

Primjena kriptografije prilikom plaćanja elektroničkim novcem

AKADEMSKA GODINA: 2021./2022.

KOLEGIJ: KRIPTOGRAFIJA

STUDENTI:

DINO GAZIĆ, MARIJAN BRČINA, JOSIP RELIĆ

FAKULTET INFORMATIKE U PULI

SADRŽAJ

1. UVOD	1
2. OSNOVNI KORACI PLAĆANJA ELEKTRONIČKIM NOVCEM	2
3. PROTOKOLI PLAĆANJA ELEKTRONIČKIM NOVCEM	4
3.1. Osnovni protokoli	4
3.1.1. Protokol bez anonimnosti	4
3.1.2. Protokol s anonimnošću	6
3.2. Konačni oblik protokola	7
4. ZAŠTITNI MEHANIZMI	10
4.1. Digitalni potpis	10
4.2. Slijepi potpis	10
4.3. Identificirajuća informacija	12
5. MOGUĆI NAPADI I OBRANE	14
5.1. Višestruko korištenje ili kopiranje novčanice	14
5.2. Krivotvorenje elektroničkih novčanica	15
5.3. Krađa elektroničke novčanice	15
5.4. Problemi sigurnosti transakcija	15
5.5. Pokušaj napada s kopiranom porukom iz neke od prethodnih transakcija	16
5.6. Pokušaj napadača da pošalje novčanicu banci prije nego što je to kupac stigao učiniti	16
6. MOGUĆI NAPADI I OBRANE – ZAKLJUČAK	17
7. IMPLEMENTACIJA KRIPTOGRAFIJE PRILIKOM PLAĆANJA ELEKTRONIČKIM NOVCEM	18
7.1. KAKO RADI?	18
8. PRIKAZ I OBJAŠNJENJE IMPLEMENTIRANOG PROGRAMSKOG KODA	19
8.1. KLIJENT.PY	19
8.2. BANKA.PY	27
8.3. TRANSAKCIJA.PY	31
8.4. PRIKAZ ISPISANIH DATOTEKA NAKON POKRETANJA PROGRAMA	32
8.4.1. Slijepi novčani nalozi	32
8.4.2. Odslijepljeni novčani nalozi	33
8.4.3. Potpisani novčani nalog	34
8.4.4. Odslijepljeni potpisani novčani nalog	34
9. LITERATURA	35
10. POPIS SLIKA	36

1. UVOD

Elektronički novac je jedan od načina ostvarivanja plaćanja na Internetu. Elektronički novac je i mnogo više od toga. On je zamjena za novac i plaćanje elektroničkim novcem nalikuje na obično plaćanje gotovinom. Kako se transakcije elektroničkim novcem odvijaju preko Interneta, potrebno je ostvariti visoku razinu sigurnosti takvih transakcija te razviti posebne metode zaštite. Postupci zaštite uključuju kriptiranje prometa između strana koje sudjeluju u novčanim transakcijama, provjeru autentičnosti obiju strana te sprječavanje zlouporabe sustava.

Podlogu spomenutim postupcima pružaju kriptografski algoritmi te dodatno razvijeni protokoli koji osiguravaju zaštitu elektroničkog novca, kao i sudionika transakcija. Privatnost i autentičnost su bitne osobine elektroničkog sustava plaćanja.

U ovom dokumentu opisan je elektronički novac, metode plaćanja elektroničkim novcem te sustavi plaćanja. Također, opisani su protokoli koji se primjenjuju kod plaćanja elektroničkim novcem te su kroz kod opisani sustavi i način plaćanja elektroničkim novcem. Razmatrani su i mogući sigurnosni problemi koji se javljaju u elektroničkom poslovanju.

2. OSNOVNI KORACI PLAĆANJA ELEKTRONIČKIM NOVCEM

Radi što bolje analize problema zaštite postupka plaćanja elektroničkim novcem, na početku smo objasnili osnovne korake tog postupka.

Elektroničko poslovanje je, po definiciji, svaka financijska transakcija koja koristi podatke razmijenjene elektroničkim putem. Elektroničko plaćanje je zaseban dio elektroničke trgovine. Protokol elektroničkog plaćanja čini niz međukoraka na čijem kraju je plaćanje obavljeno. Plaćanje se može obaviti korištenjem tzv. žetona, objekata koji sadrže vrijednost i koje je izdao posrednik. Ni osoba koja plaća ni osoba kojoj je plaćena roba ili usluga ne izdaje žeton kojim je plaćanje obavljeno, već obje prihvataju žeton kojeg je izdala banka, organizacija ili država kao valjano platežno sredstvo.

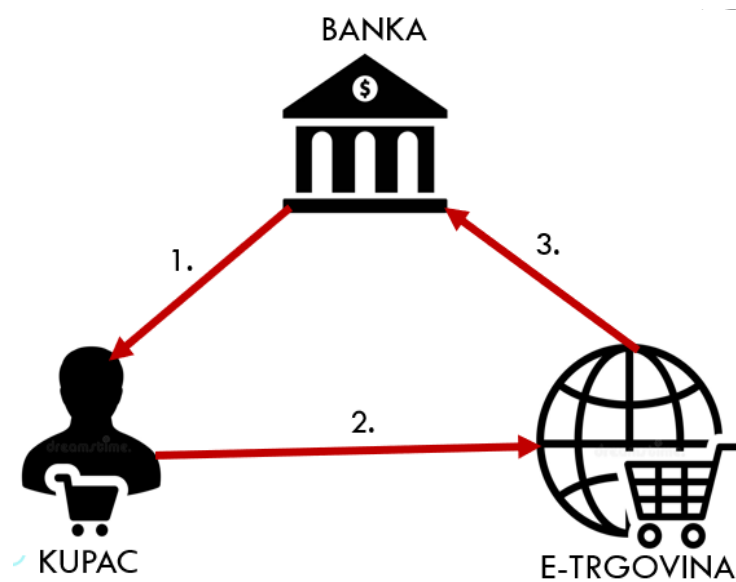
U svakom obliku protokola plaćanja elektroničkim novcem javljaju se tri vrste sudionika, kao što možete vidjeti i na slici 1:

- osoba koja plaća elektroničkim novcem (Kupac),
- osoba koja je plaćena elektroničkim novcem (Trgovac) i
- izdavač elektroničkih novčanica (Banka).

Osnovni protokol elektroničkog plaćanja čine tri koraka:

- podizanje novca - kupac u zamjenu za pravi novac dobiva neki oblik elektroničkog novca.
- plaćanje—kupac prenosi dio elektroničkog novca trgovini.
- 3. polaganje novca - trgovina šalje elektronički novac dobiven od kupca banci i banka mu zauzvrat povećava stanje na njegovom računu (ili isplaćuje gotovinu).

Komunikacija je uspostavljena između sva tri sudionika. S obzirom da se sva tri opisana koraka odvijaju putem Interneta, nazire se da takav nezaštićen oblik komunikacije nije siguran niti za jednog od sudionika transakcije. To znači da poruka, elektronička novčanica koja, na primjer, u 2. koraku putuje od kupca prema trgovcu, nije sigurna da će stići na odredište u izvornom obliku, da će uopće stići na odredište, da će biti neovlašteno kopirana ili da će u 2. koraku biti simuliran od lažnog kupca, dakle nije sigurno da će odredište primiti važeću ili izmišljenu poruku. Isti zaključak vrijedi za svaki pojedini korak protokola plaćanja elektroničkim novcem.



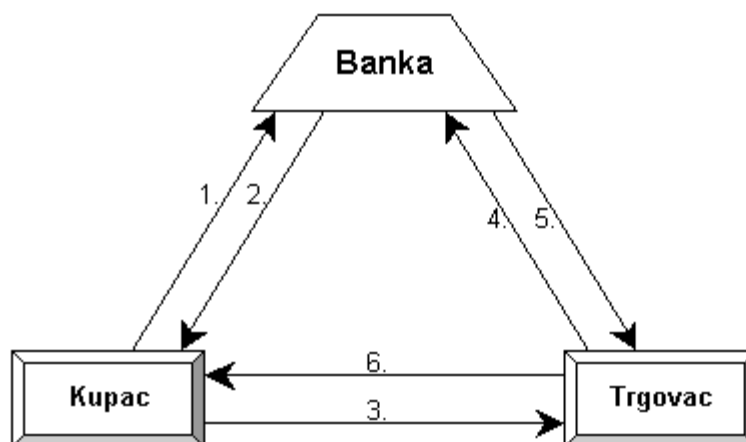
Slika 1. Osnovni koraci plaćanja elektroničkim novcem

3. PROTOKOLI PLAĆANJA ELEKTRONIČKIM NOVCEM

3.1. Osnovni protokoli

U daljnjem dijelu teksta slijede opisi dvaju jednostavnih protokola čija je jedina razlika u prisutnosti osobine anonimnosti kupca u sustavu plaćanja elektroničkim novcem. Podjela na dva osnovna protokola više je uvedena radi jednostavnijeg shvaćanja uvođenja i funkcioniranja mehanizma koji osigurava anonimnost kupca u transakciji, nego radi njihove pojedinačne uporabivosti i međusobne različitosti.

3.1.1. Protokol bez anonimnosti



Slika 2. Osnovni protokol

U ovom shematskom prikazu vidimo da su sudionici transakcije plaćanja elektroničkim novcem kupac, trgovac i banka te se radi o online sustavu plaćanja elektroničkim novcem. On-line sustav znači da se provjera valjanosti elektroničkog novca obavlja odmah nakon što trgovac primi elektronički novac od kupca. Suprotno on-line sustavu razlikujemo off-line sustav plaćanja elektroničkim novcem u kojem se provjera elektroničkog novca odgađa za kasnije. Da bi se provjera obavila, novac se prosljeđuje banci.

Šest koraka sa slike mogu se grupirati u tri faze. Prva faza predstavlja podizanje elektroničkog novca s bankovnog računa:

1. kupac šalje zahtjev banci za određenom količinom elektroničkog novca (korak 1.)
2. banka oblikuje elektroničku novčanicu te ostavlja digitalni potpis
3. banka šalje elektroničku novčanicu kupcu te umanjuje njegov račun (korak 2.)

Druga faza se nastavlja gdje kupac plaća odabrane artikle

1. kupac šalje elektronički novac trgovcu (korak 3.)
2. trgovac provjerava digitalni potpis banke na primljenoj novčanici

I završna treća faza predstavlja provjeru elektroničkog novca od strane banke i isporuku artikla kupcu od strane trgovca:

1. trgovac šalje elektroničku novčanicu banci (korak 4.)
2. banka provjerava potpis na novčanici
3. banka uspoređuje serijski broj novčanice s postojećima u bazi uporabljenih elektroničkih novčanica
4. banka unosi serijski broj novčanice u bazu uporabljenih novčanica
5. banka povećava račun trgovca
6. banka šalje odgovor trgovcu (korak 5.)
7. trgovac šalje kupljenu robu kupcu (korak 6.)

U drugoj točki prve faze protokola, banka potpisuje elektroničku novčanicu. Točnije, banka pridodaje elektroničkoj novčanici digitalni potpis iste čime je riješen potencijalni problem krivotvorenja elektroničkog novca. Ako se uz elektroničku novčanicu ne nalazi bankin potpis, novčanica je nevažeća.

Prilikom potpisivanja elektroničke novčanice u prvoj fazi, banka može zapamtiti vezu serijskog broja na novčanici i osobe koja je podigla novac, dakle kupca. Naknadno primanje iste te novčanice omogućuje banci praćenje transakcije koju je prethodno pokrenuo kupac. Opisani nedostatak protokola plaćanja elektroničkim novcem rješava se mehanizmom slijepog potpisa koji onemogućuje banku da dovede u vezu podizanje elektroničke novčanice s njenim deponiranjem.

Time će biti ostvarena anonimnost kupca dok će banka biti sigurna u valjanost elektroničke novčanice.

3.1.2. Protokol s anonimnošću

Ovaj protokol, za razliku od protokola iz prethodnog poglavlja, karakterizira osobina anonimnosti kupca u transakciji, odnosno nemogućnost praćenja transakcije u sustavu plaćanja elektroničkim novcem.

Ovaj protokol se može pratiti po koracima prošlog protokola. Jedina bitna razlika u odnosu na prošli protocol je već u prvoj fazi, odnosno fazi stvaranja digitalnog novca.

Ovaj protokol možemo opisati prema slijedećim koracima:

1. kupac oblikuje elektroničku novčanicu te ju “pokriva”
2. kupac šalje “pokrivenu” elektroničku novčanicu banci zajedno sa zahtjevom za podizanje određene svote elektroničkog novca (korak 1.)
3. banka potpisuje primljenu “pokrivenu” elektroničku novčanicu i umanjuje račun kupcu
4. banka šalje potpisanu elektroničku novčanicu kupcu (korak 2.)
5. kupac uklanja faktor pokrivanja s elektroničke novčanice

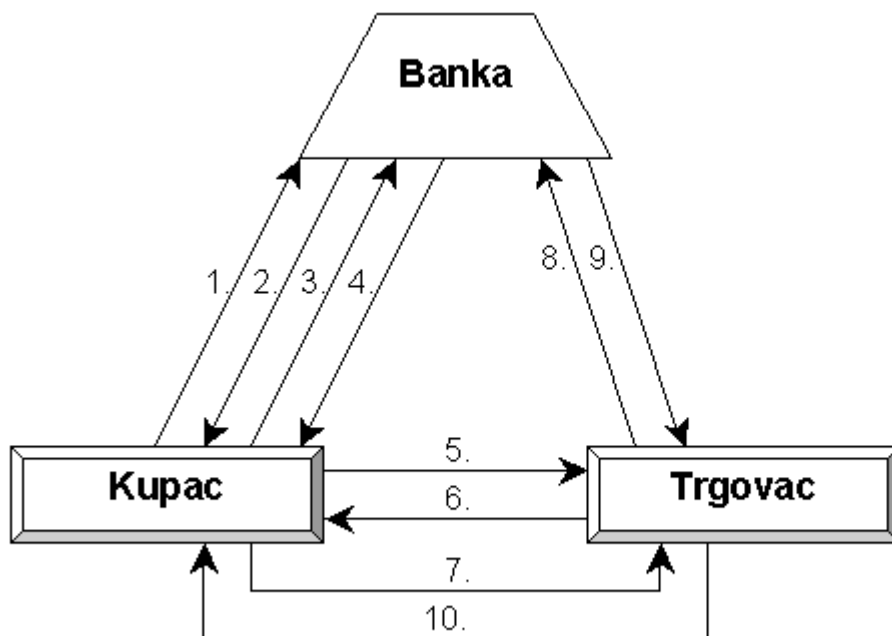
Druga i treća faza ovog protokola ne razlikuju se od druge i treće faze protokola iz prethodnog poglavlja, stoga opis te dvije faze pogledati u prethodnom poglavlju.

U prvoj točki prve faze protokola kupac sam oblikuje elektroničku novčanicu. To je preduvjet da bi se ostvarila osobina anonimnosti kupca u transakciji. Nakon toga kupac “pokriva” elektroničku novčanicu koju je upravo oblikovao te na taj način čini serijski broj novčanice nevidljivim banci. Banka prilikom potpisivanja novčanice ne može zabilježiti serijski broj koji se nalazi na novčanici, stoga ne može dovesti u vezu podizanje elektroničke novčanice s njenim deponiranjem na kraju transakcije.

3.2. Konačni oblik protokola

Protokol plaćanja elektroničkim novcem iz prethodnog poglavlja predstavlja osnovu za konačni oblik protokola čiji opis slijedi u ovom poglavlju. Također, protokol iz prethodnog poglavlja ima jedan nedostatak, a taj je da bi u takvom protokolu nemoguće bilo otkriti prijevaru dvostruke potrošnje[17]. Prijevara dvostruke potrošnje javlja se kada se isti elektronički novac iskoristi u dvije ili više transakcija. Stoga se jedna elektronička novčanica upotrebljava samo u jednoj transakciji. Da bi se spriječila potrošnja iste elektroničke novčanice u dvije ili više transakcija koristi se mehanizam identificirajuće informacije. Identificirajuća informacija postaje djelom elektroničke novčanice. Temeljni princip je takav da ako kupac koji sudjeluje u transakciji koristi elektronički novac samo unutar jedne transakcije, njegov identitet ostaje sačuvan, dakle osobina anonimnosti je prisutna. Ako se već potrošen elektronički novac pokuša koristiti u novoj transakciji, identitet kupca iste transakcije otkrit će se već pri prvoj provjeri takvog novca u banci.

Shematski prikaz protokola možemo vidjeti na slici ispod:



Slika 3. Konačni oblik protokola plaćanja elektroničkim novcem

Ovaj protokol se sastoji od ukupno deset koraka, a možemo ga podijeliti u tri faze.

Prva faza:

1. kupac oblikuje n elektroničkih novčanica koje nose jednaki iznos, ali različiti serijski broj
2. kupac prikriva n elektroničkih novčanica
3. kupac šalje n prikrivenih elektroničkih novčanica banci (korak 1.)
4. banka šalje zahtjev kupcu za otkrivanje n-1 slučajno odabrane elektroničke novčanice (korak 2.)
5. kupac šalje banci n-1 traženi faktor prikrivanja i n-1 odgovarajuću identificirajuću informaciju (korak 3.)
6. banka provjerava valjanost n-1 elektroničke novčanice (iznos i identificirajuću informaciju)
7. banka potpisuje preostalu elektroničku novčanicu
8. banka šalje potpisanu elektroničku novčanicu kupcu te umanjuje račun kupca (korak 4.)
9. kupac uklanja faktor prikrivanja s potpisane novčanice te provjerava potpis banke

Za razliku od prvog protokola bez anonimnosti, ovaj konačni protokol u prvoj fazi ima četiri koraka. Dodatna dva koraka malo više opisuju kriptografski mehanizam slijepog potpisa. Također, banka provjerava prisutnost ispravne identificirajuće informacije na n-1 elektroničkoj novčanici čime se osigurava u slučaju dvostruke potrošnje.

Druga faza:

1. kupac šalje potpisanu elektroničku novčanicu trgovcu (korak 5.)
2. trgovac provjerava digitalni potpis banke uz elektroničku novčanicu
3. trgovac šalje kupcu slučajno generirani odabirući niz[18] (korak 6.)
4. kupac šalje tražene informacije trgovcu (korak 7.)
5. trgovac provjerava valjanost dijela identificirajuće informacije na elektroničkoj novčanici

U ovoj fazi spominje se odabirući niz. Taj pojam je vezan uz mehanizam identificirajuće informacije koji će biti opisan dolje niže u poglavljima. U posljednjoj točki druge faze trgovac se uvjerava da je elektronička novčanica zbilja vlasništvo kupca koji komunicira s trgovcem. Na taj način može se

odmah uštedjeti posao slanja elektroničke novčanice banci u slučaju primljene tuđe novčanice, dakle u slučaju prijevare.

Treća faza:

1. trgovac šalje potpisanu elektroničku novčanicu, identificirajući niz, dio identificirajuće informacije i broj bankovnog računa banci (korak 8.)
2. banka provjerava digitalni potpis uz primljenu elektroničku novčanicu
3. banka uspoređuje serijski broj elektroničke novčanice s onima u bazi uporabljenih novčanica
4. banka unosi serijski broj elektroničke novčanice, odabirući niz i dio identificirajuće informacije u bazu uporabljenih elektroničkih novčanica
5. banka šalje odgovor trgovcu o ispravnosti elektroničke novčanice i uvećava račun trgovca (korak 9.)
6. trgovac provjerava odgovor banke
7. trgovac šalje robu kupcu (korak 10.)

U posljednjoj fazi, kao i u prethodne dvije, pretpostavlja se da se svaki korak odvija ispravno, dakle da se transakcija odvija u najboljem redu, bez ikakvih pokušaja krivotvorenja, prijevare ili višestruke potrošnje.

Konačni oblik protokola ima sve predispozicije za njegovu implementaciju, odnosno korištenje u realnim uvjetima.

Krivotvorenje se sprječava digitalnim potpisom banke u kojoj se nalaze bankovni računi kupca i trgovca. Ugrađena je osobina anonimnosti koju osigurava slijepi potpis. Višestruka potrošnja sprječava se mehanizmom identificirajuće informacije. Integritet elektroničke novčanice osiguran je digitalnim potpisom. Sve nabrojane osobine konačnog oblika protokola plaćanja elektroničkim novcem osiguravaju ispravnu i prihvatljivu transakciju unutar trokuta kupac-trgovac-banka, dakle gledano s više razine svakog od tri spomenuta sudionika transakcije.

4. ZAŠTITNI MEHANIZMI

4.1. Digitalni potpis

Digitalni potpis osigurava banku od krivotvorenja elektroničkog novca na način da se uz svaku elektroničku novčanicu mora nalaziti njen digitalni potpis.

4.2. Slijepi potpis

Prilikom izdavanja elektroničke novčanice, banka može uz odgovarajući serijski broj zapamtiti i osobu kojoj je izdala novčanicu te na taj način narušiti privatnost iste osobe prilikom primanja iste te novčanice tijekom neke transakcije. U tom slučaju, u protokol plaćanja elektroničkim novcem ugrađuje se mehanizam slijepog potpisa koji osigurava anonimnost kupca koji ulazi u transakciju.

Postoji više načina na koji se slijepi potpis može izvesti, dok će ovdje biti opisan jedan koji je implementiran u konačnom obliku protokola plaćanja elektroničkim novcem.

Kupac, prije nego što pošalje elektroničku novčanicu banci, prikriva novčanicu sa slučajno generiranim brojem, tzv. faktorom prikrivanja. Prikrivenu elektroničku novčanicu šalje banci. Banka potpisuje primljenu prikrivenu novčanicu te umanjuje račun kupca za dogovoreni iznos. Banka šalje potpisanu novčanicu kupcu koji po primitku novčanice uklanja faktor prikrivanja te dobija potpisanu elektroničku novčanicu. U ovom bi slučaju jedino kupac bio zadovoljan jer banka zapravo ne zna što potpisuje, pogotovo kada se radi o iznosu elektroničke novčanice. Opisani postupak se proširuje tako da kupac umjesto jedne prikrivene novčanice šalje banci n prikrivenih elektroničkih novčanica. Na svakoj elektroničkoj novčanici jednak je iznos novčanice, ali su serijski brojevi različiti. Kada banka primi svih n prikrivenih elektroničkih novčanica, ona slučajno odabire n-1 od tih novčanica te traži od kupca faktore prikrivanja za odabrane novčanice. Nakon što primi tražene faktore prikrivanja, otkriva odabrane elektroničke novčanice te provjerava njihovu

valjanost. Najbitnije je da je iznos na svakoj od $n-1$ novčanice jednak dogovorenom iznosu. Ako su odabrane novčanice valjane, banka potpisuje preostalu prikrivenu elektroničku novčanicu. Potpisanu novčanicu šalje kupcu te umanjuje njegov račun. Vjerojatnost da je banka prevarena je $1/n$.

Jedna od izvedbi slijepog potpisa koristi kriptiranje RSA algoritmom kao digitalnim potpisom. Pretpostavimo:

(n, e) je javni ključ

(n, d) je tajni ključ

m je elektronička novčanica koja zadovoljava $0 \leq m < n$

Prvi korak:

Kupac slučajno odabire broj k između 1 i n te prikriva elektroničku novčanicu m :

$$T = mk^e \bmod n$$

Drugi korak:

Banka potpisuje prikrivenu elektroničku novčanicu t :

$$t^d = (mk^e)^d \bmod n$$

Treći korak:

Kupac uklanja faktor prikrivanja:

$$s = t^d / k \bmod n = m^d k / k \bmod n = m^d \bmod n$$

$$(mk^e)^d = m^d k^{ed} = 1 \bmod n \quad \text{gdje } ed = aL(n) + 1 = m^d k^{aL(n)+1} \bmod n$$

4.3. Identificirajuća informacija

Identificirajuća informacija sastavni je dio svake elektroničke novčanice koju potpisuje banka da bi se pri tom osigurala od prijevare dvostruke potrošnje. Mehanizam identificirajuće informacije otkriva identitet sudionika transakcije koji je pokušao ili izvršio prijevaru dvostruke potrošnje, dok poštenog sudionika ostavlja u anonimnosti.

U postupku oblikovanja identificirajuće informacije koristi se niz nizova identifikacijskih bitova. Identifikacijski bitovi generiraju se na temelju podataka karakterističnih za osobu koja sudjeluje u transakciji, odnosno za osobu koja generira elektroničku novčanicu. Ti podaci mogu biti ime i prezime osobe, e-mail, telefonski broj te ostale bitne informacije o sudioniku transakcije koje ga identificiraju. Tako generirani identifikacijski nizovi razdvajaju se postupkom dijeljenja tajne na dva dijela. Opširnije o tom postupku nešto kasnije. Svaki dio za sebe ne govori ništa o osobi koja je generirala identifikacijske nizove, ali oba dijela zajedno čine identifikacijski niz koji identificira osobu. Nakon što su svi identifikacijski nizovi razdijeljeni na dva dijela, svi parovi zajedno predstavljaju identificirajuću informaciju. Svaki par sastoji se od lijeve i desne polovice.

Odabirući niz je zapravo niz 0 i 1 pri čemu, ovisno o dogovoru, 1 znači jednu, dok 0 predstavlja drugu polovicu identifikacijskog niza. Redoslijed 0 i 1 u odabirućem nizu predstavlja redoslijed identifikacijskih nizova čije se polovice traže. Nakon što kupac pošalje polovice identifikacijskih nizova, trgovac ih uspoređuje s onima na elektroničkoj novčanici. Treba napomenuti da se na elektroničkoj novčanici nalaze sažeci polovica identifikacijskih nizova, tako da trgovac prije usporedbe obavlja hash funkciju nad svakom polovicom. Sažetak polovice identifikacijskog niza osigurava tajnost te polovice. U trenutku depozita, trgovac uz elektroničku novčanicu šalje banci odabirući niz i primljene polovice identifikacijskih nizova. Banka uspoređuje serijski broj elektroničke novčanice s onima u bazi uporabljenih novčanica. Ako je elektronička novčanica već korištena, banka uparuje dvije različite polovice istog identifikacijskog niza i otkriva identitet kupca

koji je pokušao prijevaru dvostruke potrošnje. Jedna od polovica koje se uparuju je iz baze, dok je druga ona upravo primljena od strane trgovca zajedno s elektroničkom novčanicom. Utvrđivanje identiteta na taj način moguće je jer su, s velikom vjerojatnošću, odabirući niz iz baze i upravo primljeni odabirući niz različiti. Odabirući nizovi trebali bi biti različiti jer ih trgovac generira slučajno u svakoj transakciji. Ako se identificirajuća informacija sastoji od n parova polovica identifikacijskih nizova, vjerojatnost da trgovac dva puta generira isti odabirući niz je $1/2^n$. Dakle vjerojatnost je zanemarivo mala i opada s brojem identifikacijskih nizova koji se nalaze na elektroničkoj novčanici. Poželjno je staviti čim više identifikacijskih nizova, ali opet ne previše radi veličine same elektroničke novčanice.

Da trgovac ne bi sam postavio lažnu identificirajuću informaciju na novčanicu te time generirao lažni odabirući niz i odgovarajuće polovice identifikacijskih nizova, sprječava ga digitalni potpis banke na elektroničkoj novčanici.

Iz opisanog mehanizma identificirajuće informacije jasno se vidi da je protokol plaćanja elektroničkim novcem osiguran od prijave dvostruke potrošnje bilo od strane kupca bilo od strane trgovca.

5. MOGUĆI NAPADI I OBRANE

5.1. Višestruko korištenje ili kopiranje novčanice

Kako bi se valjanost elektroničkog novca mogla provjeriti i dokazati koristi se metoda digitalnog potpisa. Svaka valjana novčanica nosi potpis financijske institucije koja ju je izdala. Elektronički novac sastoji se od niza bitova čije je kopiranje jednostavno. Kopija se ne razlikuje od originala pa bi krivotvorenje bilo nemoguće otkriti. Jednostavni sustavi bi dozvoljavali kopiranje elektroničkog novca i potrošnju obje kopije. Sustavi elektroničkog plaćanja moraju sprječavati dvostruku potrošnju. Višestruko korištenje iste novčanice u nekoliko transakcija ili kopiranje iste elektroničke novčanice sprječava se upisivanjem serijskog broja korištene novčanice u bazu podataka banke. Svaki puta kada banka primi neku novčanicu, ona provjerava serijski broj u svojoj bazi podataka i zna je li novčanica već bila korištena ili nije. Ukoliko banka otkrije pokušaj prijevare, identificira osobu koja je pokušala prijevaru preko podataka o identifikaciji koje klijent šalje uz novčanicu

Kod online sustava višestruka potrošnja sprječava se tako što se obvezuje trgovca da stupi u vezu s bankom tokom svake prodaje. Računalo banke održava bazu podataka potrošenog elektroničkog novca i može jednostavno javiti trgovcu ako je korišteni elektronički novac još uporabljiv. U protivnom slučaju trgovac odbija prodaju.

Kod offline sustava postoje dva pristupa otkrivanju dvostruke potrošnje, sklopovski i programski pristup. Sklopovski pristup se oslanja na posebnu pametnu karticu koja sadrži čip otporan na neovlaštene promjene. U tom čipu čuva malu bazu podataka o elektroničkom novcu koje je ta pametna kartica potrošila. Ako vlasnik kartice pokuša kopirati manju svotu elektroničkog novca i potrošiti ga dva puta, ugrađeni čip bi otkrio pokušaj i ne bi dozvolio transakciju. Spomenuti je čip otporan na neovlaštene promjene i vlasnik ne može obrisati bazu podataka bez trajnog oštećenja kartice. Programski pristup uključuje oblikovanje elektroničkog novca i kriptografskih protokola koji otkrivaju identitet osobe koja je dva puta upotrijebila novčanicu do trenutka kada elektronički novac dolazi u banku.

5.2. Krivotvorenje elektroničkih novčanica

Krivotvorenje elektroničkih novčanica nije moguće jer banka stavlja digitalni potpis na svaku novčanicu i taj potpis se ne može krivotvoriti. Potpis se obavlja tajnim ključem banke koji zna samo ona. Kada se novčanica vrati u banku, ona provjerava svoj potpis. Na taj način je osigurano da novčanicu nije nitko drugi stvorio.

5.3. Krađa elektroničke novčanice

U posljednjoj točki druge faze konačnog protokola za plaćanje elektroničkim novcem, trgovac provjerava valjanost podataka o identifikaciji na elektroničkoj novčanici. Na taj način se uvjerava da je elektronička novčanica uistinu vlasništvo kupca te je onemogućena njena krađa.

5.4. Problemi sigurnosti transakcija

Sigurnost sustava za elektroničko plaćanje ovisi o sigurnosti koju pružaju kriptografski algoritmi. Kriptografski algoritmi i protokoli pružaju visok stupanj sigurnosti i ako su ispravno primijenjeni, sigurnost ne bi trebala biti ugrožena. Međutim, i dalje postoji prostor za napredak kriptanalize te neizbježni ljudski faktor (gubitak tajnog ključa, provala u sustav, ucjena) kojim se ta sigurnost može ugroziti. Omogućavanje sigurnih transakcija zahtijeva stvaranje elektroničkih sustava sigurnosti. Spomenuti sustavi moraju štiti poruke koje zahtijevaju prijenos novca ili sam novac te pružati uslugu stvaranja digitalnih potpisa koji se ne mogu krivotvoriti niti poricati.

Takvi sustavi moraju osiguravati:

- **privatnost** - sadržaj prenesenih poruka, činjenica da je uopće poslana, tko ju je poslao i kome je namijenjena trebaju ostati poznati samo sudionicima u komunikaciji
- **vjerodostojnost** - sudionici u komunikaciji moraju biti u stanju jedan drugom dokazati svoj identitet,

- **autorizacija sudionika** – pravo da identificirani sudionik ima ovlasti koristiti se određenom uslugom u određenom trenutku,
- **neopozivost** - niti jedna strana ne može drugoj strani osporiti da je sudjelovala u komunikaciji
- **cjelovitost, integritet** - poruka mora na svoje odredište stići u nepromijenjenu obliku, odnosno primatelj poruke mora moći otkriti bilo kakvu promjenu u komunikacijskom kanalu.

Navedeni zahtjevi mogu se ispuniti primjenom raznih tehnoloških rješenja.

5.5. Pokušaj napada s kopiranom porukom iz neke od prethodnih transakcija

Ako treća strana pokuša napad s kopiranom porukom iz neke od prethodnih transakcija, komunikacija se odmah odbija jer se u svakoj novoj transakciji stvaraju novi simetrični ključevi te se izvorne poruke kriptiraju s tim novim ključevima. Na taj način, određena poruka kriptirana simetričnim ključem u jednoj transakciji nije jednaka poruci iz istog koraka u drugoj transakciji iako se možda kupuje identična roba kod istog trgovca.

5.6. Pokušaj napadača da pošalje novčanicu banci prije nego što je to kupac stigao učiniti

Kod obavljanja transakcija elektroničkim novcem upotrebom protokola elektroničkog plaćanja, napadač može prisluškivati i presresti elektronički novac kojeg kupac šalje trgovcu. Napadač može poslati novčanicu banci prije nego što je to kupac stigao učiniti. Napadač će povećati iznos na svojem računu, a za kupca će se smatrati da je kriminalac jer pokušava drugi put unovčiti istu novčanicu. Ovakav se napad može spriječiti uspostavljanjem tajnog komunikacijskog kanala između kupca i trgovca tako da napadač ne može presresti novčanice niti saznati odvija li se transakcija.

6. MOGUĆI NAPADI I OBRANE – ZAKLJUČAK

Protokoli koji se koriste pri razmjeni elektroničkog novca počivaju na sigurnosnom sustavu koji koristi više elemenata: algoritme za kriptiranje, funkcije sažetka, jednosmjerne funkcije, generatore slučajnih brojeva, lozinke itd. Svaki od ovih elemenata podložan je napadu zlonamjernog napadača. No, u praksi se pokazalo da su najčešći i najuspješniji napadi usmjereni na ljude, korisnike sustava. Ako treća strana može prisluškivati komunikaciju između dvije strane te ju želi samo ometati, to joj je uvijek omogućeno. Dakle, umjesto poruke koja je na putu od jedne strane ka drugoj, treća strana može spomenutu poruku promijeniti i dalje ju pustiti prema cilju. Osim toga, napadač ju može ukloniti s njenog puta i/ili poslati novu poruku umjesto izvorne.

Uz to može jednostavno izmisliti neku novu poruku i poslati je postojećem cilju iako nikakva poruka nije namijenjena tom cilju. Bar jedna od spomenutih operacija je moguća bez obzira na kriptografske metode koje se koriste u komunikaciji. Rezultat takvog napada je, ovisno o robusnosti sustava koji se napada, ometanje pravilnog funkcioniranja sustava. U pravilu, takvim ometanjem ne bi se trebala nanijeti ozbiljnija šteta, već se uglavnom usporava rad sustava koji se ometa. Na primjer, započeta transakcija se tijekom obavljanja ometa te ju je potrebno ponoviti. Problem se rješava tako da se onemogući bilo kakvo prisluškivanje od neželjene treće strane. Na Internetu je tako nešto nemoguće ostvariti zbog njegove veličine, količine protokola koji se koriste te činjenice da nisu svi komunikacijski kanali zaštićeni.

Ako kupac ne zaštiti svoje računalo dovoljno dobro protiv napada preko mreže, napadač može situaciju iskoristiti za neovlašten pristup sustavu, te na taj način i direktorijima te datotekama koje se nalaze na ugroženom sustavu. Napadač može kopirati digitalne novčanice i potrošiti ih umjesto kupca. Također, može dva puta iskoristiti novčanicu. Dobra zaštita protiv takvih napada su razni enkripcijski i kriptografski programi i protokoli.

7. IMPLEMENTACIJA KRIPTOGRAFIJE PRILIKOM PLAĆANJA ELEKTRONIČKIM NOVCEM

Svrha je implementirati protokole za korištenje elektroničkog novca pomoću raznih protokola za očuvanje tajnosti, anonimnosti, integriteta i uzajamnog povjerenja. Implementacija sustava elektroničkog novca unutar kojeg elektronički novac ne može biti kopiran ili korišten više od jedanput i u kojem je privatnost klijentovog identiteta zagarantirana. Novčane transakcije se odvijaju između dvije strane: klijenta i banke.

Elektronički novac korišten prilikom tih transakcija je datoteka koja sadržava:

- Iznos transakcije
- Jedinstveni string broj
- Klijentov ID
- Potpis banke (prije nego klijent može koristiti elektronički novac)

7.1. KAKO RADI?

- Klijent generira tri novčana naloga
- Različiti nasumični jedinstveni string brojevi su primijenjeni na svaki novčani nalog
- Razdijeljivanje tajne (eng. Secret splitting) i obvezivanje bitovima (eng. Bit commitment) su protokoli implementirani na identifikacijski string koji opisuje klijenta, te je slijepi potpis protokol koji je implementiran na sva tri novčana naloga
- Banka nasumično odabire jedan od tri novčana naloga poslana od strane klijenta da ostane neotvoren
- Te na kraju algoritam potvrđuje da su dva novčana naloga ispunjena sa validnim informacijama

8. PRIKAZ I OBJAŠNJENJE IMPLEMENTIRANOG PROGRAMSKOG KODA

U nastavku slijedi prikaz programskog koda za svaku kreiranu datoteku, te objašnjenje pojedinih funkcija odnosno metoda unutar samog koda kao i objašnjenje zasebnih dijelova koda unutar funkcija. Proći ćemo kroz svaku od datoteka. Dakle, najprije ide prikaz `klijent.py` datoteke, zatim prikaz `banka.py` i za kraj prikaz datoteke `transakcija.py`.

8.1. KLIJENT.PY

Krenimo najprije s programskim kodom koji je kreiran za klijenta. Na samom početku importamo potrebne library-e. Zatim kreiramo klasu **Klijent** unutar kojeg izvršavamo inicijalizaciju samog klijenta. U donjoj for petlji kreiramo tri različita novčana naloga.

```
import hashlib
import random
import gmpy2

class Klijent(object):
    def __init__(self, iznos, identitet, kljucevi):
        print("Inicijalizacija klijenta")
        self.iznos = iznos
        self.identitet = identitet
        self.kljucevi = kljucevi
        self.novcani_nalozi = {}

        for i in range(1, 4):
            ime_novcanog_naloga = "novcani_nalog" + str(i)
            self.novcani_nalozi[ime_novcanog_naloga] = self.kreiranje_novcanog_naloga(ime_novcanog_naloga)
```

Slika 4. KLIJENT.PY – importanje library-a i kreiranje klase klijent

Nadalje imamo funkciju naziva **obvezivanje_bitovima** koja nam zapravo izvršava sami process obvezivanja bitovima (eng. Bit Commitment) korištenjem dostavljene podijeljene tajne **identifikacijski_integer**. Koristi sha256 (eng. Secured Hash Algorithm) za kreiranje obvezivanja bitovima i vraća polje koje sadržava hash vrijednost i dva nasumično generirana broja. Isto tako, u

samoj funkciji imamo dvije varijable **nasumicni_broj1** i **nasumicni_broj2** u koje spremamo dva nasumično generirana broja kroz pozivanje funkcije **generator_slucajnog_broja()** koju ćemo prikazati kasnije. U ostatku koda unutar funkcije imamo izračun hasha od varijable **identifikacijski_integer** i nasumičnih brojeva. Sha256 biblioteka zahtjeva stringove, a ne integer, pa su integeri konkatenerani i nakon toga hashirani.

```
def obvezivanje_bitovima(self, identifikacijski_integer):  
  
    nasumicni_broj1 = self.generator_slucajnog_broja()  
    nasumicni_broj2 = self.generator_slucajnog_broja()  
  
    int_string = str(identifikacijski_integer) + str(nasumicni_broj1) + str(nasumicni_broj2)  
    byte_string = bytes(int_string, encoding='utf-8')  
    hash_vrijednost = hashlib.sha256(byte_string).hexdigest()  
    int_vrijednost = int(hash_vrijednost, 16) % self.kljucevi['n']  
  
    return [int_vrijednost, nasumicni_broj1, nasumicni_broj2]
```

Slika 5. KLIJENT.PY – prikaz funkcije za obvezivanje bitovima

Nakon prikaza funkcije za obvezivanje bitovima slijedi prikaz funkcije **ProcesZasljepljivanjaNovcanogNaloga**. Proces zasljepljivanja novčanog naloga koristi isporučenu tajnu za zasljepljivanje baziranu na **self.kljucevi** te vraća zasljepljene informacije. **slijepi_nocani_nalozi** je varijabla odnosno dictionary koji sadržava zasljepljene novčane naloge. Varijabla **n** nam služi kao lokalna varijabla unutar koje spremamo ključeve['n']. Zatim, iteriramo kroz sve novčane naloge kako bi ih zasljepljili. Kreiramo kontejner naziva **slijepi_novcani_nalog** za kreiranje slijepih novčanih naloga. Onda nam slijedi izvlačenje originalnog novčanog naloga za rad te pohrana istog u varijablu **originalni_novcani_nalog** i kreiranje zasljepljujućeg faktora te pohrana istog u varijablu **zasljepljujuci_faktor**. Nakon toga imamo pokretanje procesa zasljepljivanja za iznos i jedinstvenost. Za kraj unutar funkcije unutar for petlje iteriramo kroz ključeve u novčanom nalogu i pronalazimo stringove identiteta ključeva. **Identifikacijski_string** je lista od dva elementa koja sadrži listu od dva elementa.

```

def ProcesZasljepljivanjaNovcanogNaloga(self):
    print("Zasljepljivanje novcanih naloga")
    self.slijepi_novcani_nalozi = {}
    n = self.kljucevi['n']
    for novcani_nalog in self.novcani_nalozi.keys():
        slijepi_novcani_nalog = {}
        originalni_novcani_nalog = self.novcani_nalozi[novcani_nalog]
        zasljepljujuci_faktor = originalni_novcani_nalog['k'] ** self.kljucevi['e'] % n

        slijepi_novcani_nalog['naziv'] = originalni_novcani_nalog['naziv']
        slijepi_novcani_nalog['iznos'] = (originalni_novcani_nalog['iznos'] * zasljepljujuci_faktor % n)
        slijepi_novcani_nalog['jedinственost'] = (originalni_novcani_nalog['jedinственost'] * zasljepljujuci_faktor % n)

        for kljuc in originalni_novcani_nalog.keys():
            if not kljuc.startswith('I'):
                continue
            slijepi_novcani_nalog[kljuc] = {}
            slijepi_novcani_nalog[kljuc]['identifikacijski_string'] = []
            for i in originalni_novcani_nalog[kljuc]['identifikacijski_string']:
                slijepi_hash = (i[0] * zasljepljujuci_faktor % n)
                slijepi_nasumicni = (i[1] * zasljepljujuci_faktor % n)
                slijepi_novcani_nalog[kljuc]['identifikacijski_string'].append([slijepi_hash, slijepi_nasumicni])

        self.slijepi_novcani_nalozi[novcani_nalog] = slijepi_novcani_nalog

```

Slika 6. KLIJENT.PY – prikaz funkcije procesa zasljepljivanja novcanog naloga

Nakon procesa zasljepljivanja novcanog naloga imamo funkciju **kreiranje_stringa_identiteta** koja kreira potrebne dijelove stringa identiteta te vraća dictionary koji izgleda ovako:

identifikacijski_string : [[lijevi_hash, lijevi nasumicni_broj1], [desni_hash, desni nasumicni_broj1]]

razotkrivajuće_polje: [[lijevi generirani brojevi], [desni generirani brojevi]]

```

def kreiranje_stringa_identiteta(self):
    n, t = self.razdijeljivanje_tajne()
    lijevi_hash, n1, n2 = self.obvezivanje_bitovima(identifikacijski_integer=n)
    desni_hash, t1, t2 = self.obvezivanje_bitovima(identifikacijski_integer=t)
    id_string = [[lijevi_hash, n1], [desni_hash, t1]]
    razotkrivajuće_polje = [[n, n1, n2], [t, t1, t2]]

    return {'identifikacijski_string':id_string, 'razotkrivajuće_polje':razotkrivajuće_polje}

```

Slika 7. KLIJENT.PY – prikaz funkcije za kreiranje stringa identiteta

Zatim, imamo funkciju naziva **kreiranje_novcanog_naloga** koja kreira dictionary koji sadržava potrebna polja za novčani nalog. Koristi isporučeni naziv za novčani nalog te vraća kreirani novčani nalog.

```
def kreiranje_novcanog_naloga(self, naziv):  
  
    print("Kreiranje novcanog naloga %s" % naziv)  
    print("Pokretanje obvezivanja bitovima")  
    novcani_nalog = {}  
    novcani_nalog['naziv'] = naziv  
    novcani_nalog['iznos'] = self.iznos  
    novcani_nalog['jedinственost'] = self.generator_slucajnog_broja()  
    novcani_nalog['k'] = self.generator_slucajnog_broja()  
    novcani_nalog['I1'] = self.kreiranje_stringa_identiteta()  
    print("Kreiranje stringa identiteta I1")  
    novcani_nalog['I2'] = self.kreiranje_stringa_identiteta()  
    print("Kreiranje stringa identiteta I2")  
    novcani_nalog['I3'] = self.kreiranje_stringa_identiteta()  
    print("Kreiranje stringa identiteta I3")  
  
    return novcani_nalog
```

Slika 8. KLIJENT.PY – prikaz funkcije za kreiranje novčanog naloga

Nakon kreiranja novčanog naloga imamo funkciju **ispisi_novcani_nalog** koja služi za ispis novčanog naloga u datoteku. Koristi ključ novčanog naloga kao **novcani_nalozi** i **vrsta_novcanog_naloga** da bi generirao ime datoteke koju naposljetku i kreira.


```

def ispisi_novcani_nalog(self, novcani_nalozi, vrsta_novcanog_naloga):

    print("Ispisivanje novcanog naloga...")
    for novcani_nalog in novcani_nalozi.keys():
        polje_stringova = []
        ime_datoteke = '%s%s.txt' % (vrsta_novcanog_naloga, novcani_nalog)
        ispisi_novcani_nalog = novcani_nalozi[novcani_nalog]
        naziv_str = "Naziv: %s" % ispisi_novcani_nalog['naziv']

        polje_stringova.append(naziv_str)
        elektronicni_iznos = "Iznos: %d" % ispisi_novcani_nalog['iznos']

        polje_stringova.append(elektronicki_iznos)
        elektronicna_jedinstvenost = "Jedinstvenost: %d" % ispisi_novcani_nalog['jedinstvenost']

        polje_stringova.append(elektronicka_jedinstvenost)
        I1_identifikacijski_string = "I1 identifikacijski string: %s" % str(ispisi_novcani_nalog['I1']['identifikacijski_string'])

        polje_stringova.append(I1_identifikacijski_string)
        I2_identifikacijski_string = "I2 identifikacijski string: %s" % str(ispisi_novcani_nalog['I2']['identifikacijski_string'])

        polje_stringova.append(I2_identifikacijski_string)
        I3_identifikacijski_string = "I3 identifikacijski string: %s" % str(ispisi_novcani_nalog['I3']['identifikacijski_string'])

        polje_stringova.append(I3_identifikacijski_string)

        if 'potpis' in ispisi_novcani_nalog:
            potpis_string = "potpis: %s" % str(ispisi_novcani_nalog['potpis'])
            polje_stringova.append(potpis_string)
        with open(ime_datoteke, 'w') as f:
            for i in polje_stringova:
                f.write(i + '\n')

```

Slika 9. KLIJENT.PY – prikaz funkcije za ispis novčanog naloga

Slijedi prikaz ranije spomenute funkcije **generator_slucajnog_broja** koja generira slučajni broj i vraća ga.

```

def generator_slucajnog_broja(self):
    '''Generira slucajni broj i vraca ga'''
    nasumicna_donja_vrijednost = 100
    nasumicna_gornja_vrijednost = 10000
    return random.randint(nasumicna_donja_vrijednost, nasumicna_gornja_vrijednost) % self.kljucevi['n']

```

Slika 10. KLIJENT.PY – prikaz funkcije za generiranje slučajnog broja

Primanje_potpisa je funkcija koja primi potpis banke za novčani nalog i izvršava potpisivanje. Forsira **potpisani_nn[novcaninalog]** u dictionary, inače je varijabla povezana referencom a ne jedinstvena vrijednost.

```
def primanje_potpisa(self, novcaninalog, potpis):
    print("Priljen potpisani novčani nalog")
    potpisani_nn = {}

    potpisani_nn[novcaninalog] = dict(self.slijepi_novcani_nalozi[novcaninalog])
    potpisani_nn[novcaninalog]['potpis'] = potpis
    self.potpisani_novcani_nalog = potpisani_nn
```

Slika 11. KLIJENT.PY – prikaz funkcije za primanje potpisa

Slijedi funkcija **razotkrivanje** koja otkriva dijelove identiteta za zahtjevane novčane naloge. Koristi listu ključeva novčanih naloga (nn1, nn2, itd.) i vraća self.novcaninalozi[nn*][n/t, n1, n2].

```
def razotkrivanje(self, novcaninalozi):
    print("Razotkrivanje odabranih novčanih naloga")
    razotkriveni_brojevi = {}
    for nn in novcaninalozi:
        originalni_nn = self.novcani_nalozi[nn]
        razotkriveni_brojevi[nn] = {}
        for kljuc in originalni_nn.keys():
            if kljuc.startswith('I'):
                razotkriveni_brojevi[nn][kljuc] = originalni_nn[kljuc]['razotkrivajuće_polje']
    return razotkriveni_brojevi
```

Slika 12. KLIJENT.PY – prikaz funkcije za razotkrivanje odabranih novčanih naloga

Funkcija **razdijeljivanje_tajne** obalja process razdijeljivanja tajne. Koristi self.identitet da bi kreirala n i t, te vraća n i t.

```
def razdijeljivanje_tajne(self):
    n = self.generator_slucajnog_broja()
    t = n ^ self.identitet

    return n, t
```

Slika 13. KLIJENT.PY – prikaz funkcije za razdijeljivanje tajne

Nadalje imamo funkciju **odsljepljivanje** koja izvršava proces odsljepljivanja za novčani nalog. Očekuje dostavljenu listu imena, odnosno naziva novčanih naloga te postavlja varijablu `odsljepljeni_novcani_nalozi`. **Odsljepljeni_novcani_nalozi** je self varijabla za pohranu odsljepljenih novčanih naloga. Unutar n varijable spremamo ključevi['n'] kao lokalne varijable. Nakon toga iteriramo kroz sve dane ključeve novčanih naloga kako bi ih odsljepili. Zatim, imamo kreiranje lokalnih varijabli za originalne i slijepe novčane naloge, kreiranje faktora odsljepljivanja, prazni kontejner odnosno dictionary **odsljepi_nn** za svaki odsljepljeni novčani nalog te pokretanje samog procesa odsljepljivanja. Nakon toga iteriramo kroz ključeve u novčanom nalogu i pronalazimo ključeve stringa identiteta.

```
def odsljepljivanje(self, novcaninalozi):  
  
    print("Odsljepljivanje novčanih naloga")  
    self.odsljepljeni_novcani_nalozi = {}  
    n = self.ključevi['n']  
  
    for nn in novcaninalozi:  
        originalni_nn = self.novcani_nalozi[nn]  
        slijepi_nn = self.slijepi_novcani_nalozi[nn]  
  
        inv_k = int(gmpy2.invert(originalni_nn['k'], n))  
        faktor_odsljepljivanja = (inv_k ** self.ključevi['e']) % n  
  
        odsljepi_nn = {}  
  
        odsljepi_nn['naziv'] = originalni_nn['naziv']  
        odsljepi_nn['iznos'] = (slijepi_nn['iznos'] * faktor_odsljepljivanja % n)  
        odsljepi_nn['jedinственost'] = (slijepi_nn['jedinственost'] * faktor_odsljepljivanja % n)  
  
        for ključ in slijepi_nn.keys():  
            if not ključ.startswith('I'):  
                continue  
            odsljepi_nn[ključ] = {'identifikacijski_string': []}  
            for i in slijepi_nn[ključ]['identifikacijski_string']:  
                odsljepi_hash = (i[0] * faktor_odsljepljivanja % n)  
                odsljepi_nasumicne = (i[1] * faktor_odsljepljivanja % n)  
                odsljepi_nn[ključ]['identifikacijski_string'].append([odsljepi_hash,  
                                                                    odsljepi_nasumicne])  
  
        self.odsljepljeni_novcani_nalozi[nn] = odsljepi_nn
```

Slika 14. KLIJENT.PY – prikaz funkcije za odsljepljivanje novčanih naloga

Za kraj prikaza programskog koda što se klijenta tiče imamo funkciju naziva **odsljepi_potpisani_novcani_nalog** koja obavlja process odsljepljivanja potpisanog novčanog naloga te postavlja varijablu **odsljepljeni_potpisani_novcani_nalog**. **Odsljepljeni_potpisani_novcani_nalog** je varijabla za pohranu odsljepljenih potpisanih novčanih naloga. Varijabla **n** pohranjuje ključevi['n'] kao lokalne varijable. Zatim, imamo iteraciju kroz sve ključeve danih novčanih naloga da bi ih odsljepili, kreiranje lokanih varijabli za originalne i slijepe novčane naloge, kreiranje faktora odsljepljivanja, kreiranje praznog kontejnera za svaki odsljepljeni novčani nalog, pokretanje procesa odsljepljivanja, te iteriranje kroz ključeve u novčanom nalogu i ključeve stringa identiteta kao i ključa potpisa.

```
def odsljepi_potpisani_novcani_nalog(self):
    print("Odsljepljivanje novčanih naloga")
    self.odsljepljeni_potpisani_novcani_nalog = {}
    n = self.ključevi['n']

    for nn in self.potpisani_novcani_nalog.keys():
        originalni_nn = self.novcani_nalozi[nn]
        slijepi_nn = self.potpisani_novcani_nalog[nn]

        inv_k = int(gmpy2.invert(originalni_nn['k'], n))
        faktor_odsljepljivanja = (inv_k ** self.ključevi['e']) % n

        odsljepljeni_nn = {}

        odsljepljeni_nn['naziv'] = originalni_nn['naziv']
        odsljepljeni_nn['iznos'] = (slijepi_nn['iznos'] * faktor_odsljepljivanja % n)
        odsljepljeni_nn['jedinstvenost'] = (slijepi_nn['jedinstvenost'] * faktor_odsljepljivanja % n)

        for kljuc in slijepi_nn.keys():
            if kljuc == 'potpis':
                odsljepljeni_potpis = []
                for i in slijepi_nn[kljuc]:
                    odsljepi_i = i * faktor_odsljepljivanja % n
                    odsljepljeni_potpis.append(odsljepi_i)
                odsljepljeni_nn[kljuc] = odsljepljeni_potpis
            if kljuc.startswith('I'):
                odsljepljeni_nn[kljuc] = {'identifikacijski_string': []}
                for i in odsljepljeni_nn[kljuc]['identifikacijski_string']:
                    odsljepi_hash = (i[0] * faktor_odsljepljivanja % n)
                    odsljepi_nasumicni = (i[1] * faktor_odsljepljivanja % n)
                    odsljepljeni_nn[kljuc]['identifikacijski_string'].append([odsljepi_hash,
                                                                              odsljepi_nasumicni])

        self.odsljepljeni_potpisani_novcani_nalog[nn] = odsljepljeni_nn
```

Slika 15. KLIJENT.PY – prikaz funkcije za odsljepljivanje potpisanih novčanih naloga

8.2. BANKA.PY

Nakon prikaza programskog koda koji je kreiran za klijenta, slijedi prikaz programskog koda koji se tiče banke. Kao u slučaju s klijent datotekom na samom početku importamo potrebne library-e. Zatim kreiramo klasu **Banka** unutar koje izvršavamo inicijalizaciju banke. Na donjoj slici također imamo prikaz funkcije **hash_izracun** koja koristi isporučeno **polje_int** te izračunava hash.

```
import hashlib
import random

class Banka(object):
    def __init__(self, kljucevi):
        print("Inicijalizacija banke")
        self.kljucevi = kljucevi

    def hash_izracun(self, polje_int):
        int_string = str(polje_int[0]) + str(polje_int[1]) + str(polje_int[2])
        byte_string = bytes(int_string, encoding='utf-8')
        hash_vrijednost = hashlib.sha256(byte_string).hexdigest()
        int_vrijednost = int(hash_vrijednost, 16) % self.kljucevi['n']

        return int_vrijednost
```

Slika 16. BANKA.PY – prikaz kreiranja klase Banka i funkcije za za hash izračun

Funkcija **verifikacijski_izracun** verificira da se otkrivene informacije podudaraju sa slijepim novčanim nalogom te da su svi iznosi isti.

```

def verifikacijski_izracun(self):
    iznosi = []
    for nn in self.odsljepiti_novcane_naloge:
        otkrij_informacije = self.otkrij_informacije[nn]
        odsljepi_nn = self.odsljepi_novcane_naloge[nn]

        iznosi.append(odsljepi_nn['iznos'])

        for kljuc in otkrij_informacije.keys():
            identifikacijski_string = odsljepi_nn[kljuc]['identifikacijski_string']

            lijevi = otkrij_informacije[kljuc][0]
            desni = otkrij_informacije[kljuc][1]

            izracunato_lijevo = [self.hash_izracun(lijevi), lijevi[1]]
            izracunato_desno = [self.hash_izracun(desni), desni[1]]
            izracunati_identifikacijski_string = [izracunato_lijevo, izracunato_desno]

            if not izracunati_identifikacijski_string == identifikacijski_string:
                print("Novcani nalog: %s, Kljuc: %s" % (nn, kljuc))
                print("Izracunato: %s, Dato: %s" % (str(izracunati_identifikacijski_string),
                                                    str(identifikacijski_string)))
                return False
    iznosi = set(iznosi)
    if len(iznosi) > 1:
        print("Iznosi se ne podudaraju")
        return False
    return True

```

Slika 17. BANKA.PY – prikaz funkcije za verifikiranje informacija

Slijedi prikaz triju funkcija za dohvaćanje pod nazivima **dohvati_slijepe_novcane_naloge**, **dohvati_otkrivajuće_informacije** te funkcije **dohvati_odslijepljene_novcane_naloge**. Funkcija **dohvati_slijepe_novcane_naloge** dohvaća slijepe novčane naloge od klijenta te postavlja self varijablu za pohranu proslijeđenog parametra funkcije **novcaninalozi**. Funkcija **dohvati_otkrivajuće_informacije** dohvaća otkrivajuće informacije za odslijepljene novčane naloge postavlja self varijablu za pohranu **otkrij_informacije**. Dok funkcija **dohvati_odslijepljene_novcane_naloge** dohvaća odslijepljene novčane naloge od klijenta te postavlja self varijablu za pohranu **novcaninalozi** koja nam je zapravo parametar funkcije.

```

def dohvati_slijepe_novcane_naloge(self, novcaninalozi):
    print("Banka je primila slijepe novčane naloge")
    self.slijepi_novcane_nalozi = novcaninalozi

def dohvati_otkrivajuće_informacije(self, otkrij_informacije):
    print("Banka je primila informacije o novčanim nalogima")
    self.otkrij_informacije = otkrij_informacije

def dohvati_odsljepljene_novcane_naloge(self, novcaninalozi):
    print("Banka je primila odsljepljene novčane naloge")
    self.odsljepi_novcane_naloge = novcaninalozi

```

Slika 18. BANKA.PY – prikaz funkcija za dohvaćanje

Nakon prikaza funkcija za dohvaćanje slijedi prikaz funkcije **potpisi_novcane_nalog** koja služi za potpis datog novčanog naloga.

```

def potpisi_novcane_nalog(self):
    if not self.verifikacijski_izracun():
        print('Odsljepljeni novčani nalozi ne mogu biti verificirani')

    potpis_banke = []
    d = self.kljucevi['d']
    n = self.kljucevi['n']
    slijepi_nn = self.potpisati_novcane_nalog

    potpis_banke.append(slijepi_nn['iznos'] ** d % n)
    potpis_banke.append(slijepi_nn['jedinственост'] ** d % n)
    id_kljucevi = ['I1', 'I2', 'I3']
    for ključ in id_kljucevi:
        for i in slijepi_nn[ključ]['identifikacijski_string']:
            potpis_banke.append(i[0] ** d % n)
            potpis_banke.append(i[1] ** d % n)
    print("Verificirano: Banka potpisuje")

    self.potpis_banke = potpis_banke

```

Slika 19. BANKA.PY – prikaz funkcije za potpis novčanog naloga

Za kraj slijedi prikaz funkcije **odsljepi_zajtjeve** koja kreira listu novčanih naloga koje treba odsljepiti. Imamo dvije varijable **nasumicni_donji_broj** i **nasumicni_gornji_broj** koje nam zapravo služe kao parametri za generiranje nasumičnog broja na temelju duljine primljenih slijepih novčanih naloga. Zatim imamo varijablu **potpisati_nn_index** unutar koje spremam nasumični index broja za potpis, nakon toga slijedi kreiranje liste svih mogućih ključeva i uzimanje nasumično odabranoh indexa, postavljanje self varijable za slijepe novčane naloge koje treba potpisati I za kraj kreiranje self varijable za sve novčane naloge koje treba odsljepiti.

```
def odsljepi_zajtjeve(self):  
    print("Banka zahtijeva odsljepljenje")  
  
    nasumicni_donji_broj = 0  
    nasumicni_gornji_broj = len(self.slijepi_novcani_nalozi.keys()) - 1  
  
    potpisati_nn_index = random.randint(nasumicni_donji_broj, nasumicni_gornji_broj)  
  
    lista_kljuceva = list(self.slijepi_novcani_nalozi.keys())  
    potpisati_nn_kljuc = lista_kljuceva[potpisati_nn_index]  
  
    self.potpisati_novcani_nalog_kljuc = potpisati_nn_kljuc  
    self.potpisati_novcani_nalog = self.slijepi_novcani_nalozi[potpisati_nn_kljuc]  
  
    odsljepiti_nn = list(self.slijepi_novcani_nalozi.keys())  
    odsljepiti_nn.pop(odsljepiti_nn.index(potpisati_nn_kljuc))  
    self.odsljepiti_novcane_naloge = odsljepiti_nn
```

Slika 20. BANKA.PY – prikaz funkcije za odsljepljivanje zahtjevanih novčanih naloga

8.3. TRANSAKCIJA.PY

Nakon prikaza programskog koda za klijenta i banku slijedi prikaz programskog koda koji izvršava transakcijski dio. Za početak importamo prethodno kreirane programske datoteke **klijent** i **banka**, nakon toga ključeve klijenta i banke, iznos koji šaljemo te klijentov id. Slijedi spremanje podataka klijenta i banke u varijable i za kraj izvođenje transakcije te ispis novčanih naloga.

```
import klijent
import banka

klijent_kljucevi = {'e': 324, 'n': 3853}
banka_kljucevi = {'d': 534, 'n': 3853}
iznos = 560
klijent_id = 253183

klijent_podaci = klijent.Klijent(iznos=iznos, identitet=klijent_id, kljucevi=klijent_kljucevi)
banka_podaci = banka.Banka(kljucevi=banka_kljucevi)

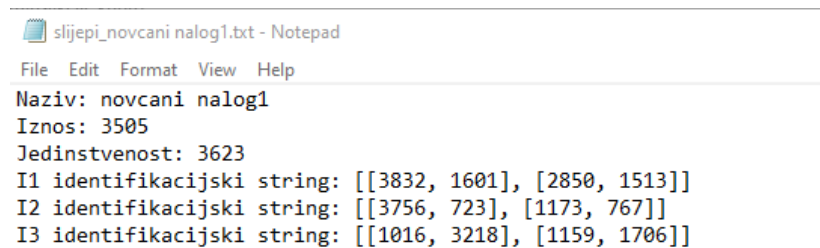
klijent_podaci.ProcesZasljepljivanjaNovcanogNaloga()
banka_podaci.dohvati_slijepe_novcane_naloge(klijent_podaci.slijepi_novcane_nalozi)
banka_podaci.odsljepi_zahitjeve()
klijent_podaci.odsljepljivanje(banka_podaci.odsljepiti_novcane_naloge)
banka_podaci.dohvati_odsljepljene_novcane_naloge(klijent_podaci.odsljepljeni_novcane_nalozi)
banka_podaci.dohvati_otkrivajuće_informacije(klijent_podaci.razotkrivanje(banka_podaci.odsljepiti_novcane_naloge))
banka_podaci.potpisi_novcane_nalog()
klijent_podaci.primanje_potpisa(banka_podaci.potpisati_novcane_nalog_kljuc, banka_podaci.potpis_banke)
klijent_podaci.odsljepi_potpisani_novcane_nalog()
klijent_podaci.ispisi_novcane_nalog(klijent_podaci.slijepi_novcane_nalozi, 'slijepi')
klijent_podaci.ispisi_novcane_nalog(klijent_podaci.odsljepljeni_novcane_nalozi, 'odsljepljeni')
klijent_podaci.ispisi_novcane_nalog(klijent_podaci.potpisani_novcane_nalog, 'potpisani')
klijent_podaci.ispisi_novcane_nalog(klijent_podaci.odsljepljeni_potpisani_novcane_nalog, 'odsljepljeni potpisani')
```

Slika 21. TRANSAKCIJA.PY – prikaz programskog koda unutar datoteke transakcija.py

8.4. PRIKAZ ISPISANIH DATOTEKA NAKON POKRETANJA PROGRAMA

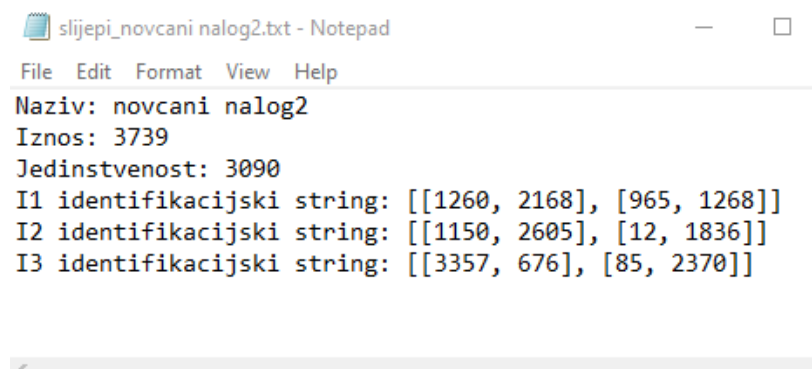
Slijedi prikaz ispisanih datoteka za prethodno objašnjeni programski kod. Prikazane datoteke se kreiraju nakon pokretanja programa preko komande: **python transakcija.py**

8.4.1. Slijepi novčani nalozi



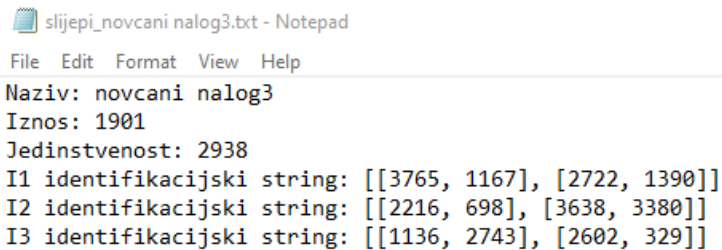
```
slijepi_novcani_nalog1.txt - Notepad
File Edit Format View Help
Naziv: novcani nalog1
Iznos: 3505
Jedinstvenost: 3623
I1 identifikacijski string: [[3832, 1601], [2850, 1513]]
I2 identifikacijski string: [[3756, 723], [1173, 767]]
I3 identifikacijski string: [[1016, 3218], [1159, 1706]]
```

Slika 22. slijepi_novcani_nalog1



```
slijepi_novcani_nalog2.txt - Notepad
File Edit Format View Help
Naziv: novcani nalog2
Iznos: 3739
Jedinstvenost: 3090
I1 identifikacijski string: [[1260, 2168], [965, 1268]]
I2 identifikacijski string: [[1150, 2605], [12, 1836]]
I3 identifikacijski string: [[3357, 676], [85, 2370]]
```

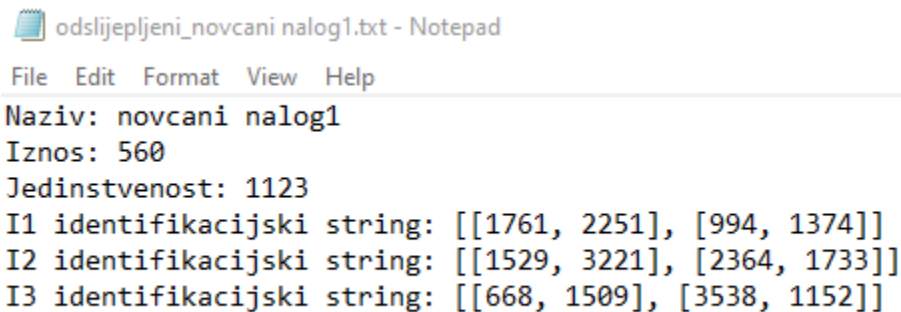
Slika 23. slijepi_novcani_nalog2



```
slijepi_novcani nalog3.txt - Notepad
File Edit Format View Help
Naziv: novcani nalog3
Iznos: 1901
Jedinstvenost: 2938
I1 identifikacijski string: [[3765, 1167], [2722, 1390]]
I2 identifikacijski string: [[2216, 698], [3638, 3380]]
I3 identifikacijski string: [[1136, 2743], [2602, 329]]
```

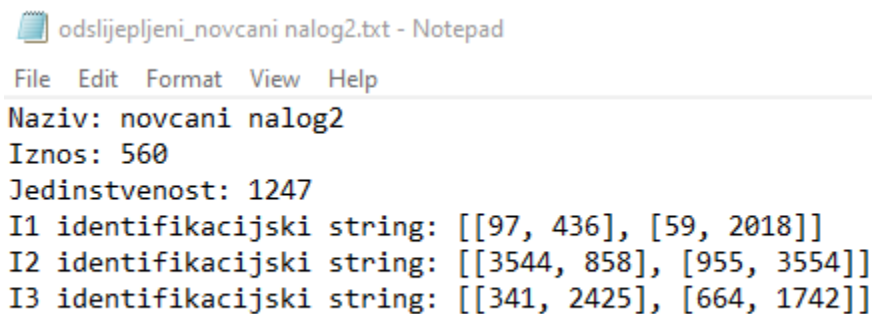
Slika 24. slijepi_novcani nalog3

8.4.2. Odslijepljeni novčani nalozi



```
odsljepljeni_novcani nalog1.txt - Notepad
File Edit Format View Help
Naziv: novcani nalog1
Iznos: 560
Jedinstvenost: 1123
I1 identifikacijski string: [[1761, 2251], [994, 1374]]
I2 identifikacijski string: [[1529, 3221], [2364, 1733]]
I3 identifikacijski string: [[668, 1509], [3538, 1152]]
```

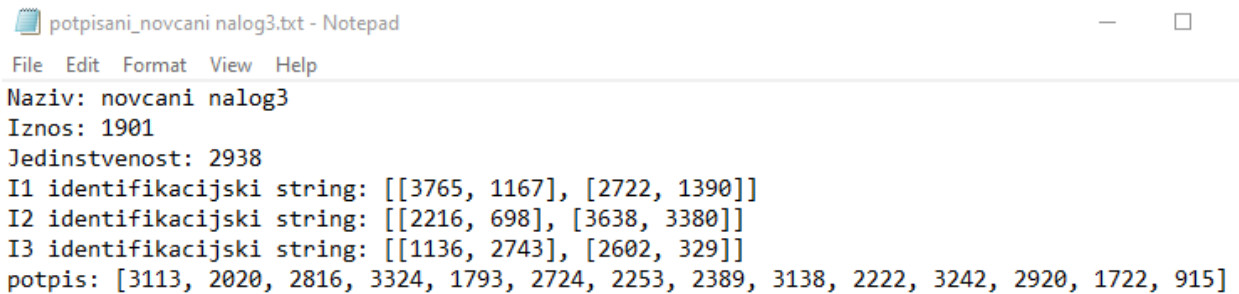
Slika 25. odsljepljeni_novcani nalog1



```
odsljepljeni_novcani nalog2.txt - Notepad
File Edit Format View Help
Naziv: novcani nalog2
Iznos: 560
Jedinstvenost: 1247
I1 identifikacijski string: [[97, 436], [59, 2018]]
I2 identifikacijski string: [[3544, 858], [955, 3554]]
I3 identifikacijski string: [[341, 2425], [664, 1742]]
```

Slika 26. odsljepljeni_novcani nalog2

8.4.3. Potpisani novčani nalog



potpisani_novcani_nalog3.txt - Notepad

File Edit Format View Help

Naziv: novcani_nalog3
Iznos: 1901
Jedinstvenost: 2938
I1 identifikacijski string: [[3765, 1167], [2722, 1390]]
I2 identifikacijski string: [[2216, 698], [3638, 3380]]
I3 identifikacijski string: [[1136, 2743], [2602, 329]]
potpis: [3113, 2020, 2816, 3324, 1793, 2724, 2253, 2389, 3138, 2222, 3242, 2920, 1722, 915]

Slika 27. potpisani_novcani_nalog3

8.4.4. Odslijepljeni potpisani novčani nalog



odslijepljeni_potpisani_novcani_nalog3.txt - Notepad

File Edit Format View Help

Naziv: novcani_nalog3
Iznos: 560
Jedinstvenost: 2189
I1 identifikacijski string: []
I2 identifikacijski string: []
I3 identifikacijski string: []
potpis: [3181, 3084, 2224, 436, 212, 3640, 838, 420, 1121, 310, 1028, 2131, 2980, 1664]

Slika 28. odslijepljeni_potpisani_novcani_nalog3

9. LITERATURA

1. B. Schneier, Applied Cryptography Second edition, John Wiley & Sons Inc., 1996.
2. AL-Matari, M. Hani, Anonymous and Non-Repudiation E-Cash Scheme Based on Partially Blind Signature, izvor: https://meu.edu.jo/libraryTheses/5873642b9f759_1.pdf
3. CARNet, Elektronički novac, NCERT-PUBDOC-2010-09-311, izvor: https://www.cis.hr/www.edicija/LinkedDocuments/NCERT-PUBDOC-2010-09-311.pdf?fbclid=IwAR3i-rTmrmqulcQX2PXaWE2dGhnx9_zG5emp_H4Unp9aX5Ax1Ke5MbmmA2c
4. Delfs Hans, Helmut Knebl, Inrouduction to Cryptography, second edition 2007.
5. Henk C.A. van Tilbork, Encyclopedia of cryptography and security. Izvor: <http://202.91.76.90:81/fdScript/RootOfEBooks/GENERAL/Encyclopedia%20of%20Cryptography%20and%20Security.pdf>
6. Digital Currencies' Role in Facilitating Ransomware Attacks: A Brief Explainer, 2021. Izvor: <https://www.thirdway.org/memo/digital-currencies-role-in-facilitating-ransomware-attacks-a-brief-explainer?fbclid=IwAR3GEaZKa6B5jcu6UvdKYyUY4dcaJfGKOfd9NgaHg8eCPMVTjmQ5IDCna4w>
7. Lake Josh, Common encryption types, protocols and algorithms explained, 2020. Izvor: <https://www.comparitech.com/blog/information-security/encryption-types-explained/?fbclid=IwAR1BuG1alzPbAKEcU21t56QZBIAJSdoCvtRUsoC9ET7E3xYf4IbKiAtsZ7M>
8. Science Direct, Blind Signature, izvor: https://www.sciencedirect.com/topics/computer-science/blind-signature?fbclid=IwAR3i-rTmrmqulcQX2PXaWE2dGhnx9_zG5emp_H4Unp9aX5Ax1Ke5MbmmA2c

10. POPIS SLIKA

Slika 1. Osnovni koraci plaćanja elektroničkim novcem

Slika 2. Osnovni protocol

Slika 3. Konačni oblik protokola plaćanja elektroničkim novcem

Slika 4. KLIJENT.PY – importanje library-a i kreiranje klase klijent

Slika 5. KLIJENT.PY – prikaz funkcije za obvezivanje bitovima

Slika 6. KLIJENT.PY – prikaz funkcije procesa zasljepljivanja novčanog naloga

Slika 7. KLIJENT.PY – prikaz funkcije za kreiranje stringa identiteta

Slika 8. KLIJENT.PY – prikaz funkcije za kreiranje novčanog naloga

Slika 9. KLIJENT.PY – prikaz funkcije za ispis novčanog naloga

Slika 10. KLIJENT.PY – prikaz funkcije za generiranje slučajnog broja

Slika 11. KLIJENT.PY – prikaz funkcije za primanje potpisa

Slika 12. KLIJENT.PY – prikaz funkcije za razotkrivanje odabranih novčanih naloga

Slika 13. KLIJENT.PY – prikaz funkcije za razdijeljivanje tajne

Slika 14. KLIJENT.PY – prikaz funkcije za odslijepljivanje novčanih naloga

Slika 15. KLIJENT.PY – prikaz funkcije za odslijepljivanje potpisanih novčanih naloga

Slika 16. BANKA.PY – prikaz kreiranja klase Banka i funkcije za za hash izračun

Slika 17. BANKA.PY – prikaz funkcije za verificiranje informacija

Slika 18. BANKA.PY – prikaz funkcija za dohvaćanje

Slika 19. BANKA.PY – prikaz funkcije za potpis novčanog naloga

Slika 20. BANKA.PY – prikaz funkcije za odslijepljivanje zahtjevanih novčanih naloga

Slika 21. TRANSAKCIJA.PY – prikaz programskog koda unutar datoteke transakcija.py

Slika 22. slijepi_novcani_nalog1

Slika 23. slijepi_novcani_nalog2

Slika 24. slijepi_novcani_nalog3

Slika 25. odslijepljeni_novcani nalog1

Slika 26. odslijepljeni_novcani nalog2

Slika 27. potpisani_novcani nalog3

Slika 28. odslijepljeni potpisani_novcani nalog3