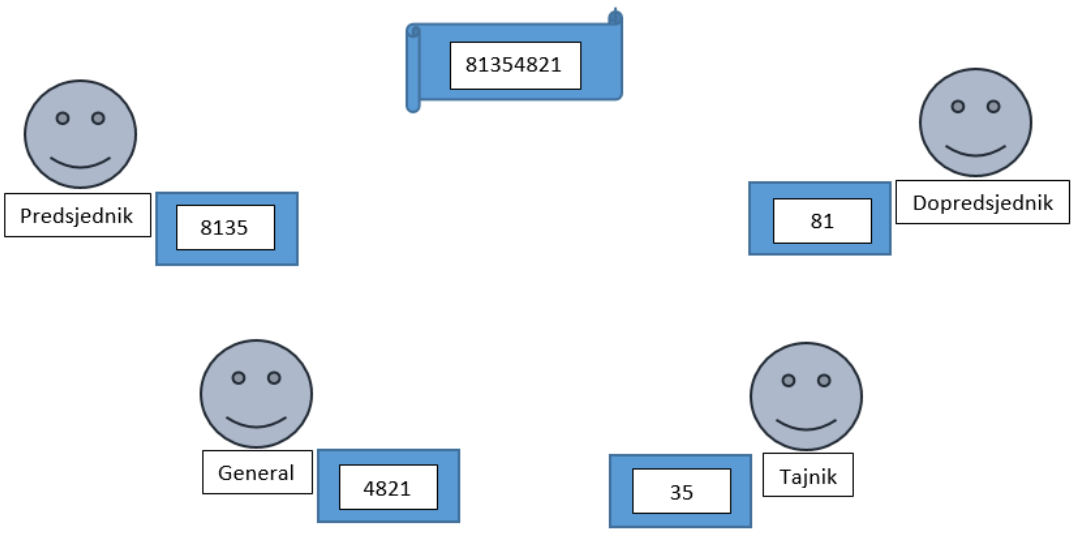
Postoji li tajna K koja se može podijeliti na set P od n sudionika. Naravno, samo određeni sudionici od ukupnog broja n sudionika bi mogli sastaviti tajnu K, ali to ne bi mogli ostali sudionici. Smart (2016.) navodi za primjer da je tajna K zapravo kod za nuklearno lansiranje i da postoje četiri sudionika (predsjednik, dopredsjednik, tajnik države i general u raketnom silosu) koji mogu pristupiti kodu, jer se glavne snage moraju složiti prije lansiranja, u ovom slučaju moraju se složiti predsjednik i general, ili ako nema predsjednika, onda glavne snage su dopredsjednik, tajnik i general. Onda bi ukupan broj sudionika n zapravo bio četiri, a sudionike bi označili s $P, D, T\ i\ G$ kao početna slova titula. Podskupovi sudionika koji mogu lansirati raketu su:

$$\{P, G\}\ i\ \{D, T, G\},$$

iz razloga koji je već ranije naveden da za lansiranje rakete su potrebni predsjednik i general ili dopredsjednik, tajnik i general.

8.1. Vizualni prikaz dijeljenja tajni

Za vizualni prikaz dijeljenja tajni, uzet ćemo da se tajni kod za lansiranje sastoji od znamenki 81354821 te se on dijeli na četiri dionice:



Slika 16 Vizualni prikaz tajnog dijeljenja

U vizualnom prikazu vidimo kako predsjednik i general imaju po četiri znamenke koda za lansiranje jer ako se treba lansirati onda su njih dvojica autorizirana za upis tajnog koda. Na drugoj strani vidimo kako tajnik i dopredsjednik imaju samo po dvije znamenke u svojim dionicama te se te znamenke podudaraju sa znamenkama predsjednikove dionice. U slučaju da nema predsjednika, tajnu mogu rekonstruirati ostalih troje sudionika.

Svaki sudionik dobija dionicu tajne K te ćemo dionice tajne označiti s $K_P, K_D, K_T \; i \; K_G$. Samo ako se odgovarajući sudionici sastanu i dobrim algoritmom dodaju svoj dio tajne, tek tada će se rekonstruirati tajna K, a ako se sastane nedovoljan broj autoriziranih sudionika, neće moći rekonstruirati tajnu K, odnosno neće moći izvući nikakvu informaciju. Svaka podskupina sudionika koja može oformiti tajnu naziva se kvalifikacijskim skupom (engl. *qualifying set*), a skup svih kvalifikacijskih skupova naziva se pristupna struktura (engl. *access structure*) (Smart, 2016: 403). Prema tome, podskupovi sudionika $\{P, G\}$ i $\{D, T, G\}$ su kvalifikacijski skupovi, ali isto tako drukčiji podskupovi sudionika kao $\{P, G, D\}, \{P, G, T\}$ i $\{P, D, G, T\}$ su isto kvalifikacijski skupovi. Razlog zašto osim prva dva glavna kvalifikacijska skupa postoje i ostala tri kvalifikacijska skupa jest u tome da se u tim skupovima nalaze već ovlaštene osobe te ako se preklapaju nečiji dijelovi tajne biti će zanemareni. Prema tomu imamo pet kvalifikacijskih skupova koje čine pristupnu strukturu, a ako se sastane bilo koji od ovih pet kvalifikacijskih skupova moći će oformiti tajnu K.

8.2. Monotone (jednolične) pristupne strukture

Monotona struktura je u Smartovom (2016.) primjeru skup P koji je kvalifikacijski skup te je prisutan u pristupnoj strukturi Γ tako da je:

$$P \epsilon \Gamma \quad ,$$

u kojem je već izjašnjeno da je P element iz pristupne strukture Γ. Prema tome možemo pretpostaviti:

$$\textit{ako } A \epsilon \Gamma, \text{ a } A \subset B \subset P, \text{ onda } B \epsilon \Gamma$$

što bi značilo da ako je skup A element od pristupne strukture Γ, a ako se B nalazi u lancu u kojem su već poznati elementi A i P, onda to znači da je i B element od pristupne strukture Γ.

Iz ovog primjera vidimo kako je u ovom slučaju pristupna struktura monotona, odnosno jednolična, zbog toga što će svojstvo svih mogućih shema tajnog dijeljenja podataka imati svi elementi pristupne strukture. U monotonoj strukturi vidimo da unutar pristupne strukture Γ svi kvalifikacijski skupovi dolaze unutar lanca $A \subset B \subset P$. Skup A koji je na početku lanca, naziva se minimalni kvalifikacijski skupovi kojeg ćemo označavati s $m(\Gamma)$.

8.2.1. Svojstva monotone pristupne strukture

Shemu dijeljenja tajni monotone pristupne strukture Γ čine algoritmi engl. *Share* i engl. *Recombine* u kojoj se nalaze strana P i njen prostor tajni T.

Share(s, Γ) uzima tajnu $s \epsilon T$ i monotonu strukturu pristupa Γ te određuje vrijednost s_A na način da je (Smart, 2016: 404):

$$s_A \; \forall \; A \epsilon P,$$

odnosno određuje se vrijednost s_A za svaki A koji je element od P, a vrijednost s_A naziva se *A*-jevom dionicom tajne.

Recombine(H) uzima skup H_0 dionice za neki podskup O od P (Smart, 2016: 404), odnosno:

$$H_O \; = \; \{s_0 : \; O \; \epsilon \; O\}.$$

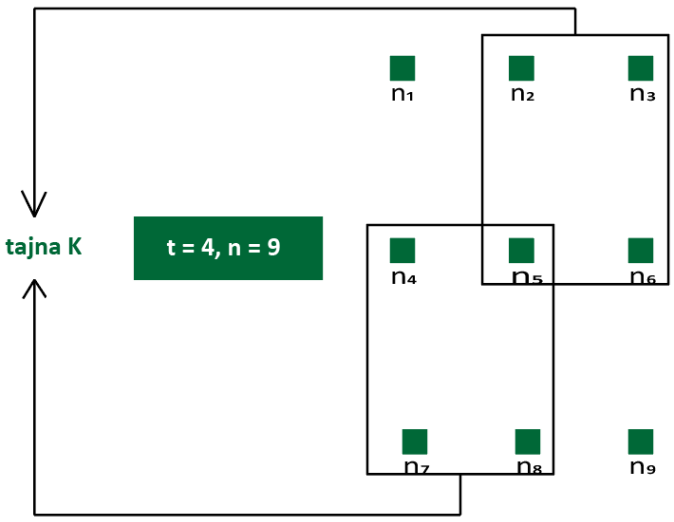
Ako je $O \epsilon \Gamma$ onda bi tajna trebala biti dobro oformljena, inače ne bi dobili ništa.

Shema dijeljenja tajni pristupnom strukturom smatra se sigurnom ako niti jedan beskrajno moćan protivnik ne može saznati ništa o temeljnoj tajni, a da nema pristup dionicama kvalificiranog skupa (Smart, 2016: 404).

9. SHAMIROVA SHEMA DIJELJENJA TAJNI

Shamirova shema dijeljenja tajni važan je kriptografski algoritam koji zapravo omogućuje sigurnosnu distribuciju privatnih informacija, pritom da ih se šalje nepouzdanom mrežom (A beginner's guide to Shamir's Secret Sharing, 2020). To je dakako tek jedna od mnogih kriptografskih tehnika koja osigurava da bilo kakvi osobni podaci ili podaci koji se ne smiju javno objaviti (biometrijski podaci, privatni ključevi i slično) čuvaju na sigurnom. Jedna od rizika dijeljenja tajni jest to da se dio tajne može izgubiti ili ugroziti. Osobe koje su ovlaštene za jedan dio tajne mogu to izgubiti, ukrasti, a može im se nešto i dogoditi.

Izraelski kriptograf Adi Shamir je 1979. godine prvi put objavio Shamirovu shemu dijeljenja tajni. Njegova shema bazira se na već poznatoj shemi praga (engl. *threshold*) (t, n) koja omogućava razbijanje tajnih informacija na n dionica, ali je samo jedan dio tih dionica potreban za rekonstrukciju izvorne tajne, te t označava prag dionica koje su potrebne. Prema primjeru, neka (t, n) bude $(4, 9)$ što znači da se tajna riječ ili informacija može podijeliti na devet dionica, ali samo su četiri dionice potrebne za rekonstrukciju tajne, ono što je važno napomenuti da nije važno koje četiri dionice. Taj minimum broja dionica koje su potrebne za rekonstrukciju, nazivaju se već spomenutim pragom. Ako ima manje dionica od praga, tada se tajna ne može rekonstruirati, a upravo to čini Shamirovo dijeljenje tajni sigurnim od napadača.



Slika 17 Vizualni prikaz Shamirove sheme tajnog dijeljenja

Prednosti Shamirove sheme dijeljenja tajni jest u tome što je algoritam fleksibilan i proširiv, odnosno, vlasnik tajne može dodati, izmijeniti ili ukloniti neke dijelove, a da pri tome ne modificira izvornu tajnu (A beginner's guide to Shamir's Secret Sharing, 2020). Ono što je iznimno važno, jest to da se šifrirane dionice tajne nikada ne otkrivaju, svaki ovlašteni vlasnik zna samo koja je njegova šifrirana tajna (ne odaje ju), a samo vlasnik tajne zna koje su sve šifrirane dionice i jedino on ima pristup cijelom skupu dionica tajne nakon što je tajna rekonstruirana.

9.1. Svojstva Shamirova dijeljenja tajni

Svojstva Shamirova dijeljenja tajni su:

- dinamičnost
    - tajni vlasnik može sigurno mijenjati pravila tajne, može dodati nove šifrirane dionice, ili može ukloniti već dodijeljene šifrirane dionice, ako nisu više potrebne (Understanding Shamir's Secret Sharing (SSS), 2018).,
- sigurnost
    - Shamirova shema je kriptoanalitički neraskidiva, zato što niti jedan ovlašteni vlasnik jedne dionice tajne ne može otkriti zajedničku tajnu bez pristupa pragovom broju tajnih dionica (Understanding Shamir's Secret Sharing (SSS), 2018).

Vrlo je važno imati pouzdane ljude kojima će se dati dionica tajne, zbog toga što postoji rizik da se više ljudi koji imaju dionice šifrirane tajne skupe i pređu prag koji je potreban za rekonstrukciju tajne te time bi dobili tajnu informaciju.

Polinomna interpolacija

Shamirova (t, n) shema praga radi na principu polinomne interpolacije. Interpolacija je metoda konstrukcije novih točaka podataka, a koristi polinom funkcije bilo kojeg stupnja. Polinomska funkcija ima oblik:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{k-1}$$ gdje je

$a_0 = K, a_0$ je slobodni koeficijent, odnosno naša tajna K,

$a_1, \dots, a_{t-1}$ su slučajno odabrani pozitivni cijeli brojevi,

9.2. Primjer Shamirove sheme dijeljenja tajni

Rafaloff (2019.) prikazuje primjer u kojem će tajna K biti 42 i koja na grafikonu može biti upisana kao $(0, 42)$, a prag neka bude 3 te će se tajna dijeliti na četiri dionice. Znači da je za rekonstrukciju tajne potrebno tri dionice od ukupno četiri. U ovom slučaju, javno je dostupan podatak praga 3 jer je potreban za rekonstrukciju tajne.

Funkcija:

$$f\,(x)\;=\;42\;+\;3x\;+\;5x^2,$$

gdje je $x = 0$ (znamo da je a_0 slobodni koeficijent i to je u ovom slučaju 42, a da bismo dobili 42, x mora biti 0). Slučajnim generiranjem radimo brojeve $(x_1, g_1), \dots, (x_t, g_t)$:

$$(18, 1716), (27, 3768), (31, 4940)\ i\ (35, 6272)$$

Rekonstrukcija tajne

$$P(x) = \sum_{j=1}^{t} L_j(x),$$

$$L_j(x) = \; g_j \prod_{\substack{n=1 \\ n=j}}^{t} \frac{x - x_n}{x_j - x_n}$$

gdje je prva formula polinoma $\sum$ za zbrajanje svih rezultata, a druga formula je Langrangeov interpolacijski polinom gdje $\prod$ znači množenje svih rezultata zajedno.

Obzirom da znamo da je prag $t = 3$, uvrštavanjem triju dionica u gornje formule dobivamo izvornu polinomsku funkciju:

$$P(x) = g_1\left(\frac{x - x_2}{x_1 - x_2} * \frac{x - x_3}{x_1 - x_3}\right) + g_2\left(\frac{x - x_1}{x_2 - x_1} * \frac{x - x_3}{x_2 - x_3}\right) + g_3\left(\frac{x - x_1}{x_3 - x_1} * \frac{x - x_2}{x_3 - x_2}\right)$$

$$(18, 1716), (27, 3768)\ i\ (31, 4940),$$

uvrštavanjem poznatih vrijednosti u funkciju dobijemo:

$$P(x) = 1716\left(\frac{x-27}{18-27} * \frac{x-31}{18-31}\right) + 3768\left(\frac{x-18}{27-18} * \frac{x-31}{27-31}\right) + 4940\left(\frac{x-18}{31-18} * \frac{x-27}{31-27}\right)$$

Koraci:

$$1716\left(\frac{x-27}{18-27} * \frac{x-31}{18-31}\right) = \frac{44(x-27)(x-31)}{3}$$

$$3768\left(\frac{x-18}{27-18} * \frac{x-31}{27-31}\right) = -\frac{314(x-18)(x-31)}{3}$$

$$4940\left(\frac{x-18}{31-18} * \frac{x-27}{31-27}\right) = 95(x-18)(x-27)$$

zbrajamo prva dva rješenja jer imaju razlomak te dobivamo:

$$\frac{44(x-27)(x-31)}{3} - \frac{314(x-18)(x-31)}{3} = 6(x-31)(-15x+248)$$

te ovaj rezultat bez razlomka zbrajamo s trećim rješenjem:

$$P(x) = 6(x-31)(-15x+248) + 95(x-18)(x-27)$$

$$P(x) = -90x^2 + 4278x - 46128 + 95x^2 - 4275x + 46170$$

dobijemo rezultat, našu već poznatu funkciju:

$$P(x) = 42 + 3x + 5x^2$$

te uvrštavanjem da je $x = 0$, dobijemo našu tajnu K:

$$P(0) = 42 + 3 * 0 + 5 * 0^2$$

$$P(0) = 42$$

9.3. Nedostaci Shamirove sheme dijeljenja tajni

Dva glavna problema Shamirove sheme dijeljenja tajni su:

- neprovjerljive dionice – sudionici koji su dobili dionicu tajne mogu prilikom rekonstrukcije tajne predati lažnu ili izmijenjenu dionicu i spriječiti rekonstrukciju same tajne,
- duljine dionica – sve dionice su uvijek iste duljine te zbog toga se može izračunati kolika je duljina same tajne.

10. IMPLEMENTACIJA

Spajanjem višefaktorske autentifikacije zajedno s dijeljenjem tajni, dobija se veća zaštita podataka korištenjem kriptografije. To znači da će bilo koji faktor autentifikacije, bio on faktor znanja, faktor vlasništva ili biometrijski faktor biti šifriran prema principu dijeljenja tajne. Iz ovog rada može se zaključiti da će svaki faktor biti podijeljen na nekoliko dionica, ovisno o Shamirovoj (t, n) shemi praga. Dobra stvar ovakve kombinacije je u tome što se prilikom autentifikacije korisnika u nekom programu neće morati upisivati prava tajna već dionice koje trećoj strani neće značiti ništa. U praktičnom primjeru koji slijedi, radi se o nekakvoj novoj vrsti programa koji bi zahtijevao da se prilikom prijave u sam program upisuje ID korisnika, njegova lozinka i PIN koji je sam korisnik odredio prilikom registracije. Od korisnika se traži da upiše dionice faktora i tek kada se sve dionice slažu i rekonstruiraju ID, lozinku i PIN, korisnik je potvrdio svoj identitet te može koristiti program.

```
1  import secrets
2  import random
3  from fractions import Fraction
```

Slika 18 Import modula

Na slici vidimo kako sam na početku napravila import secrets te import random. Modul tajni (engl. *secrets module*) koristi se za generiranje kriptografski jakih slučajnih brojeva prikladnih za upravljanje podacima kao što su lozinke, provjera autentičnosti računa, sigurnosni tokeni (secrets - Generate secure random numbers for managing secrets, 2021) i tajnih ključeva. Što znači da modul tajni za generiranje slučajnih brojeva ima pristup najsigurnijem izvoru slučajnosti koji neko računalo može pružiti (secrets - Generate secure random numbers for managing secrets, 2021). Te iz tog razloga ću koristiti funkcije iz modula tajni prilikom dijeljenja tajne na određene dionice. Funkcija koja se spominje kasnije u radu je secrets.randbits(k) koja vraća cijeli broj s *k* nasumičnih bitova, koliko je veliki k toliko će biti nasumičnih bitova. Modul slučajnosti (engl. *random module*) je manje sigurnija funkcija koja kao i modul tajni generira nasumično brojeve, ali je puno manje sigurnija za probijanje jer nije najbolje zaštićena.

Modul slučajnosti više je namijenjen za računalne igre i simulacije nego za kriptografsko šifriranje i dešifriranje. Znači ako uzmemo u slučaj te dvije funkcije, vidimo da je sigurnije i bolje odabrati neki nasumično veliki broj od *k* bitova nego samo nasumično odabrani broj. Zato ću unutar ovog primjera koristiti samo random funkciju kako bi se prilikom rekonstrukcije tajne nasumično odabrale već napravljene dionice tajne koje će biti uvrštene u već poznat Langrangeov algoritam za rekonstrukciju tajne. Zatim sam iz modula fractions dodala funkciju Fraction, modul fractions pruža podršku za racionalnu aritmetiku brojeva koja omogućuje stvaranje instance razlomka od cijelih brojeva (engl. *integer*), decimalnih brojeva (engl. *floats, decimal*) i nizova (engl. *strings*) (Fraction module in Python, 2020). Korištenjem funkcije Fraction omogućeno mi je da bilo koji tip može biti upisan, ali on će znati upravljati njime. Funkciju Fraction koristit ću prilikom rekonstrukcije tajne jer se prilikom rekonstrukcije treba dobiti cijeli broj pošto će i početne vrijednosti tajne biti numerički cijeli brojevi. Kada bismo koristili funkcije decimal onda bi morali prilikom rekonstrukcije pretvarati taj decimal u integer.

```
5   def koeficijent(t, K):
6       koef = []
7       for _ in range(t-1):
8           a = secrets.randbits(10)
9           koef.append(a)
10      koef.append(K)
11
12      return koef
```

Slika 19 Metoda koeficijent(t, K)

Na slici vidi se metoda koeficijent(t, K), unutar ove metode generira se koeficijent polinoma. Ova metoda prima parametre t i K, odnosno prag koji je potreban da bi se rekonstruirala tajna i vrijednost te tajne. For petlja se kreće do t-1 (zato što je zadnji t zapravo tajna). U varijablu a sprema se nasumično odabran broj funkcijom secrets.randbits(10), ova funkcija vraća cijeli broj s 10 nasumično odabranih bitova. Zatim, funkcija koef.append(a) dodaje taj izgenerirani a u listu koef. I na kraju se još na tu listu koef funkcijom koef.append(K) doda i vrijednost tajne.

```
14  def polinom(nasumicno, koef):
15      polinomi = []
16      for i in range(len(koef)):
17          vrijednost = nasumicno**(len(koef)-i-1) * koef[i]
18          polinomi.append(vrijednost)
19      polinomi = sum(polinomi)
20
21      return polinomi
```

Slika 20 Metoda polinom(nasumicno, koef)

Na slici prikazana je metoda polinom(nasumicno, koef) koja prima dva parametra nasumicno i koef, koji se računaju u metodi generirajDionicu() i koeficijent(). Napravila sam listu polinomi u koju će se spremati rezultati pojedinačnih pojmova polinoma. Range(len(koef)) označava stupanj polinoma, odnosno i će se kretati do veličine polinoma koef. U varijablu vrijednost radi se računanje gore spomenutih pojmova za svaki stupanj i, koje ide ax^i + ax^i-1 + ax^i-2…Rezultat polinoma se dobije sa sumom sum(polinomi) svih pojmova u listi polinom.

```
23  def generirajDionicu(t, n, K):
24      koef = koeficijent(t, K)
25      dionice = []
26
27      for i in range(0, n):
28          nasumicno = secrets.randbits(10)
29          dionice.append([nasumicno, polinom(nasumicno, koef)])
30
31      return dionice
```

Slika 21 Metoda generirajDionicu(t, n, K)

Na slici prikazana je metoda generirajDionicu(t, n, K) koja prima parametre t, n i K. t se odnosi na broj dionica koje će biti potrebne kada se kasnije bude rekonstruirala tajna, a n se odnosi na broj koliko će se dionica napraviti, dok K označava tajnu koja je upisana u program. U ovom primjeru t je 3, a n je 5, što znači da će se napraviti pet različitih dionica od kojih će kasnije samo tri bilo koje biti potrebne za rekonstrukciju tajne. Varijabla koef poziva već poznatu funkciju koeficijent(t, K). Zatim sam napravila listu dionice koja je za početak prazna, a unutar liste će se spremati svaka dionica koja se izgenerira. Unutar for petlje generirat će se dionice tajne. Stavila sam da petlja ide

od 0 do n, odnosno od 0 do 5 jer je n = 5, što bi značilo da će se generirati pet dionica. Opet sam koristila secrets.randbits(10) koja vraća slučajno generirani cijeli broj od 10 bitova. A zatim će u već spomenutu listu dionice funkcija append spremati varijablu nasumicno, i poznatu metodu polinom(nasumicno, koef). Unutar metode polinom šalje se naš nasumično odabran broj za dionicu te se šalje koef u kojem je već spremljen šifrirani dio tajne. I na kraju nam metoda generirajDionicu vraća listu dionice koja će ispisivati pet dionica tajne.

```
33  def rekonstrukcijaTajne(dioniceZaRekonstrukciju):
34      suma = 0
35      poljeZaDionice = []
36
37      for j in range(len(dioniceZaRekonstrukciju)):
38          xj, yj = dioniceZaRekonstrukciju[j][0], dioniceZaRekonstrukciju[j][1]
39          poljeZaDionice = Fraction(1)
40
41          for i in range(len(dioniceZaRekonstrukciju)):
42              xi = dioniceZaRekonstrukciju[i][0]
43              if i != j:
44                  poljeZaDionice *= Fraction(Fraction(xi)/(xi-xj))
45
46          poljeZaDionice *= yj
47          suma += Fraction(poljeZaDionice)
48
49      return suma
```

Slika 22 Metoda rekonstrukcijaTajne(dioniceZaRekonstrukciju)

Na slici prikazana je metoda rekonstrukcijaTajne(dioniceZaRekonstrukciju), ovom metodom dionice tajne će se rekonstruirati i dati točnu tajnu. Napravila sam varijablu suma u koju će se spremiti tajna te sam napravila listu poljeZaDionice u koju će se spremati dijelovi rekonstruirane tajne iz dionica. Za rekonstrukciju tajne u Shamirovoj shemi koristi se Langrangeov interpolacijski polinom, čija formula ide:

$$L(x) = g_1\left(\frac{x-x_2}{x_1-x_2} * \frac{x-x_3}{x_1-x_3}\right) + g_2\left(\frac{x-x_1}{x_2-x_1} * \frac{x-x_3}{x_2-x_3}\right) + g_3\left(\frac{x-x_1}{x_3-x_1} * \frac{x-x_2}{x_3-x_2}\right),$$

unutar prve for petlje računaju se g-ovi, a unutar druge for petlje računa se postupak iz zagrada. For petlje idu do veličine dionice koju smo nasumično odabrali. U ovom slučaju sam koristila spomenutu funkciju Fraction koja će ovakvu veliku formulu i zapravo razlomke preoblikovati kao cijele brojeve te nam prilikom vraćanja rezultata ne treba nikakva dodatna funkcija koja bi na primjer decimalne brojeve morala vratiti kao cijele brojeve.

```
if __name__ == '__main__':

    t, n = 3, 5
    print('REGISTRACIJA')
    print('\n')

    print('Unesite vaš ID (ID mora biti numerički): ')
    K1 = int(input())

    print('\n')

    dionice1 = generirajDionicu(t, n, K1)
    print('Generirane dionice za vaš ID su:', *dionice1)
    print('')
    print('Molimo zapišite vaše dionice za ID jer prilikom prijave morate unijeti tri različite dionice. ')
    print('*Nemojte izgubiti navedene dionice!*')

    print('\n')

    print('Unesite vašu LOZINKU (LOZINKA mora biti numerička): ')
    K2 = int(input())

    print('\n')

    dionice2 = generirajDionicu(t, n, K2)
    print('Generirane dionice za vašu LOZINKU su:', *dionice2)
    print('')
    print('Molimo zapišite vaše dionice za LOZINKU jer prilikom prijave morate unijeti tri različite dionice. ')
    print('*Nemojte izgubiti navedene dionice!*')
    print('\n')

    print('Unesite vaš PIN (PIN mora biti numerički): ')
    K3 = int(input())
    print('\n')

    dionice3 = generirajDionicu(t, n, K3)
    print('Generirane dionice za vaš PIN su:', *dionice3)
    print('\n')
    print('Molimo zapišite vaše dionice za PIN jer prilikom prijave morate unijeti tri različite dionice.')
    print('*Nemojte izgubiti navedene dionice!*')
    print('\n')
```

Slika 23 Registracija korisnika

U glavnom dijelu programa, zadala sam da se tajna dijeli na pet dionica, a da su tri dionice potrebne za rekonstrukciju (t, n = 3, 5). Nakon toga slijedi registracija, registracija u ovom dijelu je potrebna jer se svaki korisnik prvo mora registrirati kako bi uopće i dobio određene dionice koje su potrebne za prijavu u program. Zatim vidimo kako je prvo potrebno unijeti ID, napomenuto je kako ID mora biti numerički kako ne bi korisnici upisivali slova. Unutar varijable K1 sprema se ono što korisnik upiše da je njegov ID. Nakon što je upisao ID, varijabla dionice1 poziva metodu generirajDionicu(t, n, K) koja će ID podijeliti na pet dionica. Zatim se ispisuje koje su to generirane dionice koje korisnik mora zapisati jer inače neće moći pristupiti programu. Nakon toga postupak se ponavlja, od korisnika se traži da upiše neku lozinku i PIN koji će isto biti numerički te program vraća koje su to generirane dionice za lozinku i PIN.

```
94      print('PRIJAVA')
95      print('\n')
96
97      print('Kako biste unijeli točan ID, potrebno je unijeti tri različite dionice vašeg ID-a.')
98
99      dioniceZaRekonstrukciju = random.sample(dionice1, t)
100     print('Unijeli ste tri dionice za ID:', *dioniceZaRekonstrukciju)
101     print('\n')
102
103
104     tajniId = rekonstrukcijaTajne(dioniceZaRekonstrukciju)
105     print('Rekonstruirana tajna je:', tajniId)
106     print('\n')
107
108     print('Kako biste unijeli točnu LOZINKU, potrebno je unijeti tri različite dionice vaše LOZINKE.')
109
110     dioniceZaRekonstrukciju = random.sample(dionice2, t)
111     print('Unijeli ste tri dionice za LOZINKU:', *dioniceZaRekonstrukciju)
112     print('\n')
113
114
115     tajnaLozinka = rekonstrukcijaTajne(dioniceZaRekonstrukciju)
116     print('Rekonstruirana tajna je:', tajnaLozinka)
117     print('\n')
118
119     print('Kako biste unijeli točan PIN, potrebno je unijeti tri različite dionice vašeg PIN-a.')
120
121     dioniceZaRekonstrukciju = random.sample(dionice3, t)
122     print('Unijeli ste tri dionice za PIN:', *dioniceZaRekonstrukciju)
123     print('\n')
124
125
126     tajniPin = rekonstrukcijaTajne(dioniceZaRekonstrukciju)
127     print('Rekonstruirana tajna je:', tajniPin)
128     print('\n')
```

Slika 24 Prijava korisnika

Na slici vidimo kako se nakon registracije, korisnik mora prijaviti putem dionica koje su mu bile prikazano tijekom registracije i koje je bilo potrebno zapisati. Za potrebe ovog primjera nisam tražila od korisnika da upiše vlastoručno dionice već sam koristila da varijabla dioniceZaRekonstrukciju koriste funkciju random.sample koja uzima nasumično neke tri generirane dionice za ID, kasnije isto funkcija radi za lozinku i PIN. Na ekranu se zatim ispisuje koje su to tri dionice koje je „korisnik“ upisao. Linije koda 105, 116 i 127 u kojima se zapravo radi da ispišu koje su to rekonstruirane tajne, što ne bi bilo sigurno jer bi onda haker znao koje su to tajne, ovako samo po dionicama hakerima bi trebalo dosta vremena da odgonetnu na koji način dionice rekonstruiraju tajnu. Ali za ovaj rad ostavila sam da se vidi koje su to rekonstruirane tajne samo zbog provjere radi li metoda rekonstrukcijaTajne(dioniceZaRekonstrukciju) točno. Varijable tajni Id, tajnaLozinka, tajniPin postoje kako bi se u njih spremala točna tajna jer će kasnije program morati usporediti jesu li točne rekonstruirane tajne s onom početnom tajnom koju je program spremao u K1, K2 i K3.

```
130     if tajniId == K1:
131         if tajnaLozinka == K2:
132             if tajniPin == K3:
133                 print('Uspješno ste prijavljeni u sustav.')
134                 print('DOBRODOŠLI!')
135     else:
136         print('Prijava nije uspjela, dionice nisu dobro unesene.')
```

Slika 25 Provjera rekonstruiranih tajni

Kao što je već u tekstu spomenuto, ovim if-om provjeravamo jesu li rekonstruirane tajne jednake prvobitno unesenim tajnama. Sve rekonstruirane tajne moraju biti točne jer višefaktorska autentifikacije zahtjeva više od dva faktora za autentifikaciju. Ako su sve rekonstruirane tajne točne, program ispisuje da je korisnik uspješno prijavljen u sustav te ga vodi na početnu stranicu programa. Ako neka tajna nije dobro rekonstruirana, što bi značilo da korisnik nije dobro upisao neku dionicu – program vraća kako prijava nije uspjela.

```
139 #Što bi se dogodilo kada bismo upisali manji ili veći broj dionica od praga t=3
140
141 print('\n\n')
142 print("-----------------------------------------------------------------------------------------")
143 print('\n')
144 print('Što bi se dogodilo kada bismo upisali jednu manje dionicu ili jednu više dionicu?')
145 print('\n')
146 print("-----------------------------------------------------------------------------------------")
147 print('\n')
148
149 dionice = [('ID',dionice1, K1), ('LOZINKA',dionice2, K2), ('PIN',dionice3, K3)]
150 prag = [('t-1', t-1),('t+1', t+1)]
151
152 for label_dionica,dionica,value_provjera in dionice:
153     print(f'-------- provjera dionica za {label_dionica} --------')
154     print('\n')
155     for label, value in prag:
156         dioniceZaRekonstrukciju = random.sample(dionica, value)
157         print(f'Dionice koje ćemo koristiti za rekonstrukciju {label_dionica} sa < {label} > su:', *dioniceZaRekonstrukciju)
158         tajna = rekonstrukcijaTajne(dioniceZaRekonstrukciju)
159         print(f'Rekonstrukcija {label_dionica} sa < {label} > je:', tajna)
160         print('\n')
161         if value_provjera == tajna:
162             print(f'{label_dionica} JE TOČAN/TOČNA!')
163         else:
164             print(f'{label_dionica} JE NETOČAN/NETOČNA!')
165             print('\n')
166     print('\n\n')
```

Slika 26 Provjera t-1 i t+1

Na slici prikazana je provjera što bi se dogodilo kada bi se u rekonstrukcijaTajne slao manji ili veći broj dionica od zadanog praga. Unutar liste dionice stavila sam da se nalaze nazivi određenog faktora (ID, LOZINKA, PIN), generirane dionice (dionice1, dionice2, dionice3) te glavne tajne (K1, K2, K3). Unutar liste prag stavila sam nazive t-1 i t+1 što označava da je jedan broj manji od zadanog praga, a drugi je veći. Unutar

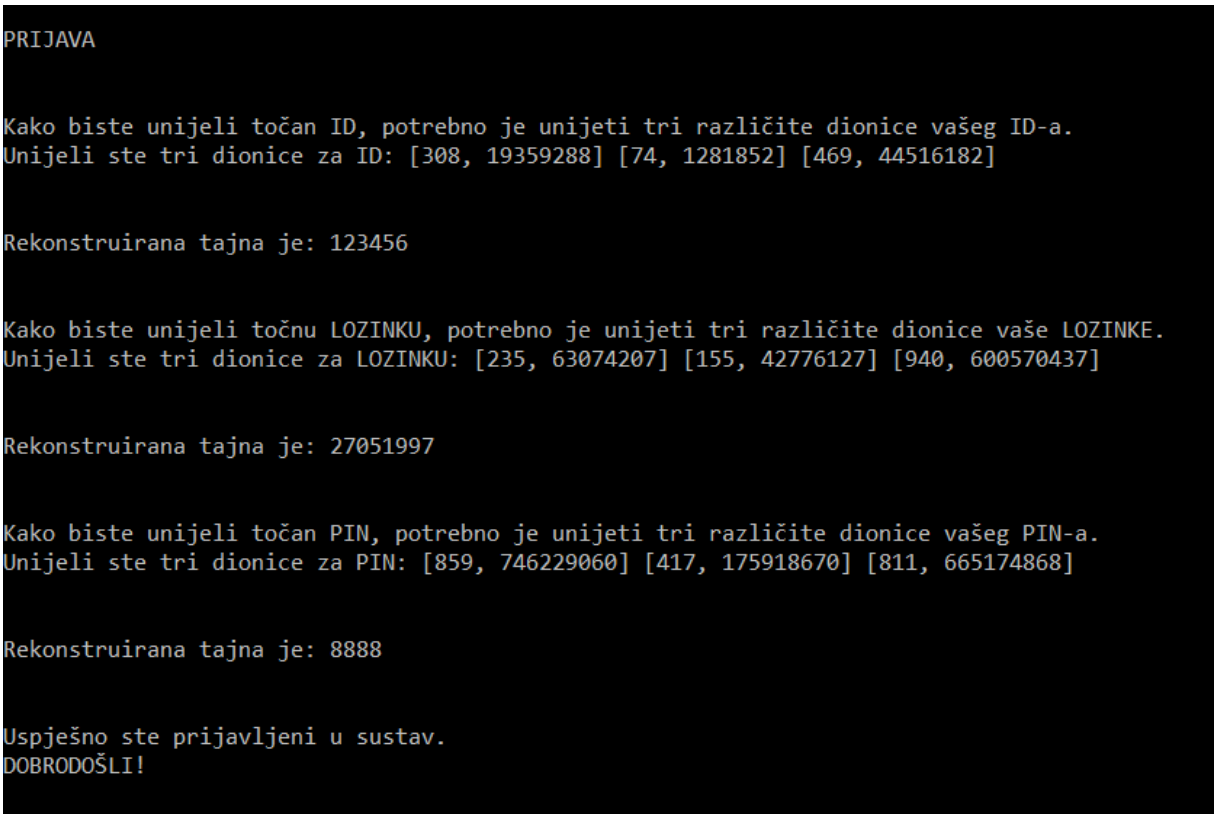
prve for petlje zapravo se prolazi kroz dionice za određeni faktor, a unutar druge for petlje se radi da se varijabli dioniceZaRekonstrukciju nasumično daju t-1 ili t+1, znači ili jedna manje dionica ili jedna više dionica. Zatim se poziva metoda rekonstrukcijaTajne u koju se šalju nasumično odabrane dionice te se vraća koja je rekonstruirana tajna. Ono što se mora dobiti ovom provjerom jest to da ako se uzme jedna dionica manje, rekonstruirana tajna će biti neki nepoznati broj koji nije točan, a ako se uzme jedna dionica više onda bi rekonstruirana tajna trebala biti točna. Jesu li rekonstruirane tajne točne ili netočne provjeravamo pomoću if-a.

Prikaz ispisa programa:

```
D:\Diplomski_Studij\Četvrti semestar>Secret_Sharing_Iva_Sabolek.py
REGISTRACIJA


Unesite vaš ID (ID mora biti numerički):
123456


Generirane dionice za vaš ID su: [308, 19359288] [74, 1281852] [343, 23946178] [469, 44516182] [387, 30407754]

Molimo zapišite vaše dionice za ID jer prilikom prijave morate unijeti tri različite dionice.
*Nemojte izgubiti navedene dionice!*


Unesite vašu LOZINKU (LOZINKA mora biti numerička):
27051997


Generirane dionice za vašu LOZINKU su: [101, 33763851] [786, 428174521] [235, 63074207] [940, 600570437] [155, 42776127]

Molimo zapišite vaše dionice za LOZINKU jer prilikom prijave morate unijeti tri različite dionice.
*Nemojte izgubiti navedene dionice!*


Unesite vaš PIN (PIN mora biti numerički):
8888


Generirane dionice za vaš PIN su: [859, 746229060] [417, 175918670] [811, 665174868] [154, 24025650] [822, 683338310]


Molimo zapišite vaše dionice za PIN jer prilikom prijave morate unijeti tri različite dionice.
*Nemojte izgubiti navedene dionice!*
```

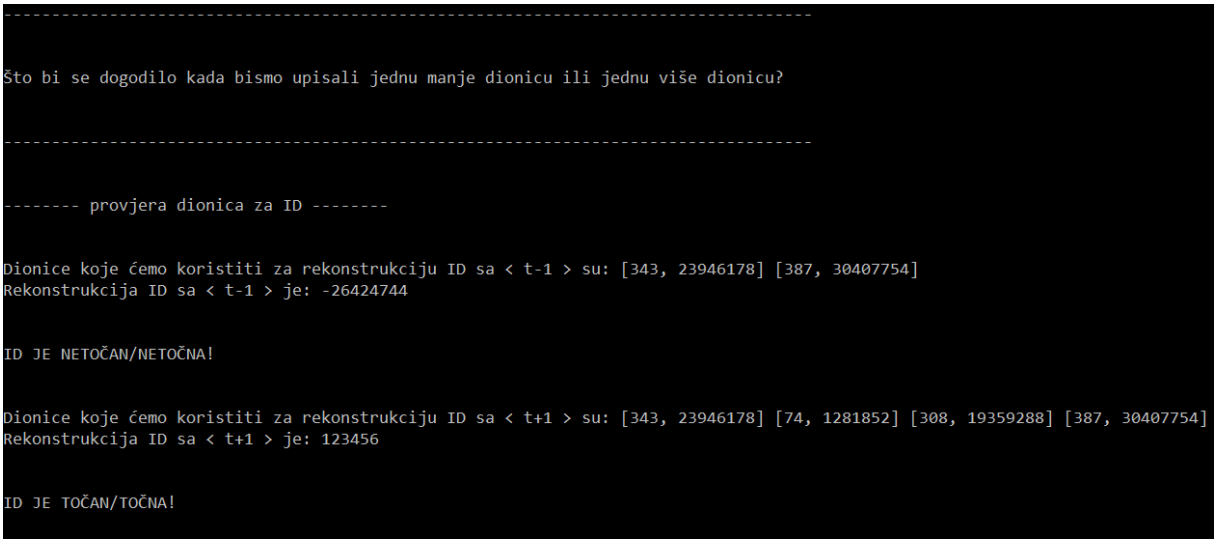
Slika 27 Ispis programa - Registracija

Na slici vidimo da prilikom pokretanja programa traži se registracija korisnika. Korisnik je upisao ID, lozinku i PIN te je za svaki faktor dobio pet dionica koje rekonstruiranjem kasnije će dati upravo tu upisanu tajnu. Ono što se može vidjeti da su sve dionice odabrane nasumično baš pomoću one funkcije secrets.randbits.

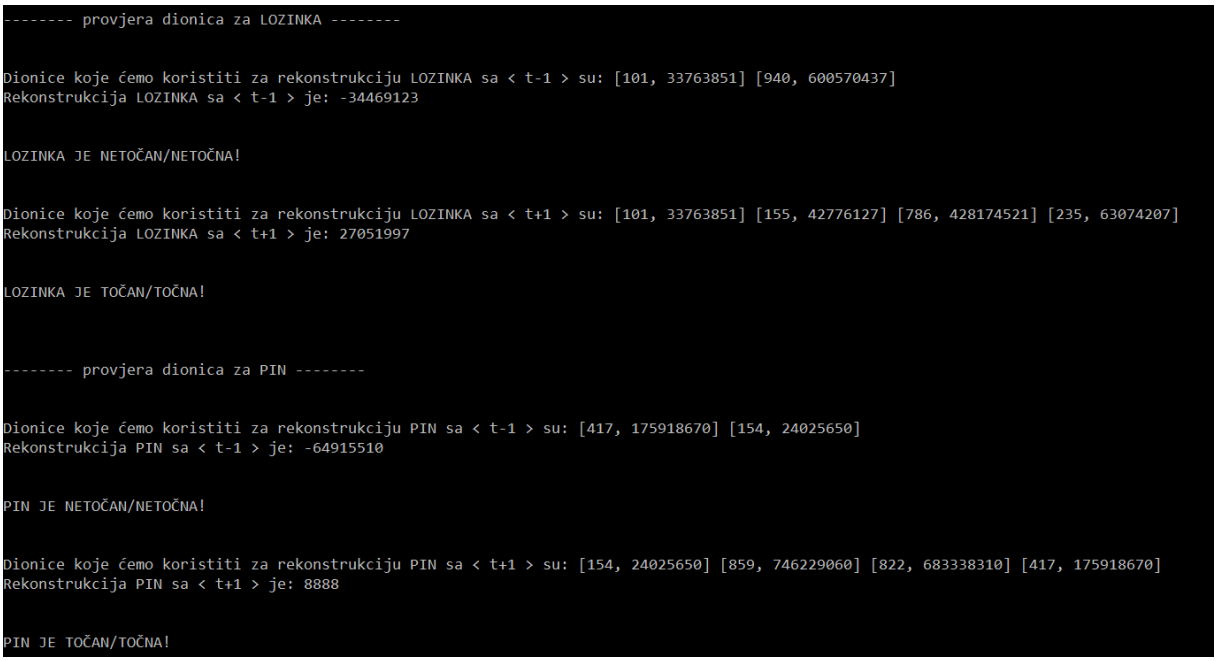


Slika 28 Ispis programa - Prijava

Na slici vidimo da se nakon registracije prikazuje prijava korisnika koji mora unijeti tri dionice za rekonstrukciju ID-a, lozinke i PIN-a. Unutar ovog primjera ostavljeno je da se vidi koje su to rekonstruirane tajne, kada bi ovo bio neki pravi program, taj dio ne bi bio prikazan radi sigurnosti. U ovom slučaju pošto su sve dionice bilo dobro unesene, provjerom jesu li svi faktori točni dobili smo da je korisnik uspješno prijavljen u sustav.



Slika 29 Ispis programa - provjera t-1 i t+1



Slika 30 Ispis programa - provjera t-1 i t+1

Na slikama prikazano je što bi se dogodilo kada bismo uzeli jednu dionicu manje ili jednu dionicu više. Ono što se može vidjeti sa slika jest to da ako se uzme jedna dionica manje, rekonstruirana tajna je netočna, ako se uzme jedna dionica više, rekonstruirana tajna je točna jer prelazi preko praga što znači da je to samo još dodatna dionica koja će pomoći u rekonstrukciji. Nakon svake rekonstrukcije tajne odrađuje se provjera je su li ID, lozinka ili PIN točni ili netočni.

11. ZAKLJUČAK

Danas je autentifikacija puno više korištenija i potrebnija nego u prošlosti, baš iz razloga što se nalazimo u digitalnoj eri te većina korisnika koristi autentifikacija, pogotovo biometrijsku koja je uz lozinke, tokene najbolja što se tiče sigurnosti sustava i autorizacije. Ono što sami nismo svjesni jest to da smo okruženi autentifikacijom u svakodnevnom svijetu, svi koristimo pametne telefone koji su zaključani nekakvom vrstom autentifikacije – još uvijek se koristi jednostavna autentifikacija odnosno da za otključavanje telefona je potrebno samo jedan faktor, ali danas postoji uz uobičajene lozinke, pinove ili uzorka postoji prepoznavanje lica i otisak prsta. Biometrijski faktori ne smatraju se samostalnima, već su nadopuna, odnosno dodataka tradicionalnim pristupima provjere autentičnosti kao što su lozinka, tokeni, PIN-ovi i tokeni. Očekuje se da će kombiniranje dva ili više faktora provjere autentičnosti zapravo pružiti još veća razina sigurnosti pri provjeri identiteta korisnika. Sama autentifikacija dobar je pružatelj kontrole pristupa sustavu ili nekoj aplikaciji, isto tako radi nje se radi povezivanje privatnih podataka samo s pojedincem te uvijek može osigurati istinitost podataka.

Ako još autentifikaciji dodamo i kriptografiju, odnosno u ovom slučaju dijeljenje tajni, sigurnost nam je još više osigurana jer se prilikom autentifikacije ne moraju koristiti točni podaci već šifrirane dionice koje se u sustavu rekonstruiraju pomoću Langrangeovog interpolacijskog polinoma. To znači da će bilo koji faktor autentifikacije, bio on faktor znanja, faktor vlasništva ili biometrijski faktor biti šifriran prema principu dijeljenja tajne. Iz ovog rada može se zaključiti da će svaki faktor biti podijeljen na nekoliko dionica, ovisno o Shamirovoj (t, n) shemi praga, od čega će samo određeni broj dionica biti potrebne za rekonstrukciju tajne.

12. LITERATURA

Knjige:

1. Jain V.K. (2017) *Cryptography and Network Security.* New Delhi: Khanna book publishing co. (P) LTD.
2. *Encyclopedia of Cryptography and Security* (2011) 2. izd. New York: Springer.
3. Smart N. P. (2016) *Cryptography Made Simple.* Switzerland: Springer.

Članci:

4. Benarous, L., Kadri B. i Bouridane A. (2017) A Survey on Cyber Security Evolution and Threats: Biometric Authentication Solutions. U: Jiang, R. et al. (eds.)., *Biometric Security and Privacy: Signal Processing for Security Technologies.* Switzerland: Springer, str. 371-411.
5. Mohsin et al. (2017) Two Factor Vs Multi-factor, an Authentication Battle in Mobile Cloud Computing Environments. U zborniku radova *International Conference on Future Networks and Distributed Systems*. Cambridge, UK, str. 1-10.
6. Ometov et al. (2018) Multi-Factor Authentication: A Survey. MDPI: *Cryptography* 2 (1), str. 1-31. URL: https://www.mdpi.com/2410-387X/2/1/1 [pristup: 20.05.2021.]
7. Kim, J.J. i Hong S.P. (2011) A Method of Risk Assessment for Multi-Factor Authentication. *Journal of Information Processing Systems*, 7 (1), str. 187-198.
8. Konoth, R.K., van der Veen, V. i Bos, H. (2016). How Anywhere Computing Just Killed Your Phone-Based Two-Factor Authentication. U: Grossklags, J. i Preneel B. (eds.), *Financial Cryptography and Data Security*. Berlin, Germany: Springer, str. 405–421.
9. Petsas et al. (2015) Two- factor Authentication: Is the World Ready? Quantifying 2FA Adoption. U zborniku radova *the 8th European Workshop on System Security*. Bordeaux, France, str. 1-7.
10. Milutinović V. (2016) *Dijeljenje tajni.* Diplomski rad. Zagreb: Sveučilište u Zagrebu.

11. Bonneau et. al. (2015) Passwords and the Evolution of Imperfect Authentication. *Communications of the ACM*, 58 (7), str. 78-87.
12. Bhragav-Spantzel et al. (2007) Privacy preserving multi-factor authentication with biometrics. *Journal of Computer Security*, 15, str. 529-560.
13. Jorgensen, Z. i Yu, T. (2011) On mouse dynamics as a behavioral biometric for authentication. U zborniku *radova ASIA Computer and Communications Security*. Hong Kong, China, str. 476-482.
14. Beimel, A. (2011) *Secret-Sharing Schemes: A Survey*. ResearchGate. URL: (PDF) Secret-Sharing Schemes: A Survey (researchgate.net) [pristup 27.05.2021.]
15. Schneier, B. (2005) Two-Factor Authentication: Too Little, Too Late. *Communications of the ACM,* 48 (4), str. 136.
16. Stinson, D. (1999) *Visual cryptography and threshold schemes.* IEEE potentials. URL: Visual cryptography & threshold schemes - IEEE Potentials (jhu.edu) [pristup 27.05.2021.]

Internetske stranice:

17. A beginner's guide to Shamir's Secret Sharing (2020). URL: https://medium.com/keylesstech/a-beginners-guide-to-shamir-s-secret-sharing-e864efbf3648 [pristup 27.05.2021.]
18. Understanding Shamir's Secret Sharing (SSS) (2018). URL: https://medium.com/vault12/understanding-shamirs-secret-sharing-6a4bd27768c9 [pristup 27.05.2021.]
19. Rafaloff, E. (2019) *Shamir's Secret Sharing Scheme.* URL: https://ericrafaloff.com/Cryptography/Shamir's+Secret+Sharing+Scheme [pristup 27.05.2021.]
20. Secrets – Generate secure random number for managing secrets (2021). URL: https://docs.python.org/3/library/secrets.html [pristup 11.09.2021.]
21. Fraction module in Python (2020.) URL: https://www.geeksforgeeks.org/fraction-module-python/ [pristup 11.09.2021.]

13. PRILOZI

13.1. Kazalo slika

13.2. Kazalo tablica