Usporedba tehnologija za poboljšanje privatnosti na javnim blockchain mrežama

Tomac Orlić, Dora

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

USPOREDBA TEHNOLOGIJA ZA POBOLJŠANJE PRIVATNOSTI NA JAVNIM BLOCKCHAIN MREŽAMA

Rijeka, studeni 2022.

Dora Tomac Orlić

0069074441

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

USPOREDBA TEHNOLOGIJA ZA POBOLJŠANJE PRIVATNOSTI NA JAVNIM BLOCKCHAIN MREŽAMA

Mentor: Prof. dr. sc. Kristijan Lenac

Rijeka, studeni 2022.

Dora Tomac Orlić

0069074441

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za računarstvo**
Predmet: **Napredni operacijski sustavi**
Polje: **2.09 Računarstvo**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Dora Tomac Orlić (0069074441)**
Studij: Diplomski sveučilišni studij računarstva
Modul: Računalni sustavi

Zadatak: **Usporedba tehnologija za poboljšanje privatnosti na javnim blockchain mrežama / Comparison of privacy enhancement technologies in public blockchains**

Opis zadatka:

Proučiti i opisati probleme očuvanja privatnosti, povjerljivosti i anonimnosti prilikom korištenja blockchain tehnologija. Analizirati i usporediti tehnike za poboljšanje privatnosti na blockchainu.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Tomac Orlić
Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Prof. dr. sc. Kristijan Lenac

Predsjednik povjerenstva za diplomski ispit:

Prof. dr. sc. Kristijan Lenac

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradila ovaj rad.

Rijeka, studeni 2022.

_____

Dora Tomac Orlić

Sadržaj

1. UVOD

Živimo u vremenu kada se osobni podaci korisnika mogu prikupiti na brojnim mjestima na kojima se prije nisu mogli prikupiti, kao što su društvene mreže, pametni uređaji, pametne kuće i sva ostala mjesta na kojima su pohranjeni digitalni podaci. Korisnici nesvjesno gube kontrolu nad svojim osobnim podacima, ni ne znajući gdje ih sve ostavljaju. U životu koji smo živjeli prije Interneta i digitalnog svijeta, bilo je nezamislivo nekome nepoznatome otkriti svoje podatke, radilo se o imenu, prezimenu, godinama, mjestu stanovanja, financijama, obitelji ili o bilo kojem drugom obliku osobnog podatka. Sada čovjek nije više ni svjestan što sve spada u osobne podatke, gdje ih sve otkriva, tko sve može doći do tih podataka niti na koji način ih netko može iskoristiti. Izreka koja je vezana uz privatnost na Internetu govori da ako ne plaćate proizvod, vi ste tada proizvod. To znači da su tvrtkama koje nude besplatne usluge, kao što su platforme za društveno umrežavanje, osobni podaci o korisnicima vrijedni i bitni. Može se očekivati da će eksponencijalne stope rasta u stvaranju i prikupljanju podataka nastaviti narušavati našu privatnost. Sve navedeno ukazuje na to da je jedan od najvećih problema današnjice u svijetu Interneta, mreža, uređaja, tehnologija pa tako i blockchain tehnologije privatnost.

Javni blockchain je otvorena, decentralizirana knjiga koja bilježi transakcije i blokove podataka koje dijele mrežni čvorovi, odnosno korisnici. Cilj javnog blockchaina je stvoriti infrastrukturu koja omogućuje korisnicima da vjeruju zajedničkoj digitalnoj knjizi podataka. Iako je blockchain tehnologija u kojoj se krije veliki potencijal korištenja u širokom spektru primjene, ima veliki problem nedostatka ozbiljnog mehanizma zaštite privatnosti. Korisnici blockchaina imaju probleme s privatnošću i anonimnošću zbog samog dizajna i svojstava te tehnologije. Primjerice, zbog transparentnosti blockchain tehnologije, svi zapisi podataka i transakcije su javno evidentirani te ih je moguće pregledavati. Javni blockchain omogućuje svakom čvoru da vidi iznose transakcija i adrese koje su u njih uključene, što dovodi do otvaranja vrata zlonamjernicima u narušavanju korisnikove privatnosti. Osim transparentnosti, decentralizacija, distribuiranost i nepromjenjivost su također svojstva blockchain tehnologije koje samo prividno pružaju zaštitu privatnosti.

Iz navedenoga, zaključuje se da nove tehnologije javne blockchain mreže moraju biti sposobne omogućiti veću privatnost u odnosu na prethodne tehnologije. Nove i buduće blockchain tehnologije bi trebale imati fokus na poboljšanju onih karakteristika i problema mreže koje mogu korisnike dovesti do ugrožavanja njihove privatnosti.

Struktura ovog rada je napravljena tako da se čitatelja u drugom poglavlju ponajprije upozna sa svojstvima blockchaina te s temeljnim načelima blockchaina, budući da će navedeno biti puno puta spominjano u radu. Dalje, u trećem poglavlju biti će opisani i analizirani problemi očuvanja privatnosti, povjerljivosti i anonimnosti u svijetu blockchain tehnologije, dok u četvrtom poglavlju slijedi opis i analiza tehnologija za poboljšanje privatnosti na javnim blockchain mrežama. Na kraju, prije samog zaključka, u petome poglavlju biti će predstavljena tablica usporedbe navedenih i opisanih tehnologija za poboljšanje privatnosti na javnim blockchain mrežama.

2. SVOJSTVA I TEMELJI BLOCKCHAIN TEHNOLOGIJE

Blockchain je distribuirana baza podataka ili knjiga koja se dijeli između čvorova računalne mreže. Kao baza podataka, blockchain pohranjuje informacije elektronički, u digitalnom formatu. Blockchain je tehnologija koja bi potencijalno mogla ograničiti narušavanje privatnosti korisnika, budući da korisnici mogu objavljivati osobne podatke onda kada su oni potrebni. Primjerice, korisnik može pohraniti osobne podatke na blockchain, a privremeno objaviti dijelove tih podataka, bez da itko ima pristup podacima osim njega, po potrebi. Sustavi s blockchain tehnologijom bi u budućnosti mogli pomoći pojedincima, ali i organizacijama da vrate kontrolu nad svojim podacima.

Kako i svakoj drugoj tehnologiji, tako se i prilikom korištenja blockchain tehnologije može naići na probleme. Kako bi shvatili probleme na koje korisnici blockchaina nailaze u smislu privatnosti, povjerljivosti i anonimnosti te svrhu i važnost postojanja tehnologija za poboljšanje privatnosti prilikom korištenja blockchain javne mreže, potrebno je prethodno pojasniti svojstva i bitna temeljna načela blockchaina, koji će u daljnjem nastavku rada biti mnogo puta spomenuti.

2.1. Decentralizacija

Centralizirani sustavi pohrane su sustavi u kojima su podaci i sve informacije pohranjene na jednom mjestu i u kojima jedna osoba, organizacija ili entitet ima kontrolu nad tim podacima. Takvi sustavi imaju samo jednu točku kvara - što ako netko, primjerice, slučajno izbriše podatke? Oni zauvijek nestaju. Također, kada bi sve baze podataka na Internetu bile centralizirane, odnosno kada bi pripadale nekolicini korporacija, te ukoliko bi u tom slučaju došlo do napada na centralni poslužitelj, moglo bi se doći do ogromnih količina korisničkih podataka.

Blockchain je suprotno od toga, decentraliziran sustav. U takvom sustavu se podaci nalaze na mnogo mrežnih čvorova na različitim lokacijama. Blockchain se kopira i širi mrežom računala. Kada se doda novi blok u blockchain, svako računalo na mreži ažurira svoj blockchain. Širenjem informacija i blokova po mreži, umjesto pohranjivanjem u jednu središnju bazu podataka, blockchain postaje teže hakirati. Ukoliko bi haker htio promijeniti blockchain, kada bi mijenjao vlastitu kopiju, ona se ne bi više usklađivala s kopijom svih ostalih. Kada bi ostali

uspoređivali svoje kopije jedni s drugima, uočili bi da se kopija od hakera ističe i ona bi tada bila odbačena kao nelegitimna. Haker bi uspio u svom naumu tek kada bi promijenio 51% ili više kopija blockchaina, no takav napad zahtijeva veliku količinu resursa. [1]

Kod decentraliziranog sustava ne postoji jedna točka kvara, što u nekim slučajevima, kao što je primjer naveden u odlomku iznad, pogoduje očuvanju privatnosti i povjerljivosti prilikom korištenja blockchain tehnologije. [1] Iako decentraliziranost blockchain tehnologije donosi neke prednosti, od kojih su neke navedene iznad, ovo svojstvo može uvelike biti temelj za mnoge probleme i napade koji se događaju u blockchain tehnologiji, što će biti prikazano u idućem poglavlju.

2.2. Distribuiranost

Svojstvo distribuiranosti povezuje komponente tako da su one povezane na više umreženih računala. Ovo svojstvo se veže na decentralizaciju te se stvara sustav povjerenja u kojem svaki čvor ima jednaku razinu povjerenja i svi su čvorovi jednako važni. Čvorovi su povezani putem peer-to-peer mreže koja im omogućuje da kontroliraju svoje podatke, smanjujući prijetnju trećih strana da manipuliraju osobnim podacima. Slika 2.1. prikazuje razliku između centraliziranog, decentraliziranog i distribuiranog sustava.

Slika 2.1. Prikaz centraliziranog, decentraliziranog i distribuiranog sustava [2]

2.3. Nepromjenjivost

Blockchain je niz blokova povezanih u lanac pomoću jedinstvene i identifikacijske hash funkcije - adrese bloka. Kada se zapis u bloku zabilježi, njezinu autentičnost se mora potvrditi.

Nakon što je zapis potvrđen, novi se blok pohranjuje kronološki, odnosno dodaje se na kraj blockchaina. Nakon što je blok dodan na kraj blockchaina, teško je vratiti se i promijeniti sadržaj bloka, osim ako većina mreže nije postigla konsenzus da se to učini. Cilj blockchaina je omogućiti pohranjivanje i distribuciju digitalnih informacija, ali ne i uređivanje. Na temelju toga, blockchain je nepromjenjiva knjiga u kojoj se zapisi ne mogu mijenjati, brisati ili uništiti. Tome pomaže već spomenuta hash vrijednost bloka, hash prethodnog bloka i vremenska oznaka koje se nalaze u zaglavlju bloka. [1] Hash funkcija će biti detaljnije opisana u potpoglavlju 3.5.2.

2.4. Transparentnost

Sve transakcije i zapisi na blockchainu se mogu transparentno pregledavati, posjedovanjem osobnog čvora ili korištenjem blockchain istraživača. Svi mogu vidjeti sve transakcije koje se događaju na blockchainu i pratiti ih uživo, budući da svaki čvor ima kopiju lanca koja se ažurira pri potvrđivanju i dodavanju novih blokova. Iako korisnici mogu pristupiti pojedinostima o transakcijama, ne mogu pristupiti identifikacijskim informacijama o korisnicima koji vrše te transakcije. Pomoću ovog svojstva korisnik blockchaina uvijek može pratiti gdje se njegovi zapisi i informacije koriste. [1] Unatoč naizgled jako dobrom svojstvu blockchain tehnologije, transparentnost otvara vrata mnogim zlonamjernicima u narušavanju tuđe privatnosti.

2.5. Kriptografija

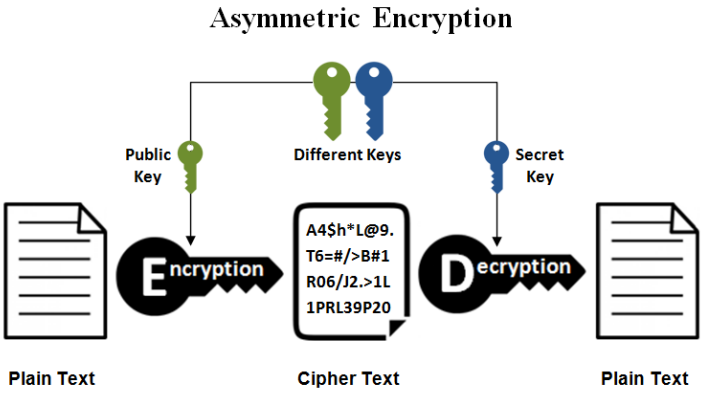
Blokovi podataka u blockchainu su povezani pomoću kriptografije, odnosno pomoću metode koja sprječava treću stranu da pristupi podacima. Kriptografija je metoda kojom se informacije i podaci pretvaraju u kod (šifriraju). Postoji kriptografija sa simetričnim ključem, kriptografija s asimetričnim ključem i hash funkcije. Blockchain koristi dvije vrste kriptografskih algoritama, a to su asimetrična kriptografija i hash funkcije. Kriptografija čini temelj blockchaina.

2.5.1. Asimetrična enkripcija

Asimetrična enkripcija koristi privatne i javne ključeve, jedan za šifriranje informacija, a drugi za dešifriranje. Javni ključ mogu vidjeti svi, a privatni ključ je tajna vrijednost. Ta dva ključa rade zajedno, poruka se šifrira javnim ključem, a dešifrira privatnim ključem. [3] Slika 2.2. vizualizira način rada asimetrične enkripcije. Na slici se vidi kako se otvoreni tekst enkriptira javnim ključem, čime se dobije šifrirani tekst, a za dekripciju se koristi tajni ključ, kako bi se ponovno dobio otvoren tekst.

U blockchainu, privatni ključevi se koriste za pokretanje transakcije, a javni za provjeru transakcije. Privatni ključ služi za digitalno potpisivanje transakcije prije emitiranja i za dekriptiranje poruka, a javni za šifriranje transakcije prije nego što se dogodi, a kasnije za dokazivanje da je digitalni potpis valjan. [4]

Asimetrična kriptografija se kod blockchaina koristi za upravljanje identitetom i autentifikaciju transakcija. Korisnici blockchaina ne moraju otkriti svoj pravi identitet kako bi stvorili račun na blockchainu ili ga koristili, već se računi identificiraju pomoću adresa koje su izvedene iz javnih ključeva. Primjerice, prilikom kreiranja transakcije na svom blockchain računu, korisnik mora digitalno potpisati transakciju svojim privatnim ključem. Digitalni potpis dokazuje da je netko tko poznaje privatni ključ računa izvršio sve transakcije povezane s tim računom. Nakon što je transakcija poslana ostatku mreže, svatko može provjeriti potpis s odgovarajućim javnim ključem, dokazujući da je transakciju autorizirao vlasnik računa. Taj proces omogućuje provjeru autentičnosti transakcije bez potrebe za otkrivanjem identiteta vlasnika računa te se time na neki način štiti korisnikova anonimnost prilikom korištenja blockchaina. [4]



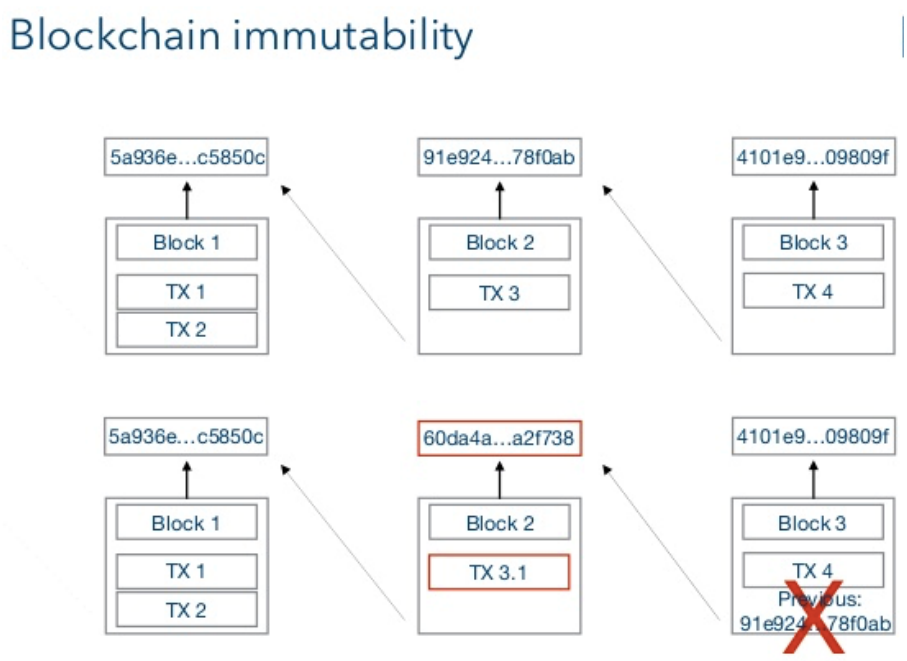
Slika 2.2. Način rada asimetrične enkripcije [5]

2.5.2. Hash funkcija

Hash funkcija je funkcija koja može bilo koju informaciju otvorenog teksta pretvoriti u jedinstveni, nerazumljivi niz teksta. Hash kodovi stvaraju se matematičkom funkcijom koja digitalne informacije pretvara u nasumični niz brojeva i slova. Hash funkcije su jedinstvene - svaki ulaz ima jedinstven izlaz i brze - lako se mogu generirati u kratko vrijeme. Hashevi su uvijek iste duljine, bez obzira na duljinu podataka. Također, inverz te funkcije nije moguć, odnosno ne možemo generirati ulaz ako imamo izlaz i hash funkciju. [3]

Nakon što podaci prođu kroz hash funkciju, proces se ne može poništiti. Po tome se raspršivanje (engl. hashing) razlikuje od asimetrične enkripcije, gdje se podaci mogu dešifrirati pomoću ključa. Ne postoji način otkrivanja izvornih podataka. [3]

Hash funkcija se koristi kako bi svi sudionici blockchaina vidjeli prikaz blockchaina. Također, ove funkcije imaju glavnu ulogu u povezivanju blokova i održavanju integriteta podataka u blokovima. Budući da će isti podaci uvijek postati isti hash, korisnici mogu usporediti podatke s konačnim hashom i otkriti je li netko imao neovlašten pristup podacima te jesu li oni promijenjeni. Svaka izmjena podataka u bloku invalidira lanac i čini ga nevažećim. [3]



Slika 2.3. Otkrivanje promjene u bloku pomoću hash-a [6]

Najčešće korišteni algoritam raspršivanja u blockchainu je SHA-256 koji sažima podatke u veličinu 256-bitnog stringa, neovisno o tome kolike duljine su podaci.

2.6. Mehanizmi konsenzusa

Kod centraliziranih sustava, kao što je banka, postoji glavno računalo, poslužitelj, kojem se povjerava knjiga transakcija. Banka vjeruje tom računalu i nema problema sa sigurnošću ili s integritetom računala. Zbog svojstva decentraliziranosti u blockchain tehnologiji, transakcije i pohranjeni podaci u blokovima se distribuiraju na mnogo čvorova u mreži. S obzirom na to, postoji potreba za postavljanjem pravila koja će osigurati sigurnost i integritet podataka kako bi se spriječili hakerski napadi i ugrožavanje korisnikove privatnosti i povjerljivosti podataka. Stoga se za provjeru valjanosti zapisa u bloku koriste mehanizmi konsenzusa kao sporazumi potrebni da bi mreža ispravno radila, čak i u slučaju kvara. [7] Mehanizmi konsenzusa ključni su za rješavanje problema grananja lanca u kojem treba znati koji lanac odabrati kao pravi te za sprječavanje dvostruke potrošnje, odnosno za sprječavanje potrošnje iste digitalne valute više puta. Za verifikaciju zapisa u blokovima, većina računalne snage blockchain mreže bi trebala potvrditi valjanost. Kako bi se spriječili iznad navedeni problemi, kao i zlonamjerni akteri u, primjerice, potvrđivanju zlonamjernih transakcija koje bi korisnicima mogle oduzeti pravo povjerljivosti podataka, lanci blokova osigurani su mehanizmima konsenzusa, kao što su dokaz o radu (Proof of Work) ili dokaz o udjelu (Proof of Stake). [1]

Budući da blokove u kojima se bilježe informacije potvrđuju rudari putem mehanizama konsenzusa, nevažeće transakcije neće biti zabilježene u lancu blokova, a svaki pokušaj zlonamjernog pristupa nečijim podacima i pokušaju izmjene podataka u lancu blokova biti će odbijen. [8]

2.6.1. Proof of Work

Proof of Work (PoW) ili dokaz o radu je mehanizam konsenzusa kojeg koriste Bitcoin i Ethereum i u kojem čvorovi, zvani rudari, rješavaju matematički problem kako bi validirali i verificirali blok podataka. Matematički problem, zagonetka, se rješava metodom pokušaja i pogreške. Nakon što rudarski čvor pronađe rješenje, ostali čvorovi provjeravaju je li ono validno te ukoliko je, tada se blok dodaje u lanac, a rudar dobiva nagradu u obliku transakcijske

naknade. Ovaj konsenzus doprinosi sigurnosti, privatnosti korisnika i povjerljivosti podataka jer zahtjeva hakiranje najmanje 50% čvorova u mreži, što nije nemoguće, ali se teško postiže. [9]

2.6.2. Proof of Stake

Proof of Stake (PoS) ili dokaz o udjelu smanjuje količinu računalne snage koja je potrebna za provjeru blokova i transakcija. Proof of Stake koristi pseudo-slučajnu funkciju za odabir čvora validatora kojemu je dopušteno validirati blok na temelju uloga. Pseudo-slučajna funkcija se koristi kako bi se izbjegao scenarij u kojem su najbogatiji korisnici uvijek odabrani da potvrde transakciju i dobiju nagradu. Ulog određenog broja kovanica kojeg čvor ulaže određuje šanse da čvor postane validator za sljedeći blok. Što je veći ulog čvora, ima veće šanse da postane validator, budući da bi njegovo zlonamjerno djelovanje dovelo do većeg nazadovanja od nekoga tko je uložio manje. Ulog validatora djeluje kao kolateral koji se može uništiti ukoliko čvor ne validira ispravan blok ili se ponaša nepošteno, što pruža sigurnost da čvor nema poticaja za zlonamjerne radnje.

Za razliku od PoW, PoS pruža smanjene hardverske zahtjeve te samim time bolju energetsku učinkovitost. Također, pojedincima se olakšava sudjelovanje u osiguravanju mreže, budući da se čvor validatora može pokrenuti na običnom računalu. [9]

3. PROBLEMI OČUVANJA PRIVATNOSTI, POVJERLJIVOSTI I ANONIMNOSTI PRILIKOM KORIŠTENJA BLOCKCHAIN TEHNOLOGIJE

Kako bi opisali probleme očuvanja privatnosti, povjerljivosti i anonimnosti prilikom korištenja blockchain tehnologije, potrebno je prvo pojasniti pojmove privatnosti, povjerljivosti i anonimnosti.

Privatnost je stanje u kojem je pojedinac slobodan od javnog ometanja. Privatnost se odnosi na ljude i to je pravo koje se može povrijediti. Privatnost ograničava pristup javnosti osobnim podacima o osobi. [10] Također, to je mogućnost da ono što pojedinac radi na mreži zadrži za sebe i za one s kojima želi dijeliti svoju privatnost.

Povjerljivost je povjeravanje da vlastita informacija neće doći u pristup neovlaštenoj osobi. Povjerljivost se odnosi na podatke i njihovu privatnost. Osobni podaci korisnika se trebaju čuvati u tajnosti kako bi se spriječila krađa identiteta, ugrožavanje računa ili bitnih sustava, pravna šteta itd. Prilikom upravljanja povjerljivošću informacija treba uzeti u obzir kome se podaci mogu otkriti, pružaju li korištene mreže i tehnologije povjerenje da će podaci ostati neotkriveni, jesu li podaci po prirodi osjetljivi i koji bi bio negativan učinak kada bi se oni otkrili te bi li ti podaci bili vrijedni onima koji nemaju dopuštenje za pristup istima. [11]

Anonimnost je zaštita identiteta, primjerice, korištenjem pseudonima. Anonimizacija je proces koji na neki način mijenja podatke o pojedincu tako da se ne može izgraditi veza prema identitetu stvarnog pojedinca. Svrha tog procesa je da aktivnost identiteta prođe neopaženo. [12] U primjeru blockchaina, korisnik ostaje anoniman ako se njegove radnje ne mogu povezati s njegovim identitetom.

Kada govorimo o privatnosti i povjerljivosti osobnih podataka, idealno bi bilo kada bi za neku informaciju njezin vlasnik imao potpunu kontrolu nad njezinim širenjem i korištenjem. Naravno, u praktičnim primjenama to nije moguće garantirati zbog ograničenja tehnologija. Sudeći po tome, ni blockchain tehnologija sama po sebi nije idealna. Koliko neke karakteristike blockchaina, kao što su decentralizacija, nepromjenjivost ili transparentnost pomažu u očuvanju privatnosti, povjerljivosti i anonimnosti, toliko i odmažu, stoga je i blockchain ranjiv na kibernetičke napade i ugroze privatnosti. Zbog svojih karakteristika, blockchain je privukao pažnju, no s porastom i širokim usvajanjem ove tehnologije, povrede podataka postale su učestale. Ljudi sa zlonamjernim namjerama mogu iskoristiti sigurnosne

propuste blockchaina u svoju korist. Privatnost može biti narušena izravno - direktnim curenjem informacija koje narušavaju povjerljivost korisnikovih podataka ili neizravno - npr. praćenjem korisnikovih transakcija gdje zlonamjerni akteri analizama zaključuju koji je korisnikov identitet.

Kao što je već napomenuto, korisnici blockchaina se suočavaju s mnogobrojnim rizicima privatnosti koji ometaju njegovu praktičnu primjenu. Raznoliki problemi i nedostatci u očuvanju privatnosti, povjerljivosti i anonimnosti prilikom korištenja blockchaina biti će navedeni u ovom poglavlju na temelju kojih će kasnije biti predstavljene postojeće tehnike za poboljšanje privatnosti u blockchainu. Budući da je još uvijek najčešća primjena blockchaina kod trgovanja kriptovalutama, većina navedenih problema odnositi će se na njih, iako bi se principi deanonimizacije i praćenja, kao i nekih navedenih napada mogli odnositi na bilo koje zapise podataka u blockchainu, a ne samo na transakcije vezane uz kriptovalute.

3.1. Deanonimizacija korisnika

Budući da kriptovalute, koje su temeljene na blockchain tehnologiji, omogućuju izravne peer-to-peer transakcije putem interneta, ideja je da su samo dvije strane uključene u aktivnost. Iako se čini da ovo postavlja savršen okvir za privatnost i anonimnost, neki primjeri daju drugačiju sliku kripto transakcija. S jedne strane, za korisnike koji koriste blockchain tehnologiju se tvrdi da su anonimni, ali s druge strane blockchain je potpuno transparentan i korisnici se mogu pratiti. Anonimnost korisnika odlikuje se na način da može postojati blockchain adresa korisnika bez otkrivanja njegovog identiteta. Kao što je već navedeno u prethodnom poglavlju, korisnici obično koriste hash vrijednosti nasumično odabranih javnih ključeva kao identifikatore kako bi sakrili svoj stvarni identitet. Javnost može vidjeti da netko nekome šalje određeni iznos putem transakcije, ali bez informacija koje transakciju povezuju s bilo kime. Jedna osoba može imati više adresa, odnosno novi javni ključ za svaku transakciju i teoretski ne bi trebalo postojati ništa što bi te adrese povezale za osobu koja ih posjeduje. Problem je što je slanje i primanje virtualne valute kod kriptovaluta kao pisanje pod pseudonimom - ako se pseudonim nekog autora ikada poveže s njegovim identitetom, sve što je isti ikada napisao pod tim pseudonimom biti će povezano s njime. Tako se recimo transakcijama kod kriptovaluta, kao što su Bitcoin transakcije, može se ući u trag, budući da je još uvijek moguće povezati niz transakcija s pojedinačnim korisnikom na način da se, primjerice, prati tijek

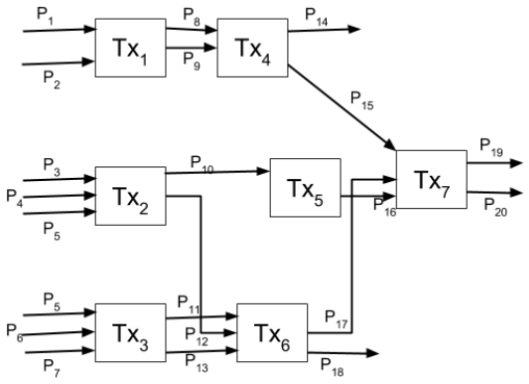
transakcijskog iznosa. Postoji nekoliko načina povezivanja adresa s identitetima, a najčešće se to radi putem analize blockchaina i promatranja načina na koji se transakcije prenose.

3.1.1. Klasteriranje adresa

Budući da blockchain uključuje sve transakcije sustava, jednostavna analiza daje informacije s kojih adresa dolazi novac i na koje adrese ide. Međutim, budući da korisnici u blockchain sustavu mogu kreirati bilo koji broj adresa, glavni cilj analize je klasterirati sve adrese u blockchainu koje pripadaju istom korisniku. Postoji mnogo takvih provedenih analiza koje se provode na različite načine, a primjer jedne analize klasteriranjem adresa koji se koristi u Bitcoin kriptovaluti prikazan je u ovome potpoglavlju.

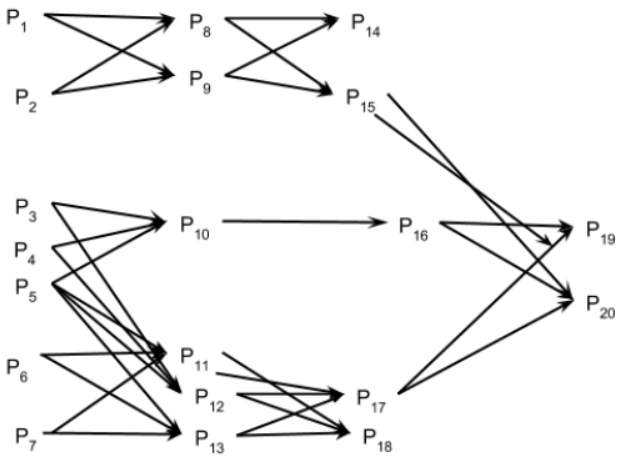
Postoje dvije heuristike koje su proizašle iz Bitcoin protokola, a na temelju kojih je izvedena ova analiza. Prva heuristika je ta da sve unose u transakciji generira isti korisnik, a druga povezuje ulazne adrese transakcije s njezinim izlaznim adresama, pretpostavljajući da ti izlazi mijenjaju adrese ako je izlazna adresa potpuno nova.

Prvi korak kod analize blockchaina je izrada transakcijskog grafa. Blockchain može biti promatran kao transakcijski graf Gt = {T, E}, gdje T označava skup transakcija na blockchainu, a E skup jednosmjernih rubova između tih transakcija. Gt je tijek bitcoin-a između transakcija u blockchainu tokom vremena. Skup ulaznih i izlaznih kriptovaluta u transakciji se može promatrati kao težina na rubovima Gt. Na slici 3.1. je prikazana instanca transakcijskog grafa za skup transakcija u blockchainu. [13]



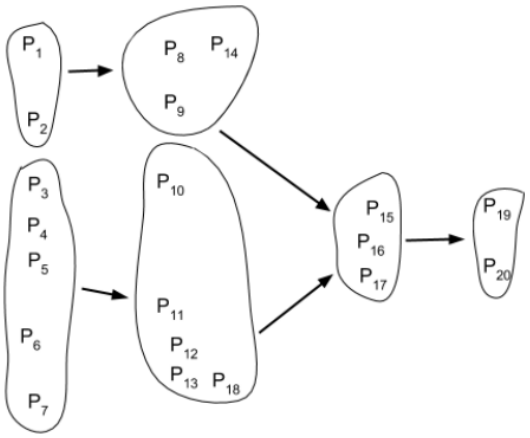
Slika 3.1. Transakcijski graf [13]

Drugi korak je izrada grafa adresa. Gledajući transakcijski graf, mogu se zaključiti odnosi između različitih ulaznih i izlaznih adresa. Graf adresa sadrži skup Ga = {P, E'}, gdje P označava skup adresa, a E' su rubovi koji povezuju te adrese. Na slici 3.2. je prikazan graf adresa koji je izveden iz prethodne slike, odnosno iz transakcijskog grafa. [13]



Slika 3.2. Graf adresa [13]

Zadnji korak je graf korisnika u kojem se, iz grafa adresa i niza zaključaka o protokolu bitcoin-a, grupiraju adrese koje izgledaju tako da pripadaju istom korisniku. Graf korisnika označen je skupom Ge = {U, E"}, gdje je U disjunktni podskup javnih ključeva (p), tako da p ∈ P, a E" su rubovi koji povezuju različite U kako bi se pokazala direktna povezanost između njih. Slika 3.3. prikazuje graf korisnika izveden iz slike 3.2. [13]



Slika 3.3. Graf korisnika [13]

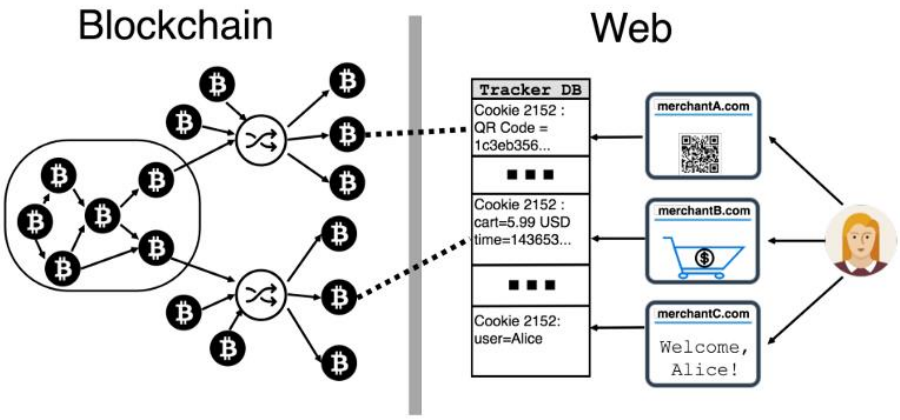
Korištenjem prve, iznad navedene heuristike, sve adrese javnih ključeva koje su bile ulazi u transakcije Tx2 i Tx3 (javni ključevi od 3 do 7) moraju pripadati istom korisniku, budući da transakcijski graf otkriva kako transakcije Tx2 i Tx3 imaju istu ulaznu adresu javnog ključa 5, dakle obje transakcije je trebao pokrenuti isti korisnik. [13]

Nadalje, vidi se da je adresa javnog ključa 14 grupirana s ostalim javnim ključevima koji su bili ulazi u transakciju Tx4 (javni ključevi br. 8 i 9). Ovo grupiranje odgovara drugoj heuristici jer je izlazna adresa potpuno nova, odnosno nikada se nije pojavila u povijesti Bitcoin blockchaina te se nikada neće ponovno koristiti na blockchainu.

3.1.2. Online kupovina

Ukoliko korisnik kupi proizvod i pri tome ga plati kriptovalutom, protivnik može jedinstveno identificirati transakciju na blockchainu korištenjem pratitelja trećih strana koji posjeduju dovoljno informacija o kupnji, kao što je kućna adresa korisnika. Nadalje bi se te transakcije mogle povezati s korisničkim kolačićima web preglednika koji omogućuju web stranicama da zapamte korisničke podatke za prijavu, košaricu za kupnju ili druge informacije, a onda i sa stvarnim identitetom korisnika, kao i s poviješću kupovine korisnika. [13]

Također, ako je zlonamjernik u mogućnosti povezati dvije kupnje istog korisnika na ovaj način, može identificirati cijeli korisnikov klaster Bitcoin adresa i transakcija na blockchainu, korištenjem standardnog softvera za praćenje i tehnika analize blockchaina. [13] Slika 3.4. prikazuje kako se može povezati korisnik i njegov identitet putem web kolačića. Na slici se vidi kako Alice kupuje i plaća Bitcoinom na stranicama merchantA.com i merchantB.com, a na merchantC.com se prijavljuje. Na prvoj stranici se pomoću kolačića odaje QR kod Bitcoin adrese, na drugoj se odaje iznos kupovine, a na trećoj stranici se odaje korisničko ime od Alice. Protivnik povezuje navedene kupovine na temelju kolačića te identificira Bitcoin novčiće koji odgovaraju kupnjama.



Slika 3.4. Povezivanje korisnika s identitetom putem kolačića [14]

3.1.3. Povezivanje IP adresa s korisnikom

Anonimnost korisnika na blockchainu može također biti ugrožena povezivanjem blockchain adrese s IP adresom i s korisnikom u peer-to-peer mreži. To se može dogoditi samim promatranjem i praćenjem mreže. Zlonamjernik koristi superčvor, koji predstavlja čvor s velikim brojem susjeda u P2P mreži te služi za poboljšanje vremena pretraživanja mreže, a povezuje se s aktivnim čvorovima i prati transakcijski promet povezan s poštenim čvorovima. Kako čvorovi simetrično šire transakcije po mreži, istraživanjem simetrične difuzije transakcija preko mreže moguće je povezati javne ključeve korisnika s njihovim IP adresama s točnošću od 30%. [13] Budući da je upotreba superčvora trivijalna i zahtijeva minimalno znanje o strukturi P2P mreže, može se pretpostaviti da bi se još veća točnost povezivanja IP adresa s korisnikom na ovaj način mogla postići korištenjem sofisticiranijih tehnika analize mrežnog prometa. [13]

Kao još jedan primjer, kod Bitcoin sustava, navodi se postojanje probabilističkog pristupa za povezivanje bitcoin adresa i transakcija s IP adresom inicijatora. U prvom koraku, propagirajuće poruke promatraju i snimaju nekoliko klijenata za praćenje kako bi se pokrio što veći dio mreže. Za svaku transakciju, klijenti koji prate bilježe popis klijenata koji su prenijeli transakciju u prvom vremenskom segmentu od 2 sekunde, budući da su ti klijenti mogući začetnici transakcije. Nakon teorijskih razmatranja, svakom klijentu dodjeljuju se vjerojatnosti koje pokazuju vjerojatnost da je on inicijator, zasebno za svaku transakciju koja je zabilježena. Zatim se grupiraju Bitcoin adrese u vlasništvu istog korisnika. Pri tome se iskorištava činjenica da Bitcoin adrese koje se pojavljuju na ulazu iste transakcije obično pripadaju istom korisniku.

Na kraju, postojanje nekoliko transakcija iste Bitcoin adrese i grupiranje Bitcoin adresa po korisniku omogućuje kombiniranje mjerenja iz više transakcija kako bi se s većom pouzdanošću identificirali korisnici. Kombiniranjem vjerojatnosti iz prvog koraka, korisnici (i njihova stanja) se uparuju s klijentima koji su najvjerojatnije inicijatori njihovih transakcija. Za izračun konačnih vjerojatnosti koristi se Bayesov model klasifikacije. Klijenti se mogu geografski lokalizirati putem svojih IP adresa, što omogućuje određivanje geografske distribucije i protoka Bitcoina. [15]

3.1.4. Praćenje novčanog tijeka

Na temelju otvorenosti i transparentnosti, lančana struktura blockchaina omogućuje da je svaka transakcija sustava slijediva. Bitcoin koristi način transakcije Unspent Transaction Output (UTXO). Transakcija može imati više ulaza i više izlaza. Trenutni ulaz transakcije je izlaz prethodne transakcije, a trenutni izlaz transakcije je ulaz za sljedeću transakciju. Prema korelaciji adrese transakcije, napadač može pratiti transakciju i dobiti novčani tijek. [8]

Korištenjem web stranica, kao što su Bitcoin Forum ili Twitter, može se dobiti adresa javnog ključa korisnika, a potom pratiti izvor i korištenje korisnikovih sredstava te izračunati stanje korisnika. [8]

3.2. Problemi korisničkih digitalnih novčanika i upravljanja ključevima

Kod kriptovaluta, koje su najupotrebljivija primjena blockchaina, svaki korisnik posjeduje skup privatno-javnih ključeva za pristup svom računu ili novčaniku. Digitalni potpis, koji se koristi za potpisivanje i potvrđivanje transakcija, je enkripcija hash funkcije transakcije koja je izračunata s privatnim ključem. Taj potpis dokazuje da transakcija nije izmijenjena i da ju je izdao vlasnik privatnog ključa. [16] Budući da su privatni ključevi ključni za potpisivanje svake transakcije te samim time za privatnost korisnika, gubitak ili krađa privatnih ključeva blokira korisnike u obavljanju transakcija s imovinom povezanom s njihovim privatnim ključem, stoga je potrebno primijeniti odgovarajuće sustave upravljanja ključevima kako ne bi došlo do curenja informacija ili krađe identiteta. [17]

Blockchain novčanici održavaju i pohranjuju ključeve povezane s vlasnikom novčanika, no novčanici su podložni krađama u kojima zlonamjernik može izbrisati ili ukrasti privatne ključeve korisnika.

Krađe novčanika se uglavnom izvode pomoću mehanizama za hakiranje sustava, instalacije softvera s greškom ili nepravilnog korištenja novčanicima. Neki od problema i prijetnji privatnosti pri korištenju digitalnih novčanika i upravljanju ključevima biti će navedeni u nastavku. [13]

3.2.1. Ranjivost digitalnog potpisa

Temelj provjere autentičnosti u blockchainu je privatni ključ. Za digitalni potpis se najčešće koristi algoritam eliptičnih krivulja, poznat kao ECDSA (Elliptic Curve Digital Signature Algorithm) algoritam. Problem tog algoritma je što nema dovoljno slučajnosti u generiranju potpisa. Naime, u procesu stvaranja potpisa koristi se unaprijed odabrana slučajna vrijednost zajedno s privatnim ključem. Ta se vrijednost koristi za izračun slučajne točke na eliptičnoj krivulji, čija se x koordinata koristi u izračunu potpisa i jedna je od dvije vrijednosti poslanih kao dio digitalnog potpisa. Također, ta slučajna vrijednost bi trebala biti različita za svaku transakciju. Zbog nedovoljnog broja slučajnih vrijednosti, događa se situacija u kojoj više javnih ključeva koristi istu slučajnu vrijednost u više od jednog potpisa, što omogućuje izračunavanje privatnih ključeva korisnika na temelju kojih zlonamjernik ima potpuni pristup i kontrolu nad blockchain računom korisnika. Zbog navedenog lošeg svojstva ovog algoritma za digitalni potpis, osiguravanje privatnih ključeva s drugim algoritmima su otvoreni izazov. [13]

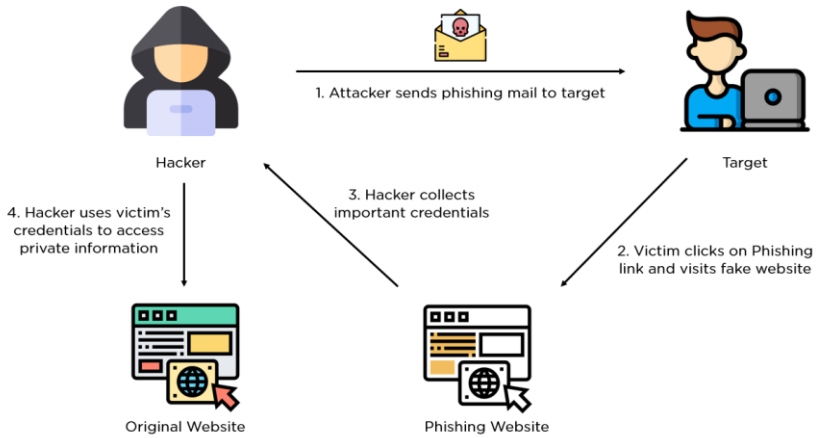
3.2.2. Mogući napadi preslika i sudara

Ukoliko hash funkcija, o kojoj smo se upoznali u prethodnim poglavljima, nije sigurna, ona može biti ranjiva na napade preslika i sudara. Postoje dvije vrste napada preslike, prvi je onaj u kojem napadač može pronaći izvorni unos iz hash-a, a drugi je onaj u kojem se napadaču daje jedan unos, a on pronalazi odgovarajući unos koji rezultira istim hash-om. [18] Napad sudara je napad u kojem napadač pokušava pronaći dva ulaza koji proizvode isti hash.

Iako bi u teoriji hash funkcije ove situacije trebale biti nemoguće, s nezamislivo ogromnom količinom računalne snage i naprednom analizom blockchaina, napadač bi mogao izvesti ovaj napad i pomoću njega otkriti adresu korisnika ili napraviti krađu korisničkih kovanica. [13]

3.2.3. Phishing napadi

Phishing napadi su napadi u kojima su pojedinci često mete pokušaja krađe identiteta. Cilj zlonamjernika je ukrasti korisnikove vjerodajnice. Zlonamjernici mogu poslati e-poruke legitimnog izgleda vlasniku ključa novčanika koja sadrži link na lažnu web stranicu. Nakon što žrtva klikne na poveznicu i unese podatke za prijavu, napadač ih zlokoristi za pristup svom računu i time krade identitet korisnika. [19] Navedeni tijek napada prikazan je i skiciran na slici 3.5. Prvi korak na slici je onaj u kojem napadač šalje phishing mail žrtvi, dok u drugom koraku žrtva otvara phishing link i posjećuje lažnu web stranicu. Zatim u idućem koraku napadač sakuplja potrebne vjerodajnice te koristi iste kako bi pristupio privatnim informacijama.



Slika 3.5. Phishing napad [20]

Ovakav napad pretrpjela je kripto tvrtka bZx, u kojem je haker 2021. godine ukrao milijune u raznim valutama, nakon što je jedan od njenih programera nasjeo na phishing napad. Procjena ukupno ukradenog iznosa je 55 milijuna dolara. Povreda je započela s phishing e-poštom poslanom na osobno računalo programera, koja je imala zlonamjernu makronaredbu u Word dokumentu koja je bila prerušena u legitimni privitak e-pošte, koji je potom pokrenuo skriptu

na njegovom osobnom računalu. To je dovelo do kompromitacije njegove osobne mnemoničke fraze novčanika. [21]

3.2.4. Bugovi i zlonamjerni softveri

Digitalni novčanik može biti napravljen kao obična aplikacija otvorene platforme. Aplikacije koje se temelje na blockchainu su softver koji razvijaju ljudi, a ljudi također mogu pogriješiti u kodiranju aplikacija. Te greške u kodiranju stvaraju kanale za prijetnje blockchain aplikacija. [22]

Greške se mogu pronaći i u, primjerice, hardverskim novčanicima koji izvanmrežno pohranjuju kriptovalute. Napad na njih bi mogao dovesti do otkrivanja svih privatnih ključeva u novčanicima.

Zlonamjerni softveri su također vrlo popularni pri pokušajima pristupa i krađe tuđih podataka. Primjer jednog takvog softvera je ElectroRat, napravljen za više operacijskih sustava, kojim napadači pokušavaju prikupiti privatne ključeve žrtava za pristup novčanicima. Stranice za preuzimanje zlonamjernih aplikacija su stvorene samo za prigodu napada i dizajnirane tako da izgledaju kao legitimne. Korištenjem lažnih društvenih medija i korisničkih profila te oglašavanjem na forumima za kriptovalute i blockchain tehnologiju, napadači promoviraju te zlonamjerne aplikacije i potiču korisnike da preuzmu aplikacije koje su u stvarnosti zlonamjerni softver te služe za prikupljanje osobnih podataka i privatnih ključeva. [23]

3.2.5. Rizici autentifikacije korisnika

Rizici autentifikacije korisnika su uzrokovani nedostatkom kontrola za sigurnost oko lozinke i tijeka autentifikacije. Ukoliko se dogodi da korisnik svoj uređaj s instaliranim novčanikom ostavi bez nadzora, netko bi mogao pokušati pogoditi korisnikovu lozinku koristeći grubu silu.

3.3. Problemi kod pametnih ugovora

Pametni ugovori, kao algoritmi koji upravljaju automatizacijom ugovora, ne dopuštaju posredničkim uslugama da izvrše transakciju, već se pokreću automatski kada sve uključene

strane ispune određene uvjete. Oni su samoizvršavajući kod koji koristi uvjetne naredbe koje pokreću radnju za izvođenje nekog zadatka. Na temelju navedenog o pametnim ugovorima, logika bi trebala nalagati da će se izostankom treće strane kao posrednika u ugovorima povećati privatnost, sigurnost i povjerljivost podataka iz istih. Ipak, u pametnim ugovorima također postoji problem narušavanja privatnosti i povjerljivosti podataka, gdje zlonamjerni korisnici mogu ometati blockchain aplikaciju kako bi otkrili stanje ugovora.

Postoje alati za analizu pametnih ugovora, kao što je Oyente, koji bi mogli pomoći u analizi ranjivosti pametnih ugovora. Također, pametni ugovori se za bolju privatnost mogu izvoditi u pouzdanom izvršnom okruženju, Intel SGX. [17]

Neki od napada na pametne ugovore, kao i ranjivosti koje dovode do ugroze privatnosti i povjerljivosti su opisane u idućim potpoglavljima.

3.3.1. Ponovni ulazak

Ponovni ulazak, kao tehnika napada u pametnom ugovoru, može uništiti ugovor ili ukrasti vrijedne informacije iz njega. Do ovog napada može doći kada funkcija pozove drugi ugovor putem vanjskog poziva. Napadač izvršava rekurzivni povratni poziv glavne funkcije, stvarajući nenamjernu petlju koja se ponavlja mnogo puta. Na primjer, kada ranjivi ugovor sadrži funkciju opoziva, ugovor može nezakonito pozivati funkciju opoziva više puta kako bi se iscrpio sav raspoloživi saldo koji ugovor sadrži. [24]

3.3.2. Napad neovlaštenim pristupom

Napad neovlaštenim pristupom se događa zbog neuspjeha u eksplicitnoj vidljivosti funkcije ili neuspjeha u provođenju provjera dopuštenja, što može uzrokovati napadačev pristup izmjeni funkcije ili varijable kojoj ne bi smio pristupiti. Također, zbog nedostatka ili nedovoljne kontrole pristupa, zlonamjernici mogu pozvati funkciju samouništenja kako bi uništile ugovor i time stanje u uništenom ugovoru prebacili na neovlašteni račun. Inače funkcija samouništenja omogućuje programerima uklanjanje pametnih ugovora iz Ethereuma kada se dogodi napad, no s druge strane, ta funkcija može povećati složenost razvoja i otvoriti mogućnost napada. [25]

3.4. Napadi na mrežu

Unutar blockchain tehnologije postoji, već spomenuta, peer-to-peer mreža, koja uključuje sve čvorove koji održavaju i pokreću blockchain protokole te pružaju usluge. Za zaključiti je da postoje različiti napadi koji se mogu dogoditi na tom mrežnom sloju blockchain tehnologije. Ne uzrokuju svi napadi koji postoje na mrežnom sloju ugrozu na razini privatnosti, povjerljivosti ili anonimnosti, stoga će biti navedeni samo neki od napada koji ugrožavaju navedeno, kako bi se ostalo u okviru zadane teme rada.

3.4.1. Napadi usmjeravanjem

Napadi usmjeravanja mogu utjecati na pojedinačne čvorove ili ciljati cijelu mrežu. To su napadi na razini Internet Service Provider-a (ISP), odnosno pružatelja internetskih usluga, kako bi se utjecalo na sudjelovanje u web sustavu, kao što je blockchain. Ovdje spada BGP (Border Gateway protocol) otmica u kojem ISP preusmjerava internetski promet oglašavanjem lažnih najava u sustavu usmjeravanja na internetu. [13, 16]

Blockchain mreža i aplikacije oslanjaju se na kretanje ogromnih količina podataka u stvarnom vremenu. Zlonamjernici mogu, primjerice, koristiti anonimnost računa za presretanje podataka dok se prenose davateljima internetskih usluga i na taj način ukrasti tuđe podatke ili kriptovalute. Također, ovaj napad bi mogle iskoristiti vlasti za nadzor prometa u blockchain tehnologiji, čime se ugrožava privatnost korisnika presretanjem njegovih podataka. U slučaju ovoga napada, korisnici blockchaina nisu svjesni prijetnje jer se prijenos podataka i operacije odvijaju kao i inače. [19]

Zaključno o ovoj vrsti napada, presretanje prometa je glavni rizik s kojim se susrećemo kod napada usmjeravanja te taj tip napada najčešće može biti pripremna faza za neku drugu vrstu napada.

3.4.2. Sybil napadi

U Sybil napadima, hakeri stvaraju brojne lažne mrežne čvorove odnosno pseudonimne identitete (lažne korisničke račune) kako bi preuzeli kontrolu nad mrežom. Na taj način se

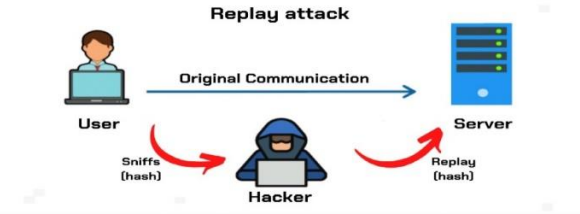
može kontrolirati protok informacija na mreži i dobivati informacije o IP adresama korisnika koji se na mrežu spajaju. Prikupljene informacije o IP adresama se koriste za daljnju analizu koja bi na kraju omogućila pristup nečijim podacima. [13, 16]

Također, ti lažni čvorovi mogu poremetiti izborni proces ubacivanjem lažnih informacija u mrežu kao što je pozitivno glasanje za pogrešnu transakciju. Na taj način i koristeći te čvorove, hakeri mogu postići konsenzus većine i poremetiti transakcije i blokove u lancu. [19]

Blockchain se pokušava suprotstaviti ovim napadima korištenjem konsenzusa udjela o radu u kojem rudari moraju riješiti matematički problem kako bi dokazali da nisu virtualni entiteti. [13]

3.4.3. Napadi ponavljanjem

Napad ponavljanjem događa se kada zlonamjernik prisluškuje mrežnu komunikaciju u svrhu odgode ili presretanja prijenosa podataka koji se odvija preko mreže, što omogućuje izvođenje napada bez potrebe za dešifriranjem. Dakle, zlonamjernik presreće, a potom ponavlja valjani prijenos podataka koji prolazi kroz mrežu, po čemu je napad i dobio ime. Ovaj napad može, u tradicionalnim sustavima, omogućiti hakeru da preuzme identitet drugog korisnika, nakon čega može dobiti vjerodajnice za mrežni pristup tog korisnika, čime ugrožava njegovu privatnost i povjerljivost. U slučaju blockchaina, napadi ponavljanjem se mogu zloupotrijebiti za krađu novčića iz novčanika dok se provodi teško račvanje (engl. hard fork). [26] Kada dođe do račvanja, blockchain se dijeli na dva dijela, pri čemu jedan pokreće naslijeđenu verziju softvera, a drugi pokreće novu, ažuriranu verziju. Tijekom račvanja napadačima postaje moguće koristiti napade ponavljanjem, budući da transakcija koju je na jednoj knjizi obradila osoba čiji je novčanik bio važeći prije račvanja, također će biti važeća na drugoj. Kao rezultat toga, osoba koja je primila određeni broj novčića od nekog drugog putem jedne knjige mogla bi se prebaciti na drugu knjigu, replicirati transakciju i prijevarom drugi put prenijeti identičan broj jedinica na svoj račun. [27]



Slika 3.6. Prikaz napada ponavljanjem [28]

3.5. Nepromjenjivost

Iako je svojstvo nepromjenjivosti u poglavlju iznad opisano kao prednost blockchain tehnologije, ono može biti i nedostatak te problem u nekim slučajevima. Ovo svojstvo može dovesti do potencijalno zabrinjavajućih rizika. Loša strana nepromjenjivosti se očituje u tome da ne postoji neka treća strana koja ima odgovornost i može ukloniti pogrešku ukoliko se nečiji podaci ukradu ili promjene. Također, u slučaju pametnih ugovora, onemogućeno je ispravljanje greške u kodu, ukoliko se naiđe na nju.

Primjer problema koji se dogodio kod nepromjenjivosti je "DAO Hack", u kojem je decentralizirani investicijski fond, stvoren na Ethereumu, prikupio više od 150 milijuna dolara u kriptovaluti. Međutim, pogreška u DAO kodu omogućila je korisniku da premjesti gotovo trećinu tih sredstava pod vlastitu kontrolu. Budući da je kod bio javan, ali nepromjenjiv, bilo je nemoguće zaustaviti hakiranje dok se događalo, a pritom ostati vjeran konceptu nepromjenjivosti. [29]

Još jedan primjer nepromjenjivosti u lošem kontekstu za korisnike je softverski novčanik Parity za kriptovalute. Naime, zbog ranjivosti u softverskoj knjižnici na koju se novčanici oslanjaju, više od 500 000 novčanika je trajno zamrznuto i izgubljeno za korisnike, a kao rezultat toga je gubitak od otprilike 150 milijuna dolara. Kao i kod prvog primjera, radi nepromjenjivosti se hakiranje nije moglo zaustaviti. [29]

3.6. Ostali problemi

Postoje neki problemi koje bih također voljela navesti, kao što je kvantno računarstvo koje bi u budućnosti moglo uništiti neke od kriptografskih principa koji stoje iza blockchain tehnologije. Također, smatram da je potrebno ukazati na problem sukladnosti s propisima o privatnosti i zaštiti podataka, iako to nije direktan problem očuvanja privatnosti kod blockchaina, ali se na neki način veže uz temu poglavlja.

3.6.1. Kvantno računarstvo

Kvantno računarstvo je nadolazeće, buduće polje istraživanja koje koristi kvantnu mehaniku kako bi se određeni izračuni izvodili jako brzo, puno brže od običnog računala, a time i učinkovitije. Postoji opravdan strah da takva računala predstavljaju prijetnju u očuvanju privatnosti blockchain sustava, budući da bi kvantna računala mogla imati loš utjecaj na kriptografiju korištenu u blockchain tehnologiji. Naime, u kriptografiji se algoritmi generiraju da funkcioniraju kao javni i privatni ključevi kako bi se osigurao ovlašteni pristup. Ova dva ključa su vođena složenim matematičkim odnosom koji je vrlo teško hakirati normalnim računalnim metodama u stvarnom vremenu. Međutim, postoji mogućnost da se puno brža kvantna računala mogu koristiti za probijanje kriptografskih ključeva i time ugroziti korisnike, a i cijeli sustav.

3.6.2. Problem sukladnosti s propisima o privatnosti i zaštiti podataka

Postojanje propisa, kao što je Opća uredba o zaštiti podataka, poznatija kao GDPR, dolazi u sukob s blockchain tehnologijom. Blockchain rješenja bi trebala biti u skladu s ovim propisima. Različita prava, uključujući pravo na informiranost, pravo na povlačenje privole, izravan pristup podacima, ispravljanje podataka, prenošenje podatke, pravo na informiranje o povredama podataka itd. bi trebala također biti zadovoljena u blockchainu. Postoje tri glavna pitanja sukoba između ove dvije strane. Prvi je problem s identifikacijom i obvezama voditelja obrade podataka i izvršitelja obrade, budući da je blockchain decentraliziran te nema određenog voditelja obrade podataka. Drugo pitanje koje se nameće su problemi s anonimizacijom osobnih podataka, budući da se hash vrijednosti i šifrirani podaci koji se koriste u blockchainu ne mogu smatrati anonimizacijom, prema GDPR-u. Treće pitanje oko kojeg se razilaze načela blockchaina i GDPR-a je, već navedena nepromjenjivost blockchaina te nemogućnost uklanjanja ili ispravljanja podataka. Pitanje je hoće li se ikada naći neko rješenje za ova tri problema, kako blockchain tehnologija ne bi dolazila u sukob s Općom uredbom o zaštiti podataka. [17]

4. TEHNIKE ZA POBOLJŠANJE PRIVATNOSTI NA BLOCKCHAINU

Problemi na koje se nailazi prilikom korištenja blockchain tehnologije u smislu privatnosti, povjerljivosti i anonimnosti izazivaju zabrinutost kod pojedinaca i organizacija te su stoga oni još uvijek oprezni u usvajanju blockchaina u njihovom poslovanju. Očuvanje privatnosti u javnom blockchainu nije trivijalno, budući da je to decentraliziran i otvoren sustav kojem nedostaju snažnija ovlaštenja za održavanje sustava i osiguranje privatnosti.

Postoji nekoliko glavnih zahtjeva za očuvanje privatnosti na blockchainu. Prvi je povjerljivost gdje korisnici žele da se njihove informacije o transakcijama, podacima zapisanim u blokovima i računima u sustavu blockchaina ne otkrivaju nikome, stoga je potrebno poduzeti mjere za ograničavanje pristupa tim podacima. Drugi zahtjev je anonimnost, koji podrazumijeva da protivnik ne može identificirati osobu u skupu identiteta. Također, jedan od najbitnijih zahtjeva je i nepovezivost transakcija kod kriptovaluta kako bi se spriječilo pogađanje pravog identiteta blockchain adrese analiziranjem putanje širenja blockchain transakcija. [8]

Sa sve većim ponavljanjem niza krađa, napada i prijetnji opisanih u prethodnom poglavlju, hitno je potrebno uspostaviti sigurnosna rješenja za poboljšanje sigurnosnih performansi blockchain sustava. Privatnost se može poboljšati implementiranjem tehnologija za poboljšanje privatnosti u blockchainu, koje pokušavaju usvojiti iznad navedene zahtjeve za očuvanje privatnosti. Najbolje od tih tehnologija biti će predstavljene i analizirane u nastavku.

4.1. Zero Knowledge Proof

Zero Knowledge Proof (ZKP) ili Dokaz nultog znanja prva je tehnologija za poboljšanje očuvanja privatnosti u blockchainu koja će biti opisana u radu. Ovo je jedna od najpopularnijih metoda za osiguranje anonimnosti i povjerljivosti transakcija. ZKP je shema enkripcije s kojom jedna strana zvana dokazivač može dokazati istinitost određenih informacija drugoj strani, verifikatoru, bez otkrivanja bilo kakvih dodatnih informacija. Da bi to bilo moguće, ZKP se oslanja na algoritme koji uzimaju neke podatke kao ulaz i vraćaju "točno" ili "netočno" kao izlaz.

4.1.1. Vrste

Zero Knowledge Proof dijeli se na interaktivni i neinteraktivni. Interaktivni zahtijevaju od dokazivača da učini niz radnji kako bi uvjerio verifikatora u istinu. Ovdje postoji slijed aktivnosti koji je povezan s matematičkim načelima vjerojatnosti. Neinteraktivni ZKP je onaj u kojem dokazivač može dizajnirati sve poteškoće, a verifikator može na njih odgovoriti naknadno. Dokazivač prosljeđuje tajne informacije posebnom algoritmu za izračunavanje dokaza bez znanja. Taj se dokaz šalje verifikatoru, koji pomoću drugog algoritma provjerava poznaje li dokazivač tajne podatke. Interakcija između dokazivača i verifikatora nije uvjet kod neinteraktivnog ZKP-a, ali ovaj tip ZKP-a zahtjeva dodatna računala ili softver. [30]

4.1.2. Osnovna struktura interaktivnog procesa

ZKP se jednostavnim rječnikom može opisati pomoću tri uzastopne radnje između sudionika A i B koje daju osnovni okvir. To su svjedočenje, izazov i odgovor. Kada sudionik A zna tajnu, koja je svjedok dokaza, u svakom trenutku može odgovoriti na određeni skup pitanja. Na početku A odabire nasumično pitanje s popisa i izračunava dokaz te daje sudioniku B dokaze. Izazov se događa kada sudionik B nasumično odabire pitanje iz skupa i pita A za odgovor. Tada A odgovara i vraća odgovor osobi B te osoba B tada može potvrditi da A govori istinu na temelju svog odgovora. Operacija se može ponoviti onoliko puta koliko se želi, dok vjerojatnost da će A dati krivi odgovor ne postane jako mala.

4.1.3. Matematički primjer interaktivnog Zero-Knowledge Proof

Jedan od najčešće korištenih dokaza znanja je onaj o diskretnom logaritmu. Protokol ima tri kruga, a definiran je nad cikličkom grupom G, reda q s generatorom g. Kako bi dokazao nulto znanje, koristi se izraz (4.1):

$$L=\{(x,w):x=g^w\} \tag{4.1}$$

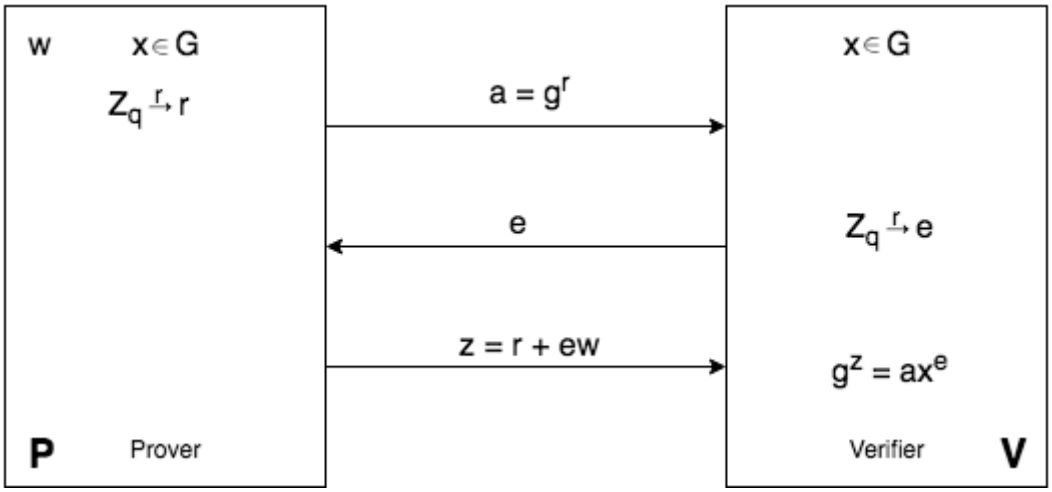
gdje je:

w svjedok, tajni ulaz,

g generator,

x javni ulaz.

Svjedoka w, odnosno diskretni logaritamski eksponent je teško izračunati. Dokazivač komunicira s verifikatorom kao što je prikazano na slici 4.1. [31]



Slika 4.1. Interaktivni proces ZKP-a [31]

Na početku dokazivač P ima tajni ulaz w i javni ulaz $x=g^w$, a verifikator V ima samo javni ulaz $x=g^w$. Oboje imaju zajedničke parametre, g i q. [31]

Proces dokaza započinje tako što dokazivač P generira element slučajne grupe h, uzorkuje slučajni cijeli broj r te šalje slučajnu vrijednost $a=g^r$ verifikatoru. Potom verifikator V odabire slučajni nasumični izazov $e \in \{0,..,q-1\}$ i šalje dokazivaču koji odgovara na izazov odgovorom $z=r+ew$. Verifikator prihvaća odgovor i uvjeren je u dokaz ako i samo ako je g^z jednako ax^e. [31]

4.1.4. Svojstva

Svojstva ZKP-a su potpunost, ispravnost i nulto znanje. Potpunost se odlikuje u tome da verifikator može potvrditi kako dokazivač posjeduje potrebne ulazne podatke, ukoliko je izjava istinita, dok ispravnost tvrdi da ukoliko je tvrdnja neistinita, verifikator ne bi bio uvjeren u tvrdnju ni u kojim okolnostima. Ova dva svojstva dovode do nultog znanja u kojem verifikator nema daljnje informacije o korisniku ni u jednoj okolnosti, odnosno ne zna nikakve informacije

osim je li izjava istinita ili lažna. Osnovna ideja ZKP-a je smanjiti količinu komunikacije između dva peera, kako bi se održala povjerljivost informacija. [17]

4.1.5. Primjena

U blockchain tehnologiji, ZKP se koristi za provjeru valjanosti transakcija između korisnika bez otkrivanja informacija o prijenosu, ali i za aplikacije bazirane na blockchainu za razmjenu poruka gdje pojedinac može dokazati svoj identitet bez objavljivanja dodatnih osobnih podataka. Također, kombinacija ZKP-a i blockchaina omogućuje korisnicima sigurno dijeljenje složenih dokumenata. ZKP imaju potencijal za šifriranje podataka u dijelovima, što korisnicima omogućuje kontrolu određenih blokova i vidljivosti informacija sadržanih u njima, dopuštajući nekim korisnicima pristup, dok se drugima ograničava. ZKP bi mogao imati i glavnu ulogu u online glasovanju, ukoliko je ono implementirano pomoću blockchain tehnologije, gdje bi birači mogli dokazati svoje pravo na glasanje bez otkrivanja identiteta. To osigurava da će svaki glas doista biti anoniman. [30]

4.1.6. Prednosti i nedostatci

Prednosti ZKP tehnologije su jednostavnost, budući da jamstva nultog znanja nemaju potrebu za sofisticiranim tehnikama šifriranja, ali i privatnost u smislu ne otkrivanja nikakvih informacija. Skalabilnost je još jedna prednost jer ZKP povećava propusnost i skalabilnost blockchaina. [30]

Glavni nedostatak ZKP-a je da sadrži oko dvije tisuće izračuna po jednoj transakciji, za čije je dovršavanje potrebno određeno vrijeme, a i specijalizirani strojevi radi potrebne količine računalne snage, koji su skupi, baš kao i troškovi provjere dokaza. Iz tih razloga je ZKP neučinkovit. Također, slanje poruka za potvrdu i dokaz može biti oštećeno ili izgubljeno u prijenosu, a ukoliko inicijator transakcije zaboravi svoje podatke, svi podaci povezani s njom se gube te je s time ZKP pomalo ograničen. [30]

4.1.7. Testiranje

Testiran je kod interaktivnog zero-knowledge proof procesa, koji je preuzet s Interneta [32] te napisan u pythonu, a temelji se na Schnorrovoj metodi diskretnog logaritma. U preuzeti kod nadodane su tri linije, koje imaju funkcionalnost da računaju vrijeme izvršenja programa u ovisnosti o različitim vrijednostima svjedoka, odnosno različitim tajnim vrijednostima:

import time

start_time=time.time()

print("Proof time: %s seconds" % (time.time() - start_time)),

gdje je start_time dodan na sam početak koda, iza učitavanja potrebnih knjižnica, a ispis proteklog vremena na samom kraju koda, kao zadnja linija.

U kodu, dokazivač ima javni ključ za dokazivanje (N,g,X) i tajni ključ za dokazivanje (N,x). Ulazni parametri su tajna vrijednost x čije je znanje potrebno dokazati, generator g te vrijednost modula N (prosti broj), dok su izlazni parametri potvrda o uspješnom ili neuspješnom dokazivanju te vrijeme izvršenja programa za izračun interaktivnog dokaza.

Testirani su slučajevi gdje je g=3, N=89, a tajna vrijednost x koju je potrebno dokazati se mijenja. U početku vrijeme izvršavanja programa neznatno raste, dok se kasnije, sa sve većim porastom vrijednosti x, sve više povećava. Rezultati testiranja dani su u tablici 4.1.

Tablica 4.1. Vrijeme izvršenja programa s porastom vrijednosti dokazivanja x

x	*10*	*100*	*1000*	*10000*	*100000*	*1000000*	*10000000*
Vrijeme izvršenja programa	*0,004 s*	*0,005 s*	*0,006 s*	*0,047 s*	*0,728 s*	*17,288 s*	*432,604 s*

Unatoč tome što se u testu količina podataka koje je potrebno dokazati kreće od 1B do samo nekoliko bajtova, zaključujem da se vrijeme izvršenja programa brzo povećava s obzirom na količinu podataka koju je potrebno dokazati. S ovim jednostavnim testom se donekle može

potkrijepiti gore navedeni nedostatak ove tehnologije, a to je veliko vrijeme potrebno za izračunavanje dokaza, što za sobom vuče skupe strojeve koji su za to sposobni.

Također, ovim testom je utvrđena interaktivnost procesa, budući da je program napravljen kao simulacija interaktivnog procesa koji se odvija u više koraka između dva sudionika, kao što je opisano u potpoglavljima 4.1.2. i 4.1.3.

4.2. Zero-Knowledge Succinct Non-Interactive ARgument of Knowledge (zk-SNARK)

Kod neinteraktivnog ZKP-a postoji tehnika koja se zove sažeti neinteraktivni argument znanja (engl. Succinct Non-Interactive Arguments of Knowledge - zk-SNARK), a koja je u novije vrijeme postala jedna od najčešćih tehnika ZKP-a. Zbog svoje opširnosti, upotrebe i važnosti, u radu će biti navedena u posebnom potpoglavlju.

4.2.1. Kako radi zk-SNARK?

Zk-SNARK se sastoji od tri algoritma definirana na sljedeći način. Prvi algoritam, generator ključa G uzima tajni parametar lambda i program C te generira dva javno dostupna ključa, ključ za dokazivanje pk i ključ za provjeru vk. Ovi ključevi su javni parametri koji se moraju generirati samo jednom za određeni program C. Tajni parametar lambda je problematičan, budući da svatko tko zna taj parametar može generirati lažne dokaze. Stoga se zahtjeva pouzdan proces, pazeći da parametar lambda bude uništen u procesu. [33]

Drugi algoritam, dokazivač P generira dokaz:

$$prf = P(pk, x, w) \tag{4.2}$$

gdje je:

pk ključ za dokazivanje,

x javni ulaz,

w tajna vrijednost, svjedok

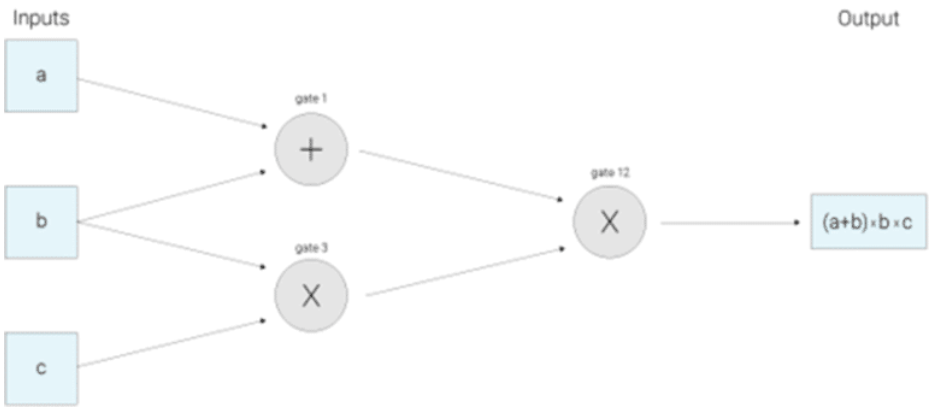
koji dokazuje da dokazivač poznaje svjedoka i da svjedok zadovoljava program. [33]

Na kraju verifikator V izračunava V(vk, x, prf) koji vraća true ako je dokaz točan, a false u protivnom. Stoga ova funkcija vraća true ako dokazivač zna svjedoka w koji zadovoljava C(x,w) == true. [33]

4.2.2. Matematički koncept rada

Tijekom procesa transakcije, zk-SNARK pretvara informacije koje treba dokazati u algebarske jednadžbe. Primjer naveden ovdje je jako jednostavan s ciljem da se razumije osnovna matematička pozadina iza zk-SNARKa. Funkcija programa u kojoj se nalazi dokaz se rastavlja na logičke korake u najmanje moguće, temeljne matematičke operacije i time se stvara aritmetički sklop. Primjer rastavljanja izraza (a+b)*(b*c) prikazan je na slici 4.2. U objašnjenju zk-SNARK, ulazne vrijednosti a, b i c sa slike se samo pomiču slijeva na desno kao po žici prema izlazu. [34]

Drugi korak je R1CS (engl. Rank 1 Constraint System) koji je važan za provjeru kreću li se vrijednosti ispravno, u pravom smjeru. U primjeru sa slike 4.2., R1CS može potvrditi da bi vrijednost koja dolazi iz vrata množenja s ulaza b i c bila b*c. Verifikator mora provjeriti različita ograničenja, odnosno jedno za svaku žicu aritmetičkog kruga. Pristup koji spaja sva ograničenja u jedno je QAP (engl. Quadratic Arithmetic Program), odnosno program kvadratne aritmetike. QAP implementira istu logiku kao i R1CS, ali koristi polinome umjesto točke. Korisnici moraju potvrditi da se dva polinoma podudaraju u jednoj nasumično odabranoj točki, što pomaže u ispravnoj provjeri dokaza. [34]



Slika 4.2. Rastavljanje matematičkog izraza na logičke korake [34]

Dalje se koristi algoritam eliptičnih krivulja. Dokazivač sa znanjem o točki koju bi verifikator odabrao za procjenu mogao bi stvoriti nevažeće polinome. Zabrinjavajući faktor ovdje je da bi nevažeći polinomi mogli zadovoljiti identitet u određenoj točki. Pomoć oko ovog problema se nalazi u uparivanju eliptičnih krivulja koja osigurava da ni dokazivač ni verifikator ne znaju koja se točka koristi za procjenu polinoma. [34]

Zaključno, Zk-SNARK je baziran na algoritmu eliptičnih krivulja koje pretpostavljaju da je pronalaženje diskretnog algoritma nasumičnog elementa (točke) eliptičke krivulje u odnosu na javno poznatu baznu točku neizvedivo. To znači da ako je vjerojatnost da bi se dokaz mogao pogoditi slučajno dovoljno niska, provjere se mogu izvršiti (dokazi se mogu potvrditi) nakon što se dosegne odgovarajući prag. [34]

4.2.3. Svojstva

Svojstva ZK-SNARKs su otkrivena u samom akronimu. Nulto znanje je prvo svojstvo, kao i kod ZKPa, što znači da verifikator ne uči ništa iz dokaza, osim da je valjan. Dalje, sažetost se odlikuje u tome da se provjera može izvršiti u kratkom vremenskom intervalu, odnosno u nekoliko milisekundi, a dokaz može biti pohranjen u malom broju bajtova. Neinteraktivnost ne zahtjeva interakciju dokazivača i verifikatora, već je potrebna samo jedna poruka od dokazivača i dokaz koji verifikator može potvrditi van mreže. Argumenti označavaju da je dokazivač ograničen polinomskim vremenom, stoga je ispravnost računska, a potpunost ista kao kod ZKPa. Znanje se odnosi na informacije koje posjeduje osoba koja dokazuje. [17]

4.2.4. Primjena

Zerocash i Zcash su najpoznatije implementacije zk-SNARKs-a, kao sustavi tokena, odnosno kriptovalute, koje se temelje na blockchainu. Zk-SNARKS se upotrebljava kao način da transakcije budu privatne i šifrirane na blockchainu, dok se još uvijek potvrđuju konsenzus mehanizmima.

Zerocash jamči povjerljivost iznosa transakcije i podržava plaćanje bilo koje denominacije. Korisnik može kovati kovanice različitih denominacija u više kovanica jednakog iznosa, svaki sa svojim iznosom i serijskim brojem. Tijekom procesa kovanja kovanica, korisnik treba generirati obvezu i dodati je na zajednički popis obveza. Za prijenos kovanica primatelju,

korisnik kriptira sadržaj transakcije (iznos i adresu primatelja) s javnim ključem primatelju i emitira kriptiranu transakciju cijeloj mreži. Nakon što primatelj dobije sadržaj transakcije s privatnim ključem, on generira serijski broj za te kovanice. Umjesto da svaki čvor validatora sam provjerava transakcije kao u Bitcoinu, oni provjeravaju argument dokazivača koji je provjerio ulazne i izlazne iznose te da privatni ključevi odgovaraju ulaznim transakcijama potrošnje. Dakle, Kada koristi zk-SNARK za provjeru transakcije, rudar treba samo potvrditi valjanost dokaza koje je dostavio inicijator transakcije. [8]

Zcash također na sličan način koristi zk-SNARK kako bi dokazao da su uvjeti za valjanu transakciju zadovoljeni bez otkrivanja ključnih informacija o uključenim adresama ili vrijednostima.

4.2.5. Prednosti i nedostatci

Prednost zk-SNARKa je taj što daje valjani dokaz ispravnog izvršenja neke funkcije koja se može zadržati u tajnosti. Nadalje, rudari ne mogu saznati informacije o transakciji, čime se osigurava anonimnost. Svaki je novčić identificiran jedinstvenim jednokratnim serijskim brojem, što može učinkovito spriječiti napade dvostrukog trošenja. Također, Korištenje zk-SNARK-ova poboljšava performanse smanjenjem veličine dokaza i vremena verifikacije čime se povećava učinkovitost. [17]

Unatoč izvrsnim performansama u očuvanju privatnosti i učinkovitosti ove tehnike, njezina sigurnost zahtjeva pouzdan proces postavljanja koji određuje argumente zk-SNARKs-a, generiranjem javnih parametara. U početku, kada se ti ključni parametri kreiraju, postoji skriveni, već spomenuti, parametar kojeg verifikator kasnije briše. Ako se ti osjetljivi podaci ne obrišu nakon nastanka, tajni parametri mogli bi se koristiti za lažno izvršavanje transakcija zavaravanjem zaštitnih mjera verifikacije. To znači da bi hakeri mogli dobiti kopiju odgovarajućih privatnih ključeva i koristiti ih za stvaranje krivotvorenih tokena ili kripto sredstava na mreži. [34]

4.2.6. Testiranje

Za metodu zk-SNARK testiran je kod preuzet s Interneta [35], napisan u pythonu. Kod prikazuje opći jednostavan proces zk-SNARK tehnologije opisan u potpoglavlju 4.2.1.

Program dokazuje neinteraktivnost procesa, budući da nema interakcije. U preuzeti kod nadodana je funkcionalnost izračunavanja proteklog vremena koje je potrebno za provjeru dokaza:

Import time

start_time = time.time()

print("Proof time: %s seconds" % (time.time() - start_time))

Prva linija nadodanog koda je stavljena na početku koda gdje se učitavaju potrebne knjižnice, druga linija koda koja pokreće računanje vremena je ubačena iznad linije koja vrši provjeru dokaza, dok je ispis proteklog vremena dodano kao zadnja linija koda. Vrijeme provjere dokaza je testirano u ovisnosti o količini podataka koja se provjerava, odnosno testirano je vrijeme provjere dokaza za stringove različite duljine. Rezultati su prikazani u tablici 4.2.

Iz rezultata je vidljivo kako se s porastom količine podataka vrijeme provjere dokaza povećava, ali ne toliko koliko kod interaktivnog ZKP-a, što je i za očekivati, budući da je u zk-SNARKu provjera sažetija.

Tablica 4.2. Vrijeme provjere dokaza ovisno o količini podataka za provjeru

Količina podataka za provjeru	*100 b*	*500 b*	*2 Kb*	*4 Kb*	*8 Kb*	*10 Kb*
Vrijeme provjere dokaza	*0,000997 s*	*0,00336 s*	*0,00441 s*	*0,0041 s*	*0,0134 s*	*0,0154 s*

4.3. Zero Knowledge Scalable Transparent ARguments of Knowledge (zk-STARK)

Zk-STARK je nastao kao posljedica nedostataka zk-SNARKa. Pouzdan proces postavljanja se protivi načelu blockchaina, prema kojem se ne vjeruje drugim čvorovima mreži. Cilj je zk-

SNARK tehnologiji dodati svojstvo transparentnosti, kako postavljanje i verifikacija nebi ovisila o tajnim vrijednostima koje se moraju izbrisati radi sigurnosti sustava. [17]

4.3.1. Svojstva

Na temelju gore navedenog zahtjeva, osmišljena je zk-STARK tehnologija koja ima i svojstvo transparentnosti te skalabilnosti. Ova tehnologija je skalabilnija od zk-SNARKa, budući da brže generira i provjerava dokaze. Programerima se omogućava da presele računanje transakcija i pohranu izvan blockchaina. Zk-STARK dokaze koji provjeravaju točnost izvanlančanih izračuna mogu proizvesti usluge izvan lanca. Nakon toga, ti se dokazi ponovno objavljuju u lancu tako da svatko može provjeriti izračun. Taj princip koriste zk-Rollups pametni ugovori. Također, iako i zk-SNARK i zk-STARK koriste polinome za provjeru dokaza, zk-STARK se u kasnijoj fazi oslanja na slučajnost koja se može javno provjeriti za generiranje javnih parametara za dokazivanje i provjeru umjesto na pouzdanu postavku. [17]

4.3.2. Prednosti i nedostatci

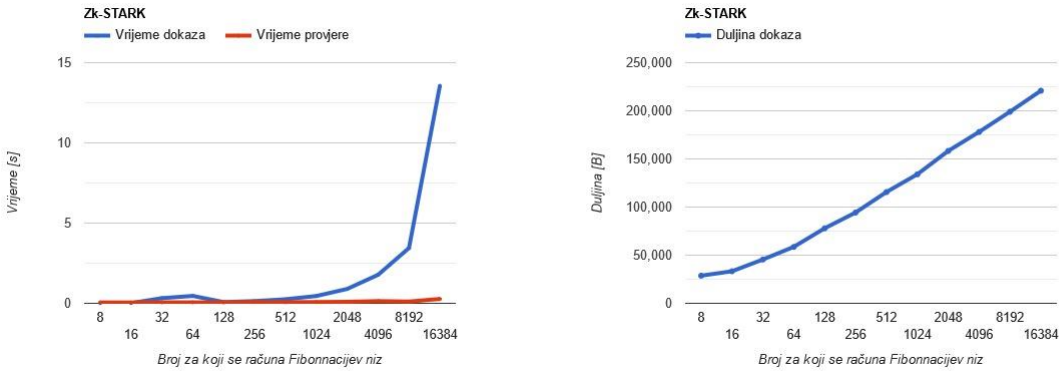
Osim što je skalabilniji i brži pri dokazivanju i verifikaciji s obzirom na količinu izračuna od zk-SNARKa, zk-STARK ima još neke važne prednosti. Budući da kvantno računarstvo predstavlja problem u budućnosti, smatra se da je zk-STARK otporan na prijetnju kvantnog računarstva jer se oslanja na hash vrijednosti otporne na sudare. Za razliku od uparivanja javno-privatnih ključeva koji se koriste u kriptografiji eliptične krivulje u zk-SNARKu, algoritmima kvantnog računalstva teže je razbiti hash otporan na kolizije. [34]

Nedostatak zk-STARKa je što proizvodi veće dokaze i samim time ima veće troškove verifikacije i električne energije.

4.3.3. Testiranje

Za tehnologiju zk-STARKS izabrala sam kod preuzet s Interneta [36], koji implementira dokaz za nulto znanje odabranog broja Fibonaccijevog niza koristeći zk-STARK tehnologiju. Pokreće se datoteka test.py u kojoj se može mijenjati Fibonaccijev broj za izračunavanje. Na

slici 4.3. su prikazani vrijeme dokaza, vrijeme provjere te duljina dokaza s obzirom na izračune različitih brojeva Fibonaccijevog niza.



Slika 4.3. Vrijeme dokaza, vrijeme provjere i duljina dokaza u zk-STARK testu

Iz grafičkih prikaza se jasno vidi kako se vrijeme dokaza povećava poli-logaritamski, vrijeme provjere neznatno raste, a duljina dokaza raste linearno s količinom potrebnih izračuna. Na temelju ovog testa se dokazuje navedeno u potpoglavlju iznad, a to je da dokazi imaju podosta veliku duljinu, što je nedostatak ove tehnologije jer time rastu troškovi.

4.4. Bulletproofs

Bulletproofs su također neinteraktivni ZKP, slični zk-SNARK-u, ali temelje se i na povjerljivim transakcijama, čineći snažnu kombinaciju ova dva sustava. Bulletproofs su dokazi bez znanja, točnije, vrsta dokaza dometa ili raspona koji potvrđuju da postoji određena dodijeljena vrijednosti unutar određenog raspona, ali ne otkriva stvarne informacije o toj vrijednosti. Ova tehnologija stvara vrlo kratke dokaze, koji se mogu brzo provjeriti, bez velike računske snage. Ne nudi potpunu anonimnost ili privatnost, ali njegova implementacija skriva iznose povezane s izvršenim prijenosima. Prikazuje se samo podrijetlo i odredište transakcije, ali iznos nije otkriven. [37]

4.4.1. Kako radi bulletproofs?

Pozadina iza Bulletproofs-a je složena, stoga ću samo ukratko predstaviti neke matematičke koncepte koje ga čine mogućim. Budući da je cilj ove tehnologije sakriti vrijednosti transakcije, ne umanjujući mogućnost provjere njihove autentičnosti putem mreže, protokoli Bulletproofs-a se temelje na Pedersenovim obvezama. Kako bi se to postiglo, koristi se matematički trik koji pokazuje da je zbroj ulaza veći od zbroja izlaza. Dakle, ne stvaraju se novčići iz ničega, ne koriste se već potrošene novčiće ni ne koristi se negativni saldo koji ne posjedujete. Jednostavno se šalje šifrirana poruka u kojoj ostali mogu provjeriti koristi li se uistinu vlastiti saldo, ali nitko ne zna koliki je taj saldo. [34] Također, ova se tehnologija temelji i na pretpostavci diskretnog logaritma, jednosmjernog načina računanja koje čini neizvedivim izračunavanje ulaza s obzirom na samo izlaz i to je ono što ju čini kompatibilnom s algoritmima eliptične krivulje. [37]

Uz pretpostavku diskretnog logaritma, iskorištava se i Fiat-Shamir heuristika, odnosno metoda koja omogućuje agregiranje različitih dokaza višeizlaznih transakcija u jedan. [37]

4.4.2. Primjena

Najpopularnija primjena bulletproofs-a je omogućavanje povjerljivih transakcija. Monero i MimbleWimble su najpoznatiji po upotrebi bulletproofs-a. Monero je zabilježio do 80% smanjenja veličine transakcija i naknada nakon implementacije uz pomoć bulletproofs-a. Dakle, ako obična transakcija s dokazom raspona zauzima 10 KB prostora, transakcija s bulletproofs može uštedjeti do 80% te veličine, umjesto toga zauzimajući samo 2 KB. [37]

Također, primjenjuje se i u omogućavanju povjerljivih pametnih ugovora, iako je ta primjena još uvijek dosta skupa, stoga nije toliko popularna.

4.4.3. Prednosti i nedostatci

Bulletproofs su učinkoviti i sigurni te ne zahtijevaju pouzdane postavke. Nadalje, pomažu u smanjenju veličine transakcija jer omogućuju dokazivaču da agregira različite dokaze za transakcije s više izlaza u jedan dokaz. Prethodni dokazi raspona s povjerljivim transakcijama

su bili veličine 2,5 Kb, a kod bulletproofs-a su 610 bajta. Bulletproofs se smatraju bržima u usporedbi s drugim dokazima raspona, omogućujući kraće vrijeme provjere. Također, sposobni su za veliku uštedu prostora, a samim time i uštedu troškova s nižim naknadama.

Nedostatak je ipak taj da bulletproofs tehnologija zahtjeva veće vrijeme provjere i dokaza u većim mrežama dokazivanja, u odnosu na prethodno dvije spomenute (zk-SNARK i zk-STARK). Također, ranjivi su na kvantne napade jer rade pod pretpostavkom diskretnog logaritma, koji može za nekoliko minuta ili čak sekundi u slučaju kvantnog računala biti ugrožen. [37]

4.4.4. Testiranje

Za tehnologiju bulletproofs je testiran jednostavan kod, preuzet s Interneta, [38] a napisan u programskom jeziku Go. U testu se provjerava nalazi li se vrijednost unutar zadanog raspona, odnosno je li vrijednost x unutar raspona $[0, 2^n - 1]$, gdje je n broj bitova u bit vektoru.

Za odabran bit vektor 8 i npr. vrijednost 234 za koju se provjerava je li u zadanom rasponu, program ispisuje generirani dokaz, budući da 234 pripada rasponu $[0, 2^8 - 1]$. Kada se bit vektor postavi na 16, a provjerava se pripada li npr. vrijednost 450 zadanom rasponu, također se ispisuje dokaz.

Slučaj gdje dokaz nije generiran je onaj gdje sam odabrala bit vektor 8 te vrijednost 450, budući da 450 ne pripada rasponu $[0, 2^8 - 1]$. Program je ispisao "Value is above range! Not proving."

4.5. Confidential Transactions

Confidential Transactions (CT) ili povjerljive transakcije su protokol koji skriva količinu i adresu Bitcoin-a koja se šalje. Stvorene su kako bi se poboljšao kripto sustav blockchain-a. U tu svrhu, one dopuštaju upisivanje određenih informacija, gdje ih strane koje razmjenjuju podatke mogu pročitati bez većih poteškoća, a bilo tko izvan te dvije strane vidi samo kodirane informacije. Jedino što vanjski akter može učiniti je potvrditi jesu li kodirane informacije istinite, bez mogućnosti izvlačenja informacija iz transakcije.

4.5.1. Način rada

Na visokoj razini pregleda načina rada, spomenuti ću da povjerljive transakcije uvode nove adrese i format transakcije koji se sastoji od scriptPubKey, Pedersenove obveze i ecdh nonce.

Skripta PubKey sastoji se od povjerljive adrese transakcije (CTA) i matematičkog uvjeta da se Bitcoin može potrošiti samo ako je vlasništvo nad privatnim ključem adrese dokazano potpisom. Povjerljiva adresa transakcije je hash zaslјepljujućeg ključa (faktora zaslјepljivanja) uz regularnu Bitcoin adresu. Navedeni zaslјepljujući ključ ili faktor zaslјepljivanja, koji je nasumični niz brojeva, se koristi za skrivanje Bitcoin adrese i iznosa u blockchainu. [39]

Načelo Bitcoin-a zahtijeva da adrese moraju održati sumu bilance 0, što znači da broj Bitcoin-a koji se šalje na adresu mora odgovarati broju Bitcoin-a koji napuštaju adresu. Budući da CT prikriva iznose transakcije, nastaje problem u kojem mreža ne može utvrditi podudara li se izlaz s adrese s ulazom kako bi se održala navedena suma bilance 0. Ovaj problem se rješava Pedersonovom obvezom.

Pedersenova obveza je hash ukupnog Bitcoin izlaza skupa sa slučajnim zaslјepljujućim faktorom. Dakle, ova obveza implementira slijepe ključeve umjesto korištenja adresa za prijenose. Pedersenove obveze imaju svojstvo homomorfnosti koje čuva strukturu između dvije algebarske strukture, što odgovara kriptografiji jer se podaci mogu raspršiti koristeći osnovnu algebru. Drugim riječima, informacije se mogu prenijeti bez otkrivanja samih podataka. [39] Izraz koji prikazuje Pedersenovu obvezu je:

$$C(BF_1+D_1) + C(BF_2+D_2) = C(BF_1+BF_2,D_1+D_2) \quad (4.3)$$

gdje je:

BF faktor zaslјepljivanja,

D_1 i D_2 podaci.

Prilikom slanja sredstava stvaraju se dvije dodatne obveze, jedna za promjenu adrese koja se vraća korisniku, a jedna za odredišnu adresu. Nitko ne zna koliko se šalje, ali može provjeriti zbrajaju li se obveze promjene i odredišta (lijeva strana jednadžbe) s izvornom adresom (desna strana jednadžbe). [39]

Ecdh nonce je ključ za otključavanje povjerljive transakcije. Koristi se za priopćavanje šifriranih podataka primatelju transakcije, kako bi on mogao naučiti izlaz Bitcoin transakcije i faktor zaslјepljivanja povjerljive transakcije. [39]

4.5.2. Primjer rada

Kako bi bilo lakše razumijeti način rada povjerljivih transakcija, prođimo kroz jedan primjer rada povjerljivih transakcija koji je zamišljen na Bitcoin-u, iako zbog problema navedenih u potpoglavlju 4.5.4. ova ideja još nije zaživjela na Bitcoin-u.

Osoba A ima 2 Bitcoin-a na svojoj adresi i želi poslati osobi B 1 Bitcoin. Osoba A uzima Bitcoin adresu od B, stvara zaslјepljujući ključ i hashira to dvoje zajedno, čime se stvara povjerljiva adresa. Iako se to bilježi u javnoj knjizi, nitko ne zna da je povjerljiva adresa vezana za BTC adresu od osobe B, osim njih dvoje. Zatim A stvara povjerljivu transakciju, uzima isti zaslјepljujući ključ i izlaz od 1 Bitcoin-a te stvara Pedersenovu obvezu. Ovo skriva iznos Bitcoin-a koji osoba A šalje osobi B, ali i A i B mogu vidjeti iznos jer imaju javni zaslјepljujući ključ. Osoba ga ima jer je stvorila zaslјepljujući ključ, dok ga osoba B može izvesti s privatnim ključem svoje Bitcoin adrese. Osoba A ima Pedersenovu obvezu od 2 Bitcoin-a za svoju povjerljivu adresu transakcije. Kada A pošalje osobi B 1 Bitcoin, koristi određenu matematičku formulu za stvaranje hash vrijednosti. Zatim koristi istu formulu za slanje 1 Bitcoin na promijenjenu adresu. Dva se hasha zbrajaju kako bi se vidjelo je li to jednako 2 Bitcoin-a, kao i kod Pedersenove obveze adrese osobe A. Ako se to učini, tada postaje valjana povjerljiva transakcija.

Osoba A zatim stvara scriptPubKey s CTA koji je stvorila s Bitcoin adresom osobe B, pod matematičkim uvjetom da se 1 Bitcoin može potrošiti ako može potpisom dokazati vlasništvo nad privatnim ključem adrese. Transakcija se potom emitira i bilježi u javnom blockchainu. [39]

4.5.3. Primjena

Monero i MimbleWimble implementiraju protokole povjerljivih transakcija kako bi povećali sigurnost i privatnost unutar svoje mreže i sakrili iznose prenesene u transakcijama. Ovaj protokol omogućuje mreži rudara da provjere da preneseni novčići nisu stvoreni od nule. [39] Monero koristi ring CT, koje rade na temelju prstenastih potpisa, a omogućuje skrivene iznose, podrijetla i odredišta transakcija s razumnom učinkovitošću i provjerljivim stvaranjem novčića.

Također, bočni lanac Liquid Network koristi CT protokole za maksimiziranje privatnosti svoje mreže, dok istovremeno osigurava brže transakcije na svom blockchainu, koji radi paralelno s Bitcoin mrežom. [39]

4.5.4. Problemi

Dodavanje protokola povjerljivih transakcija u sustav uvećava veličinu transakcija, što ne pogoduje Bitcoin-u zbog ograničene veličine njegovih blokova. Također, implementacija ovog protokola uključuje promjenu mreže s hard fork-om, što bi zahtijevalo da većina sudionika mreže pristane na promjenu koda, a to je teško izvedivo. [39]

Slabost povjerljivih transakcija se može dogoditi iz razloga što su skriveni iznosi zapravo skriveni samo za tu konkretnu transakciju. Ukoliko se dogodi da sljedeća transakcija nije povjerljiva, podaci iz nje se mogu koristiti za retroaktivno izračunavanje koliko je kriptovaluta moralo biti uključeno u povjerljivu transakciju.

4.5.5. Testiranje

Testiranje povjerljivih transakcija je uključivalo testiranje načina rada povjerljivih transakcija, gdje se dodavanjem faktora zasljepljivanja skriva iznos transakcije uz očuvanje sume bilance 0 pomoću Pedersenove obveze. Kod je preuzet s Interneta [40], a napisan je u pythonu.

Ulazni parametri programa su dvije vrijednosti transakcije (v1 i v2), u testiranom slučaju vrijednosti 5 i 10 te dvije zasljepljujuće vrijednosti (r1 i r2), u testiranom slučaju 4 i 6, od kojih se svaka množi s točkom na eliptičnoj krivulji. Vrijednosti transakcije se skrivaju, a transakcije se uspoređuju pomoću formule Pedersenove obveze te se registrira uspješnost ukoliko je suma bilance 0. Rezultat pokretanja programa prikazan je na slici 4.4.

```
Transaction (r1*G + v1*G) + (r2*G +v2*G):  (752433967636452936941313387291718489343558319292145047
5092001648135397605842, 1480169823447983529768202569280543793791549643658209106679397675810212167
682)
Transaction (r3*G + v3*G):  (752433967636452936941313387291718489343558319292145047509200164813539
7605842, 1480169823447983529768202569280543793791549643658209106679397675810212167682)

Now let's compare...
Success!
```

Slika 4.4. Rezultat pokretanja programa za testiranje povjerljivih transakcija

4.6. Homomorfna enkripcija

Homomorfna enkripcija je još jedna metoda izvođenja operacija nad podacima bez otkrivanja privatnih vrijednosti i informacija. Kao i drugi oblici enkripcije, homomorfna enkripcija koristi javni ključ za šifriranje podataka. Za razliku od drugih oblika enkripcije, koristi se algebarskim sustavom koji omogućuje izvođenje funkcija na podacima dok su još šifrirani. Homomorfna enkripcija omogućuje izvođenje izračuna, odnosno binarnih operacija ili aritmetičkih sklopova, na šifriranim podacima bez prethodnog dešifriranja. Dakle, kada se šifrirani rezultat dekriptira, odgovara rezultatu operacija kao da su izvedene na čistom tekstu.

Homomorfna enkripcija ne stvara značajne promjene u svojstvima blockchaina, on će i dalje biti javan, ali sa šifriranim podacima.

4.6.1. Osnovni procesi homomorfne enkripcije

Općeniti proces homomorfne enkripcije opisan je u 4 koraka. Prvi korak je algoritam za generiranje ključeva koji generira trojke izlaznih ključeva, odnosno tajni par ključeva te ključ za evaluaciju. Zatim slijedi algoritam enkripcije koji šifrira poruku s javnim ključem i ispisuje šifrirani tekst, a iza toga ide algoritam dekripcije koji dekriptira šifrirani tekst s tajnim ključem i vraća poruku kao izlaz. Evaluacija je algoritam koji razlikuje homomorfnu enkripciju od ostalih enkripcija s javnim ključem. Evaluacija proizvodi evaluacijski izlaz, odnosno procijenjene šifrirane tekstove, uzimajući ključ za evaluaciju kao ulaz, ulazne šifrirane tekstove poruka i prethodne rezultate evaluacije. [41]

4.6.2. Parcijalna homomorfna enkripcija

Parcijalna ili djelomična homomorfna enkripcija je vrsta homomorfne enkripcije koja dopušta da se samo jedna operacija izvrši na šifriranom tekstu beskonačan broj puta. Određeni algoritam može biti aditivno homomorfan, što znači da zbrajanje dvaju šifriranih tekstova zajedno daje isti rezultat kao kodiranje zbroja dvaju otvorenih tekstova ili multiplikativno homomorfan, što znači da množenje dva šifrirana teksta zajedno daje isti rezultat kao kodiranje umnoška dva otvorena teksta. Dakle, u slučaju parcijalne homomorfne enkripcije, operacija koja se može izvesti beskonačan broj puta može biti samo zbrajanje ili samo množenje. [41]

Ove algoritme šifriranja je relativno lako za dizajnirati. Primjer djelomično homomorfne enkripcije je ispod navedeni RSA algoritam, koji je multiplikativno homomorfan. Ostali primjeri ove vrste enkripcije su ElGamal, Benaloh, Paillier itd.

4.6.3. Donekle homomorfna enkripcija

Donekle homomorfna enkripcija omogućuje izvođenje i zbrajanja i množenja, ali ograničen broj puta, do određene dubine u logici skupa. Ovaj algoritam je teže za dizajnirati od parcijalne, budući da podržava dvije operacije. Primjeri algoritama donekle homomorfne enkripcije su BFV (Brakerski-Fan-Vercauteren), Boneh-Goh-Nissim (BGN), Sander, Young i Yung (SYY) itd. [41]

4.6.4. Potpuno homomorfna enkripcija

Potpuno homomorfna enkripcija dopušta da se i zbrajanje i množenje izvedu na šifriranom tekstu beskonačan broj puta, podržavajući proizvoljne izračune na šifriranim podacima. Primjeri algoritama potpuno homomorfne enkripcije su Lattice based (Gentry), Over integers (Van Dijk), (R)LWE based (Brakerski and Vaikuntanathan), CKKS itd. Većina prvih algoritama potpuno homomorfne enkripcije prvo konstruira donekle homomorfni sustav, a zatim ga pretvara u potpuno homomorfni sustav korištenjem bootstrappinga. Naime, nakon procjene previše vrata aritmetičkog sklopa, šum postaje prevelik i više se poruka ne može dešifrirati. Bootstrappingom se označava proces osvježavanja šifriranog teksta kako bi se proizveo novi šifrirani tekst koji šifrira istu poruku, ali s nižom razinom šuma kako bi se na njemu moglo procijeniti više homomorfnih operacija. Grubo rečeno, to je kao da dekriptirate šifrirani tekst tajnim ključem, a zatim ponovno šifrirate poruku, s tom razlikom što je tajni ključ nepoznat i zamijenjen enkripcijom tajnog ključa, koja se zove bootstrapping ključ. [41]

Kasnije, u novijim generacijama potpuno homomorfne enkripcije su uvedene nove metode za računanje aritmetičkih sklopova na šifriranim podacima koji pojednostavljuju bootstrapping. [41]

Problem ove vrste homomorfne enkripcije je troškovna učinkovitost u pogledu brzine i zahtjeva za pohranu u usporedbi s operacijama otvorenog teksta. Algoritmi su spori i mogu imati vrlo visoke zahtjeve za pohranu podataka.

4.6.5. Matematički primjer algoritama

Jedan od najčešćih primjera algoritama parcijalne homomorfne enkripcije je RSA šifriranje u kojem postoje javni ključ (e,n) i privatni ključ (d), koji su cijeli brojevi koji provjeravaju jednakosti:

$$n = p*q \tag{4.4}$$

gdje je:

p prosti broj,

q prosti broj

i:

$$\varphi(n) = (p-1)*(q-1). \tag{4.5}$$

Zatim se odabere vrijednost e koja je relativno prosta prema φ(n) te se izračuna vrijednost privatnog ključa iz izraza:

$$d*e = 1 \bmod \varphi(n). \tag{4.6}$$

Enkripcija poruke zadana je s:

$$E(x) = x^e \bmod n \tag{4.7}$$

gdje je:

x poruka,

e javni ključ,

n suma dva primarna broja.

Grupno množenje dva šifrirana teksta je tada homomorfno svojstvo RSA enkripcije:

$$E(x)E(y) = x^e y^e \bmod n = (xy)^e \bmod n = E(xy) \tag{4.8}$$

gdje je:

x prva poruka,

y druga poruka,

dok su svi ostali znakovi iz formule su objašnjeni iznad. [17]

Primjer koji će biti predstavljen za niveliranu potpuno homomorfnu enkrpiciju je BFV algoritam. Enkripcije su definirane kao:

$$E(x) = \Delta * x \tag{4.9}$$

$$E(y) = \Delta * y \tag{4.10}$$

gdje je:

x prva poruka,

y druga poruka,

Δ konstanta.

Svojstvo homomorfnosti je opisano izrazima:

$$E(x)+E(y) = \Delta*(x+y), \tag{4.11}$$

$$E(x)*E(y) = \Delta^2*(x*y). \tag{4.12}$$

Navedena shema nije sigurna, budući da svatko može otkriti poruke x i y iz dva šifrirana teksta. Da bi ovakva shema enkripcije bila sigurna, dodaje se mala komponenta šuma i javni ključ te se tako koristi izraz:

$$E(m) = (\Delta*m+e_1+pk_1*u*e_2+pk_2*u) \tag{4.13}$$

gdje je:

m poruka,

Δ konstanta,

e_1 i e_2 male komponente šuma,

pk_1 i pk_2 dijelovi javnog ključa,

u nasumična vrijednost.

Za dešifriranje je potrebno poznavanje privatnog ključa s te se time dobiva izvorna poruka s malo šuma. Komponenta šuma mora ostati dovoljno mala, tako da kada se podijeli s konstantom Δ i zaokruži na najbliži cijeli broj, dobije izvorna poruka m. U ovom algoritmu treba paziti na to da šum ne postane prevelik jer se u tom slučaju šifrirani tekst neće moći dešifrirati u ispravnu poruku. Zbrajanje dvaju šifriranih tekstova zajedno povećava šum za malu količinu, dok množenje dvaju šifriranih tekstova zajedno povećava šum za mnogo više,

stoga ukoliko unaprijed znamo broj zbrajanja i množenja koji se želi izvesti, potrebno je odabrati parametre šifriranja na način da šum nikada ne postane prevelik. [42]

4.6.6. Primjena

Homomorfna enkripcija tj. skrivanje je jedna od temeljnih metoda za stvaranje zk-SNARK tehnologije i privatnih distribuiranih izračuna koji su shema za dokazivač-verifikator metodu u blockchainu. [17]

Također, u blockchainu se ova metoda može primijeniti kada imamo homomorfno šifriranu datoteku privatnog ključa i želimo s njom poslati Bitcoin transakciju, ali ne želimo da itko zna koji je bio naš izvorni privatni ključ. Dakle, šifriramo transakciju našom homomorfnom datotekom privatnog ključa i emitiramo informacije o transakciji kao i obično.

Još jedna izravna upotreba homomorfne enkripcije u blockchainu su Bitcoin ECDSA parovi ključeva, koji imaju aditivna i multiplikativna homomorfna svojstva. Par ključeva (a, A), odnosno privatne i javne vrijednosti i drugi par (b, B) mogu stvoriti treću važeću Bitcoin adresu dodavanjem ključeva kao (a+b, A+B). Osoba X prodaje svoju adresu (b, B) osobi Y objavljujući B, b i A+B, navodeći da samo onaj tko zna privatni ključ a+b (koji osoba Y može izračunati samo putem a) može potrošiti novčić. Na ovaj način, X može prodati svoju adresu Y, bez potrebe da zaštiti isporuku privatnog ključa b. [17]

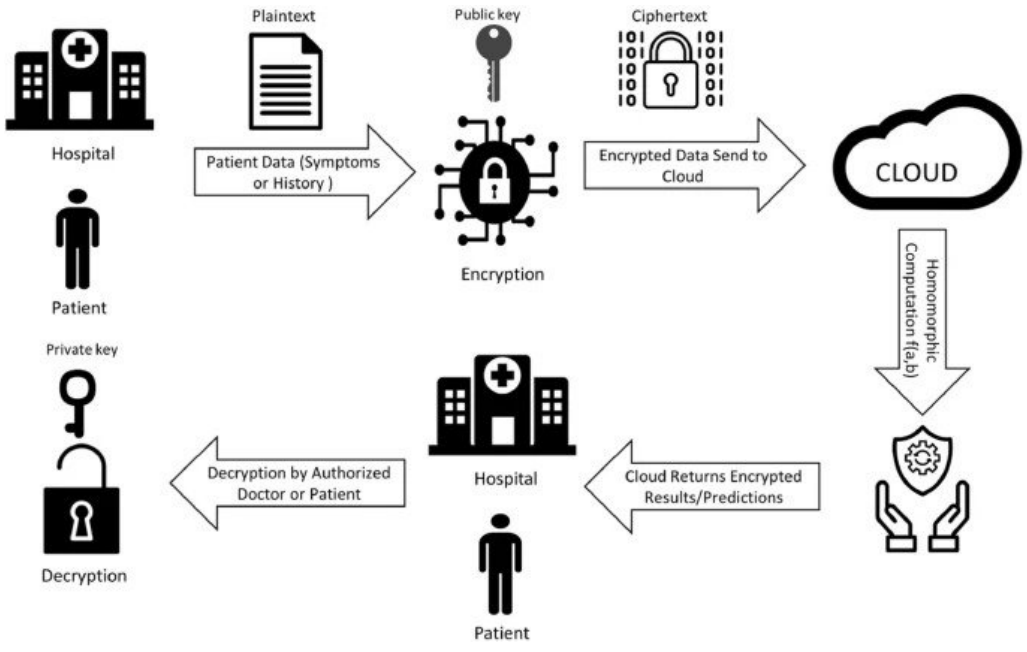
Također, homomorfna enkripcija se primjenjuje u pametnim ugovorima Ethereuma, gdje nudi slične značajke i veću kontrolu, a pritom zadržava sve prednosti Ethereuma netaknutima. [43]

Postoje mnoge mogućnosti ove tehnologije koje bi se mogle koristiti za buduće primjene homomorfne enkripcije u blockchain tehnologiji, a koje nisu vezane uz kriptovalute. Ukoliko se blockchain tehnologija počne u velikim razmjerima koristiti za medicinske informacije, informacije o nacionalnoj sigurnosti vojnih ili policijskih snaga ili za privatne informacije tvrtki i organizacija, homomorfna enkripcija će ovdje moći odigrati veliku ulogu. Primjerice, ova tehnologija može pomoći tvrtkama da se zaštite od rizika privatnosti u lancima opskrbe na način da svi podaci koje daju pouzdanim trećim stranama na obradu budu šifrirani. Nadalje, analitika podataka je način na koji mnoge tvrtke zarađuju, poput Facebook-a, koji pruža "besplatne" usluge prikupljanjem informacija o svojim korisnicima te njihovom obradom. Homomorfna enkripcija ovdje pruža rješenje, gdje bi primjerice Facebook mogao izvršiti

analizu podataka koja je potrebna bez mogućnosti pristupa izvornim podacima, ako ključeve za enkripciju osjetljivih podataka kontroliraju korisnici. [43]

Prikaz moguće primjene homomorfne enkripcije u svrhu zaštite medicinskih podataka ilustriran je na slici 4.5. Slika prikazuje kako se otvoreni podaci o pacijentu šifriraju javnim ključem te se pohranjuju u oblaku. Zatim se rade homomorfni izračuni nad šifriranim podacima, ustanovi se vraćaju enkriptirani rezultati ili predikcije koje mogu dekriptirati samo autorizirani doktor ili pacijent svojim privatnim ključem. [44]

Na kraju, spomenuti GDPR iz prethodnog poglavlja nalaže pravilo da podaci građana EU ostanu unutar EU, a njihovi zakoni navode da se zahtjevi ne odnose na šifrirane podatke. Uz homomorfnu enkripciju, tvrtke bi potencijalno mogle pohranjivati i obrađivati podatke na sustavima izvan EU, a zatim ih samo dekriptirati na poslužiteljima na lokacijama koje su u skladu sa zahtjevima GDPR-a. [45]



Slika 4.5. Primjena homomorfne enkripcije u medicini [44]

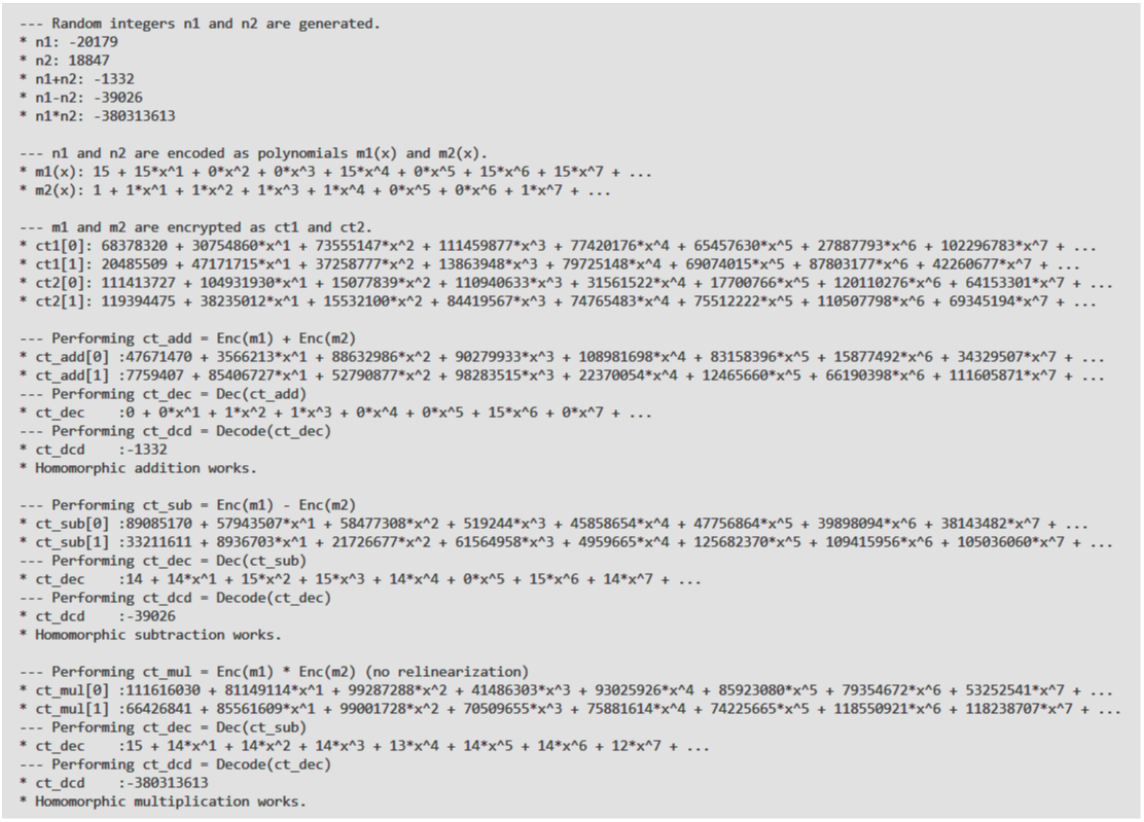
4.6.7. Prednosti i nedostaci

Homomorfna enkripcija pruža zaštitu privatnosti, kao i lak pristup šifriranim podacima. Mogućnost izvođenja operacija na šifriranim podacima bez dešifriranja, velika je prednost jer omogućuje bezbrižnost pri pohranjivanju osjetljivih podataka. Smatra se da će korištenje homomorfne enkripcije za pohranjivanje podataka na javnom blockchainu ponuditi najbolje od javnih i privatnih blockchain-a, u jednom paketu. Nadalje, nema ciljanja, odnosno kriminalci vas ne mogu ciljati jer ne mogu vidjeti koliko kriptovaluta posjedujete te nema nadzora, dakle vlade ne mogu nadzirati korisnike jer ne mogu dešifrirati vaše stanje na lancu. Također, potpuno homomorfna enkripcija sigurna je od kvantnog računarstva te nema potrebe za pouzdanim trećim stranama. [43]

Jedan od problema ove tehnologije je izvedba. Trenutni algoritmi zahtijevaju velike troškove računanja, koje može trajati mnogo više vremena na šifriranim, u odnosu na nešifrirane podatke. Time rastu režijski troškovi. [43]

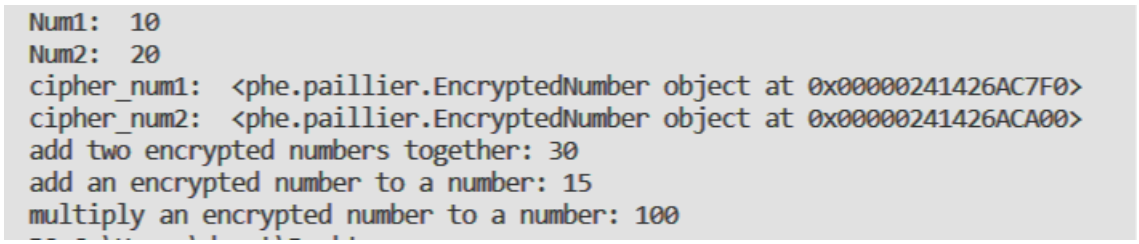
4.6.8. Testiranje

Prvi test homomorfne enkripcije je test algoritma BFV, tj. test algoritma donekle homomorfne enkripcije, koji je preuzet s Interneta [46]. U testu se generiraju nasumični brojevi koji se prvo pretvaraju u polinome otvorenog teksta, a potom se enkriptiraju. Zatim se dokazuje svojstvo homomorfnosti u zbrajanju, oduzimanju i množenju tako što se enkriptirani brojevi zbrajaju, oduzimaju ili množe, a kada se enkriptirani rezultati dekriptiraju, dobije se isti rezultat kao da smo zbrojili, oduzeli ili množili dva broja otvorenog teksta. Navedeno je prikazano na slici 4.6. koja predstavlja jedno pokretanje programa.



Slika 4.6. Pokretanje programa BFV algoritma homomorfne enkripcije

Drugi test homomorfne enkripcije testira Paillierov algoritam, koji je algoritam parcijalne homomorfne enkripcije. Kod je preuzet s Interneta [47]. Test prikazuje kako Paillierov algoritam podržava samo zbrajanje dva šifrirana teksta, dok njihovo množenje ne podržava. Također, u testu je prikazano kako se broj otvorenog teksta može dodati ili pomnožiti sa šifriranim tekstom. Slika 4.7. prikazuje slučaj gdje je zbrajanje dva šifrirana teksta podržano, a slika 4.8. slučaj gdje program ispisuje pogrešku kada se dva šifrirana teksta pokušaju pomnožiti.



Slika 4.7. Uspješno zbrajanje dva šifrirana teksta u testu Paillierovog algoritma

```
Num1:  10
Num2:  20
cipher_num1:  <phe.paillier.EncryptedNumber object at 0x0000027BA47FC7F0>
cipher_num2:  <phe.paillier.EncryptedNumber object at 0x0000027BA47FCA00>
Traceback (most recent call last):
  File "c:\Users\dorat\Desktop\pyhom.py", line 16, in <module>
    result = cipher_num1 * cipher_num2
  File "C:\Users\dorat\AppData\Local\Programs\Python\Python310\lib\site-packages\phe\paillier.py", line 508, in __mul__
    raise NotImplementedError('Good luck with that...')
NotImplementedError: Good luck with that...
```

Slika 4.8. Neuspješno množenje dva šifrirana teksta u testu Paillierovog algoritma

4.7. Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) ili sigurno višestranačko računanje je još jedna vrlo bitna tehnologija koja pomaže u očuvanju privatnosti prilikom korištenja javnog blockchaina. SMPC radi na pretpostavci da sve zainteresirane strane mogu komunicirati na sigurnom i pouzdanom kanalu, gdje svaka strana razmjenjuje šifriranu verziju svog privatnog unosa, koji se podvrgava računalnim operacijama za izgradnju željenog izlaza.

Jednostavno rečeno, SMPC je tehnologija u kojoj dvije ili više strana zajednički izračunavaju izlaz kombinirajući svoje pojedinačne ulaze, pri čemu ni jedna pojedinačna strana ne može vidjeti podatke drugih strana.

Razlika između tradicionalne kriptografije i SMPC je u tome što se tradicionalna kriptografija bavi prikrivanjem sadržaja, dok se ova nova vrsta računanja i protokola bavi prikrivanjem djelomičnih informacija o podacima dok se računaju podaci iz mnogih izvora i ispravno stvara rezultat.

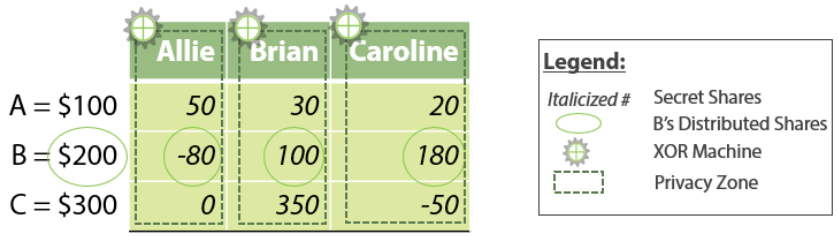
SMPC rješenja se trebaju pridržavati dva glavna principa, a to su privatnost ulaza i ispravnost. Privatnost ulaza se treba odlikovati u tome da se privatni podaci koje posjeduju strane koje surađuju na izgradnji izlaza ne mogu zaključiti ili izvesti, a ispravnost u tome da je dobiveni izlaz uvijek točan i da strane koje surađuju ne bi trebale moći utjecati na netočan rezultat. [45]

SMPC sustavi također moraju uzeti u obzir da određene strane mogu biti nepoštene i da je složenost implementacije izravno proporcionalna vrsti nepoštenih protivnika koji se očekuju u određenom slučaju upotrebe. [45]

4.7.1. Primjer

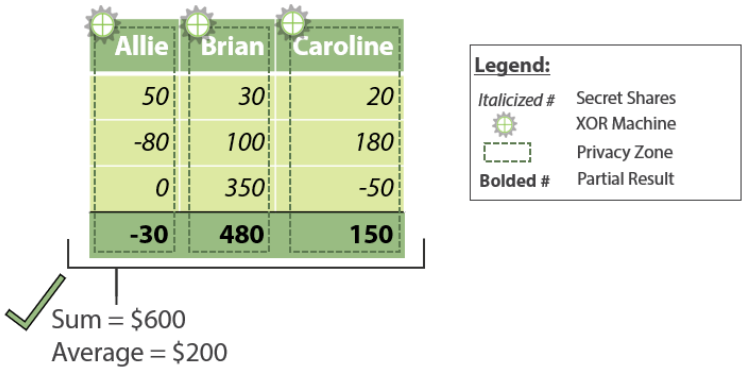
Uzmimo jednu jednostavnu situaciju kao primjer osnovnog koncepta rada višestranačkog računanja na visokoj razini. Ovaj primjer predstavlja aditivnu shemu tajnog dijeljenja. Troje kolega, Allie, Brian i Caroline žele izračunati svoju prosječnu plaću, bez da međusobno otkriju svoje podatke te bez da otkriju svoje podatke trećim stranama. Ovaj problem se može riješiti konceptom aditivnog dijeljenja tajni u višestranačkom računanju, koji se odnosi na podjelu tajne i njezinu distribuciju među skupom neovisnih sudionika.

U primjeru, Allie ima plaću od 100$, Brian 200$, a Caroline 300$. U dijeljenju tajni, svaki od tih iznosa dijeli se na tri nasumična dijela, primjerice Allie-na plaća od 100$ se dijeli na 50$, 30$ i 20$. Na taj način i ostale osobe dijele svoju tajnu. Allie zadržava jedan dio tajne od ukupno tri dijela za sebe, dok drugi dio tajne dijeli Brianu, a treći Caroline. Ostale dvije osobe naprave isti postupak dijeljenja tajne, što je prikazano na slici 4.9. [48]



Slika 4.9. Dijeljenje tajne pri višestranačkom računanju [48]

Kada se zbroje, tajna dijeljenja daju vrijedne informacije. Svaki sudionik lokalno zbraja svoje tajne udjele kako bi izračunao djelomični rezultat. Svaki od izračunata tri djelomična rezultata se zbraja, čime se dobiva suma tajni, koja je u ovom slučaju 600$. Suma tajni se dijeli na broj sudionika, u našem primjeru na 3 te se dobiva iznos od 200$ kao prosječna plaća troje sudionika. Opisan proces je prikazan na slici 4.10. [48]



Slika 4.10. Rezultat višestranačkog računanja [48]

Na ovaj način, troje kolega su saznali njihove prosječne plaće, bez otkrivanja iznosa svojih plaća međusobno.

4.7.2. Tehnike korištene pri blockchain tehnologiji

Razvijene su mnoge različite tehnike SMPC tehnologije s različitim svojstvima i za različite primjene. U nastavku će biti približena tehnika Threshold Signature Scheme (TSS) tj. kriptografija s pragom koje omogućuju dijeljenje privatnog ključa između više strana i Shamir Secret Sharing Scheme (SSSS), tj. Shamirova shema dijeljenja tajni, koja je vrsta TSS-a.

Prvo će biti pobliže pojašnjena tehnika Threshold Signature Scheme (TSS). Ova tehnologija je poznata kao kriptografija s pragom. Objašnjenje će biti više teorijsko, budući da je matematički model jako složen. Upotrebom Threshold Signature Scheme (TSS), kao tehnike MPC-a za izračunavanje digitalnih potpisa na distribuirani način, privatni ključevi mogu se podijeliti na dijelove i distribuirati na različit broj poslužitelja koji nikada i nikome ne otkrivaju svoj pojedinačni dio. U TSS-u mora biti ispunjen uvjet praga koji se odnosi na broj čvorova između kojih se dijeli ključ. Ako postoji n aktivnih potpisnika, za uspješnu transakciju mora biti ispunjen prag potpisnika t. Kada bilo kojih t aktivnih potpisnika od n daju svoj ključni udio, transakcija će biti odobrena.

Faze ove tehnike su sljedeće: faza generiranja ključa, potpisivanja i provjere. Faza generiranja ključa se sastoji od toga da svaka uključena strana generira tajni privatni ključ i da sve strane generiraju javni ključ koristeći vaš privatni ključ. U fazi potpisivanja sudionici koji se pridruže određenom procesu koriste svoje privatne ključeve kao privatne unose. Za dobivanje potpisa se ti unosi kombiniraju. U zadnjoj fazi, fazi provjere se javni ključ koji odgovara transakciji

koristi za provjeru potpisa. Shema se može koristiti s kriptografskim sustavima kao što su RSA, ECDSA ili Schnorr. [49]

Prednost TSS-a je u tome što privatni ključ nikada nije jedna točka kvara jer privatni ključ ne treba ponovno graditi, može se koristiti distribuirano. Ovaj proces je softverski te se privatni ključevi nikada ne pohranjuju zajedno, stoga ako se dogodi da potencijalni napadač uspješno napadne jedan dio ključa, nema pristup cjelokupnom ključu. [40] Ova tehnika je i fleksibilna, budući da se cijeli privatni ključ nikada ne otkriva stranama koje sudjeluju u protokolu. Dakle, može se jednostavno proširiti postojeći privatni ključ na nove sudionike protokola bez potrebe za otkrivanjem ili mijenjanjem para ključeva te bez izlaganja bilo kojeg dijela privatnog ključa. Također, transakcije obavljene putem novčanika koji koriste TSS tehniku su jeftinije, budući da se potpisi praga odražavaju kao jedan potpis te članovi ne moraju plaćati dodatne provizije za provjeru potpisa. [49]

SMPC protokoli s poštenom većinom često koriste dijeljenje tajne kao metodu, stoga ću opisati jednu od najčešćih metoda Threshold Signature Scheme, a to je Shamirova shema dijeljenja tajni, koja je također predstavljena u jednostavnom primjeru u potpoglavlju 4.7.1. iznad ovoga. Shamirova shema dijeljenja tajni rješava problem djelitelja koji želi podijeliti tajnu s između n strana, tako da bilo koji podskup od t+1 ili više strana može rekonstruirati tajnu, ali nijedan podskup od t ili manje strana ne može ništa naučiti o tajni. Shamirova shema dijeljenja tajni koristi činjenicu da za bilo koju od t+1 točaka na dvodimenzionalnoj ravnini $(x_1, y_1), \ldots, (x_{t+1}, y_{t+1})$ s jedinstvenim x_i, postoji jedinstveni polinom q(x) najvišeg stupnja t takav da je $q(x_i) = y_i$ za svaki i. Nadalje, moguće je učinkovito rekonstruirati polinom q(x) ili bilo koju specifičnu točku na njemu. Jedan način da se to učini je s Lagrangeovim baznim polinomima $\ell_1(x), \ldots, \ell_t(x)$, gdje se rekonstrukcija provodi izračunavanjem izraza:

$$q(x) = \sum_{i=1}^{t+1} \ell_i(x) \cdot y_i \tag{4.14}$$

Odavde se pretpostavlja da su svi proračuni u konačnom polju Zp, za prost broj p>n. Dalje u shemi, kako bi podijelio tajnu s, čvor odabire slučajni polinom q(x) najvećeg stupnja t pod ograničenjem da je q(0)=s. Zatim za svaki i=1,...,n, čvor daje i-toj strani udio tajne, odnosno $y_i=q(i)$. Iz tog razloga nam treba p>n, odnosno da se svakoj strani može dati različiti udio tajne. Rekonstrukcija pomoću podskupa bilo kojih t strana radi jednostavnom interpolacijom polinoma za izračunavanje q(x) i zatim deriviranjem s=q(0). Iako t+1 strana može potpuno povratiti tajnu, nije teško pokazati da bilo koji podskup t ili manje strana ne može naučiti ništa o tajni. To je zbog činjenice da imaju t ili manje točaka na polinomu, pa postoji polinom koji prolazi kroz te točke i kroz točku (0,s) za svaki mogući s koji pripada konačnom polju. Nadalje,

budući da je polinom slučajan, svi polinomi su jednako vjerojatni, pa su i sve tajne vrijednosti jednako vjerojatne. [45]

4.7.3. Primjena

SMPC se u blockchainu koristi za sigurno čuvanje i upravljanje imovinom te privatnim ključevima ili za osiguranje višestrukih potpisnika na transakciji. Kada se koristi za upravljanje ključevima, MPC softver radi lokalno na uređajima koji su određeni za sudjelovanje u funkcijama kao što su generiranje dijeljenja ključeva, djelomični potpisi, rotacija ključa, pohrana, oporavak, suspenzija i brisanje.

Nadalje, SMPC može pomoći u održavanju privatnosti i povjerljivosti transakcija. Transakcije se mogu skinuti s blockchaina i obraditi putem SMPC-a, a potvrda o transakciji se može zabilježiti na blockchainu kao dokaz.

Također, velike burze digitalne imovine mogu koristiti ovu tehnologiju za sigurno pohranjivanje imovine svojih korisnika. U budućnosti, primjenom blockchaina u drugim granama, SMPC bi se mogao koristiti za međusobno dijeljenje podataka između organizacija, držeći ih privatnima u isto vrijeme. Također, moguće buduće primjene su pružanje modela sigurnosti kao usluge u oblaku, dijeljenje podataka između banaka bez otkrivanja osobnih podataka o klijentima, izračunavanje medicinskih podataka pružateljima modela trećih strana bez curenja podataka itd. [50]

4.7.4. Prednosti i nedostatci

Prednosti SMPC tehnologije su mnogostruke, počevši od toga da niti jedna treća strana ne vidi povjerljive podatke. Dalje, uklanja se kompromis između upotrebljivosti i privatnosti podataka na način da nema potreba za ispuštanjem neke značajke kako bi se očuvala privatnost podataka. Sve se značajke mogu koristiti u analizi, bez ugrožavanja privatnosti. Osim toga, ova tehnologija je usklađena s GDPR-om u zahtjevu za prekogranični prijenos podataka jer se podaci nikada ne pomiču. Također, jako bitno za nadolazeće doba kvantnog računarstva, SMPC je siguran od kvantnih napada, budući da su podaci šifrirani tijekom upotrebe i da se "tajna" dijeli. [48] Na kraju, u usporedbi s homomorfnom enkripcijom, SMPC zahtijeva manje računalne snage.

Jedan od velikih nedostataka višestranačkog računanja su računalni troškovi, budući da se moraju generirati nasumični brojevi kako bi se osigurala sigurnost izračuna, što može usporiti vrijeme izvođenja. Također, tajno dijeljenje uključuje komunikaciju i povezanost između svih sudionika, što dovodi do viših troškova komunikacije u usporedbi s izračunom otvorenog teksta. [48] Osim navedenih problema, ova metoda zahtijeva da većina sudionika bude iskrena. Tako točnost nije zajamčena jer ispravnost izlaza ovisi o ulazima strana koje sudjeluju u izračunu, a za ulaze tih strana se samo pretpostavlja da su točni ako je ta strana iskrena. [17]

4.7.5. Testiranje

Testirala sam kod Shamirove sheme dijeljenja tajni, preuzet s Interneta [51], a napisan u pythonu. Potrebne knjižnice za pokretanje programa su cryptography i sslib. Program razbija lozinku (privatni ključ) na n dijelova koristeći Shamirovo tajno dijeljenje, a potrebno je t dijelova kako bi se lozinka rekonstruirala i mogla dešifrirati. Pokretanjem datoteke rsa_sss.py traži se proizvoljni unos broja dijeljenja na koji želimo podijeliti lozinku, minimalnog broja dijeljenja potrebnog za rekonstrukciju lozinke, lozinke te onih dijelova dijeljenja pomoću kojih želimo rekonstruirati lozinku.

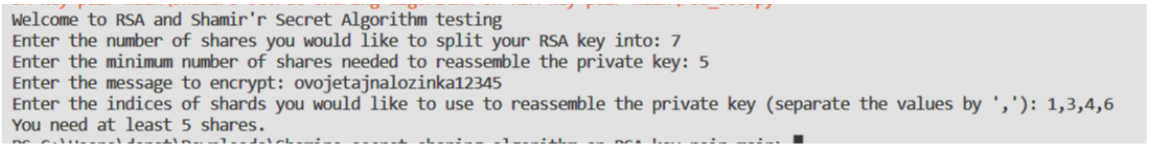
Prvi testni slučaj je onaj u kojem se program ispravno pokreće budući da je n=7, t=5 te je odabrano 5 dijelova za rekonstrukciju. Taj slučaj je uspješan te je prikazan na slici 4.11. Stvorila se datoteka public.txt s javnim ključem, Shard[t].txt datoteke s dijelovima privatnog ključa te se ispisala dešifrirana poruka.

```
Welcome to RSA and Shamir'r Secret Algorithm testing
Enter the number of shares you would like to split your RSA key into: 7
Enter the minimum number of shares needed to reassemble the private key: 5
Enter the message to encrypt: ovojetajnalozinka12345
Enter the indices of shards you would like to use to reassemble the private key (separate the values by ','): 1,3,4,6,7

Message successfully decrypted.
Decrypted message: ovojetajnalozinka12345
```

Slika 4.11. Uspješno pokretanje programa Shamirove sheme dijeljenja tajni

U drugom testnom slučaju sam odabrala n=7, t=5 te sam navela 4 dijela ključa za rekonstrukciju. Taj slučaj je neuspješan, budući da je potreban prag od minimalno 5 dijelova ključa kako bi se on ispravno rekonstruirao. Slika 4.12. prikazuje ovaj neuspješan slučaj.



Slika 4.12. Neuspješno pokretanje programa Shamirove sheme dijeljenja tajni

4.8. Diferencijalna privatnost

Iduća predstavljena tehnologija biti će diferencijalna privatnost. Prođimo kroz jedan jednostavan problem koji diferencijalna privatnost može riješiti, kako bi se shvatili svrhu diferencijalne privatnosti. Zamislimo da imamo bazu podataka o plaćama zaposlenika te da je upit koji dopuštamo u bazi podataka prosječna plaća zaposlenika u bazi podataka. Ako osoba A zna broj zaposlenika u tvrtki i pokrene ovaj upit prije i nakon što se osoba B pridruži organizaciji, tada osoba A može izračunati plaću osobe B. Način za to je jednostavan: osoba A zna broj zaposlenih u svojoj tvrtki i pokreće upit o prosječnoj plaći te dobiva određeni iznos. Zatim se osoba B pridružuje njegovoj tvrtki te osoba A ponovno pokreće upit o prosječnoj plaći i dobiva drugačiji iznos. Tada jednostavno može izračunati plaću od osobe B izrazom:

$$\text{plaća} = M(k+1)-N*k \tag{4.15}$$

gdje je:

k broj zaposlenih,

N iznos prosječne plaće prije osobe B,

M iznos prosječne plaće nakon osobe B.

Sada možemo definirati diferencijalnu privatnost. Diferencijalna privatnost je tehnika koja garantira da se rezultati statističkih upita ne mogu koristiti za prikupljanje informacija o određenim pojedincima. Ova tehnologija se bavi očuvanjem privatnosti proučavanjem otkriva li metodologija analize podataka informacije o pojedincima ili ne. Sastoji se od uvođenja određene količine nasumičnog šuma u upite podataka, na način da je bilo kakva statistička analiza cijelog skupa blizu stvarnim rezultatima, ali je zaključivanje nad bilo kojim pojedinačnim neizvedivo.

4.8.1. Definicija diferencijalne privatnosti

Matematička definicija diferencijalne privatnosti određena je izrazom:

$$\Pr[\mathcal{A}(D_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{A}(D_2) \in S] \tag{4.16}$$

gdje je:

Pr vjerojatnost,

A nasumični algoritam koji uzima skup podataka kao ulaz,

ε pozitivan realan broj,

D_1 i D_2 skupovi podataka koji se razlikuju po jednom elementu,

S skup algoritama s ulazima D_1 i D_2.

Dakle, algoritam koji djeluje na baze podataka kako bi proizveo rezultat je ε -diferencijalno privatan ako vrijedi za sve skupove podataka D_1 i D_2 koji se razlikuju u najviše jednom retku i za sve podskupove od S gdje Pr označava vjerojatnost. Ovaj mehanizam je diferencijalno privatan ako su rezultati od algoritma skupova D_1 i D_2 gotovo nerazlučivi za svaki izbor D_1 i D_2.

U diferencijalnoj privatnosti postoji kompromis između privatnosti i korisnosti podataka. Naime, dodavanje više šuma poboljšava privatnost, ali smanjuje korisnost podataka. Ovaj se kompromis kontrolira putem parametra epsilon ε koji je izveden iz Laplaceove distribucije vjerojatnosti. Što je ε manji, količina šuma raste, što rezultira boljom privatnošću i smanjenom korisnošću i obrnuto. [52]

4.8.2. Tipovi

Postoje dva tipa diferencijalne privatnosti, a to su globalna i lokalna diferencijalna privatnost.

Globalna diferencijalna privatnost dodaje šum izlazima baze podataka tj. šum se dodaje samo jednom, na kraju procesa prije nego što se podijeli s trećom stranom. Ovaj tip diferencijalne privatnosti se ne koristi u javnom, već u privatnom blockchainu.

Lokalna diferencijalna privatnost dodaje šum pojedinačnim (ulaznim) podatkovnim točkama, odnosno prije pojavljivanja u bazi podataka. U ovoj vrsti diferencijalne privatnosti se čuva individualna privatnost pojedinca. Ovdje se šum može dodati izravno u bazu podataka ili pojedinci mogu dodati šum u vlastite skupove podataka prije nego što ga stave u bazu podataka. U javnom blockchainu se koristi ovaj tip diferencijalne privatnosti. [52]

4.8.3. Svojstva

Jedno od svojstva diferencijalne privatnosti je to da nema naknadne obrade. Mehanizmi diferencijalne privatnosti su imuni na naknadnu obradu, budući da će bilo koja funkcija s tim mehanizmom ostati diferencijalno privatna. Dakle, ukoliko je mehanizam (algoritam) A diferencijalno privatan, a g je neka funkcija, onda će i g(A) također biti diferencijalno privatan. Ovo svojstvo pridonosi smanjenju napada povezivosti. Još jedno bitno svojstvo diferencijalne privatnosti je kompozicija. Primjena višestrukih mehanizama ili istog mehanizma više puta i dalje rezultira time da je ukupni mehanizam diferencijalno privatan, ali s različitim parametrom ε. Točnije, kompozicija k mehanizama od kojih je svaki ε-diferencijalno privatan najmanje je kε-diferencijalno privatna. Ovo svojstvo daje otpornost sustava na napade praćenja. [52]

4.8.4. Temelji diferencijalne privatnosti

Budući da se podaci štite dodavanjem određenog šuma u njih, temeljni dio ove tehnologije su mehanizmi dodavanja šumova. Tako se za izračun šuma u podatke se koriste različiti mehanizmi, a to su Laplaceov, Gaussov, geometrijski i uniformni za numeričke izlaze, a eksponencijalni za nenumeričke izlaze. Opisati ću one najčešće korištene mehanizme.

Laplaceov mehanizam preuzima deterministčku funkciju baze podataka i dodaje Laplaceov šum rezultatu. Upoznajmo se prvo s osjetljivošću funkcije Δf. To je gornja granica koliko moramo poremetiti njezin izlaz da bismo očuvali privatnost. Osjetljivost funkcije je dana izrazom:

$$\Delta f = \max_{\substack{x,y\in\mathbb{N}^{|\mathcal{X}|} \\ \|x-y\|_1=1}} \|f(x) - f(y)\| \tag{4.17}$$

gdje je:

Δf osjetljivost funkcije,

x i y ulazni podaci koji se razlikuju u jednom retku ili elementu,

f funkcija.

Dalje, Laplaceova distribucija je centrirana u nuli, a zadana je izrazom:

$$Lap(x|b) = \frac{1}{2b}\exp(-\frac{|x|}{b}) \tag{4.18}$$

gdje je:

x ulazni podatak,

b parametar skaliranja. [53]

Laplaceov mehanizam jednostavno će izračunati funkciju i poremetiti svaku koordinatu šumom izvučenim iz Laplaceove distribucije. Šum se skalira na 1/ε, to jest, dodavanjem šuma izvučenog iz Lap(1/ε). Laplaceov mehanizam je ε-diferencijalno privatan. Laplaceov mehanizam za bilo koju zadanu funkciju je definiran idućim izrazom:

$$M_L(x, f(\cdot), \varepsilon) = f(x) + (Y_1, \ldots, Y_k) \tag{4.19}$$

gdje je:

f funkcija sa stvarnom vrijednošću koja se planira izvršiti u bazi podataka,

Y_i nasumične varijable dobivene iz Laplaceove distribucije Lap(Δf/ε). [53]

Laplaceov mehanizam je dobar samo za upite niske osjetljivosti te je potrebna velika ε vrijednost.

Dalje, eksponencijalni mehanizam koristi se za nenumeričke upite. Umjesto dodavanja šuma izlazu, koristi se metoda zadana idućim izrazom:

$$\Pr[o] = e^{\frac{\varepsilon * u(x,o)}{2\Delta u}} \tag{4.20}$$

gdje je:

x ulaz,

o izlaz,

u funkcija korisnosti,

Δu osjetljivost.

Izlaz ovog mehanizma je uvijek član skupa R. Ovo je izuzetno korisno pri odabiru stavke iz konačnog skupa, kada odgovor sa šumom ne bi imao smisla. [53]

Idući od najčešćih mehanizama diferencijalne privatnosti je Gaussov mehanizam koji je alternativa Laplaceovom mehanizmu. Gaussov mehanizam ne zadovoljava čistu ε-diferencijalnu privatnost, već zadovoljava (ε,δ)-diferencijalnu privatnost, a zadan je izrazima:

$$F(x) = f(x) + N(\sigma^2) \tag{4.21}$$

$$\sigma^2 = \frac{2s^2 \log\left(\frac{1.25}{\delta}\right)}{\varepsilon^2} \tag{4.22}$$

gdje je:

x ulaz,

f funkcija,

s osjetljivost funkcije,

$N(\sigma^2)$ uzorkovanje iz Gaussove distribucije.

Ovaj mehanizam je prikladan jer je manje vjerojatno da će aditivni Gaussov šum poprimiti ekstremne vrijednosti u usporedbi s Laplaceovim šumom. Nasuprot tome, Laplaceov mehanizam je ε-diferencijalna privatnost i zbog toga je jači, budući da ograničava gubitak privatnosti čak i u najgorem slučaju koji može dovesti do potrebe za velikim količinama šuma. Ipak, Gaussov mehanizam se koristi jer se može proširiti na vektorske funkcije koje vraćaju vektore realnih brojeva. Takve funkcije su, primjerice, histogrami. [53]

4.8.5. Primjena

Primjena diferencijalne privatnosti u blockchainu pomaže pri očuvanju privatnosti i zaštiti identiteta na način da se dodaju nasumični šumovi podacima pohranjenim u tijelu blokova, adresama i privatnim podacima prije prijenosa poruke preko mreže, algoritmima konsenzusa, pametnim ugovorima itd.

Primjer primjene diferencijalne privatnosti pri online kupovini bitcoin kriptovalutom je sljedeći. Uz saznanje da je netko kupovao online za, primjerice, 0,000381 BTC s poznate web-lokacije za e-trgovinu, Bitcoin adrese koje su izvršile kupnju u vrijednosti od 0,000381 BTC

mogu se pronaći upitom za Bitcoin adresu web-mjesta i za transakciju s iznosom jednakim 0,000381 BTC iz blockchaina. Posljedično, otvara se prostor za istraživanje identiteta kupca, ali ukoliko se doda Laplaceov šum u iznos transakcije dok se transakcija uključuje u blockchain, vrijednost od 0,000381 ažurirala bi se kao, primjerice, 0,000383 ili 0,000377, stoga bi otkrivanje tih kupaca bilo spriječeno izravnim upitima. Štoviše, ne bi bilo jamstva da vrijednost najbliža 0,000381 odgovara povezanoj transakciji. [54]

Primjena diferencijalne privatnosti može biti i u perturbaciji grafa korisnika u kojem je tijek kriptovalute između korisnika tijekom vremena prikazan kao usmjereni grafikon. Perturbacija grafa može dodati lažne rubove (transakcije) između korisnika ili brisanje nekih postojećih rubova (stvarnih transakcija). Time se sprječava daljnja analiza i identifikacija korisnika koji sudjeluju u transakciji. [54]

U slučaju prijenosa ili emitiranja podataka u stvarnom vremenu u blockchain aplikacijama, točkasta strategija perturbacije podataka diferencijalne privatnosti može učinkovito dodati šum podacima bez narušavanja točnosti. U mehanizmu perturbacije podataka po točkama, prvo se izračunava stopa pogreške, a zatim se izračunava šum koristeći tu specifičnu stopu pogreške. Nakon izračuna specifične vrijednosti šuma, šum se dodaje određenoj vrijednosti kako bi se zaštitila njegova privatnost. Sada bilo koji zlonamjerni promatrač ne može točno pogoditi točnu vrijednost ili podatke, niti prisutnost ili odsutnost bilo kojeg pojedinca unutar decentralizirane baze podataka.

Još jedan slučaj upotrebe diferencijalne privatnosti u blockchain tehnologiji mogao bi biti učinkovito očuvanje identiteta pojedinaca tijekom emitiranja transakcija u mrežu, u kojem diferencijalna privatnost može poremetiti identitet na takav način da su informacije i dalje korisne za dovršetak transakcije, ali čvorovi ili protivnik u mreži neće moći procijeniti točan identitet pošiljatelja ili primatelja. U tom slučaju se koristi Laplaceov ili Gaussov mehanizam dodavanja šuma.

Osim toga, diferencijalna privatnost se može integrirati s računarstvom u oblaku koji se temelji na blockchainu. Tu se koristi rubno računarstvo temeljeno na federalnom učenju gdje su izvučena istraživanja rubnih čvorova gomile koristeći diferencijalnu privatnost i federalno učenje uz osiguranje da nijedan privatni podatak korisnika IoT neće biti analiziran. Za to se koristi Laplaceov šum koji se dodaje ekstrahiranim značajkama rubnih čvorova prije rudarenja rezultata u blockchain. Integracija diferencijalne privatnosti s računarstvom u oblaku koji se temelji na blockchainu je još u razvoju, stoga još u ovom području metode i tehnike nisu potpuno sigurne i privatne. [55]

Također, aktivno se radi na integraciji strategija očuvanja privatnosti sa zdravstvenim sustavima, energentskim sustavima, trgovanjem nekretnina itd. koji se temelje na blockchainu.

4.8.6. Prednosti i nedostatci

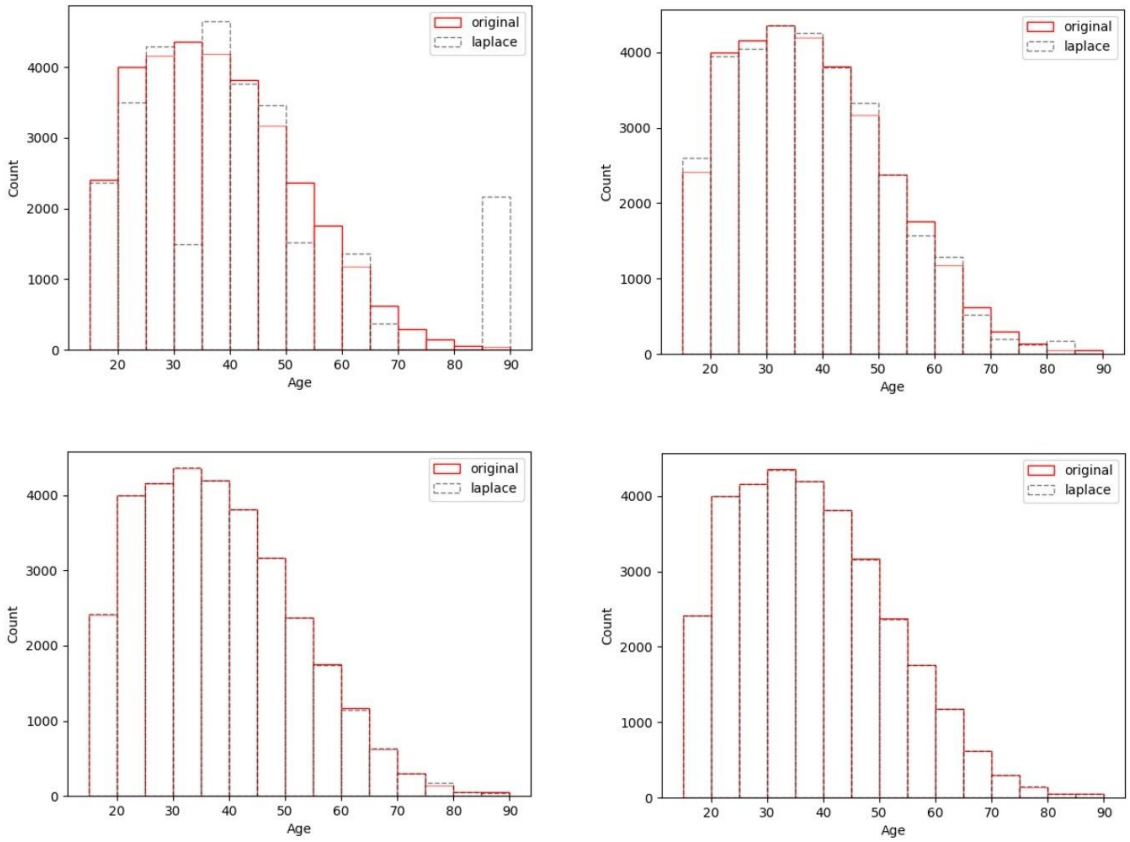
Diferencijalna privatnost je otporna na napade povezivanja i praćenja koji koriste pomoćne informacije kako bi se napad uspješno izveo. Također, kompozicijska je, što znači da možemo odrediti gubitak privatnosti izvođenjem dviju različito privatnih analiza na istim podacima jednostavnim zbrajanjem pojedinačnih gubitaka privatnosti za dvije analize.

Nedostatak ove tehnologije je to što mehanizmi, standardi i literature vezane uz nju nisu lako dostupne, budući da je tehnologija relativno novija. Također, izazov je održavanje kompromisa između privatnosti i korisnosti, opisan iznad.

4.8.7. Testiranje

Kodovi vezani uz diferencijalnu privatnost su jako kompleksni te je teško naći neki relevantan test koji bi mogao pokazati nešto korisno vezano uz temu rada. Unatoč tome, izdvajam jedan jednostavan kod, preuzet s Interneta [56], koji prikazuje Laplaceov mehanizam diferencijalne privatnosti. Program generira različito privatni histogram za dani skup podataka. Predmet testiranja u ovom programu je kako vrijednost epsilon utječe na privatnost i točnost (iskoristivost) podataka.

Na slici 4.13. prikazani su rezultati pokretanja programa za vrijednosti epsilon: 0.001 (gore lijevo), 0.01 (gore desno), 0.1 (dolje lijevo) te 0.5 (dolje desno). Prikazuje se originalni histogram te histogram gdje je podacima dodan šum s Laplaceovim mehanizmom.



Slika 4.13. Utjecaj vrijednosti ε na privatnost podataka

Iz primjera pokretanja programa s različitim vrijednostima ε se može zaključiti ono što je napomenuto u potpoglavljima iznad ovoga, a to je da se s malom vrijednošću ε povećava šum te je samim time veća privatnost podataka, a lošija iskoristivost. Također, vrijedi i suprotno, što je vrijednost ε veća, to je šum u podacima manji, s čime je i privatnost manja, dok je točnost te iskoristivost podataka veća, kao što se može vidjeti na grafu dolje desno.

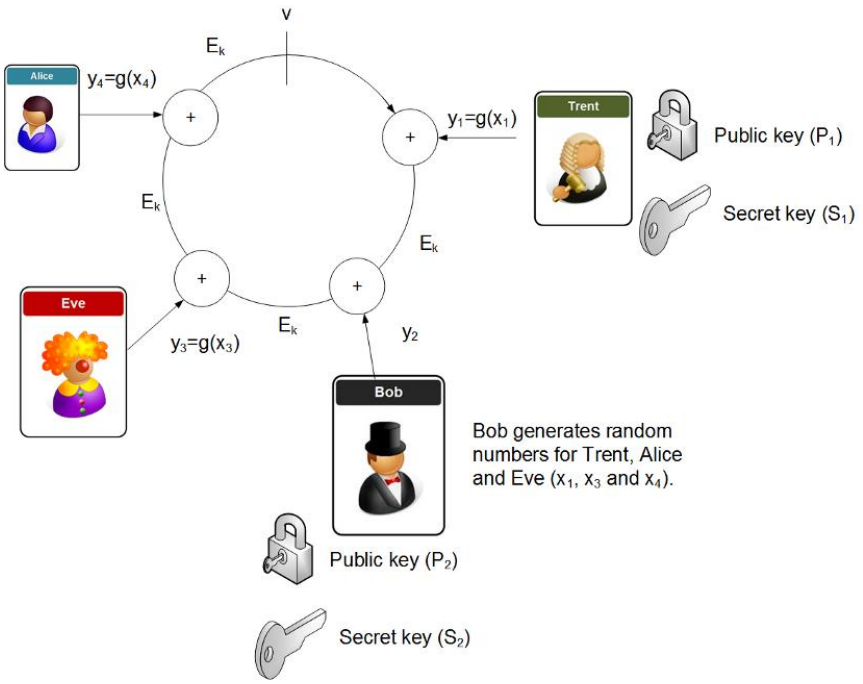
4.9. Ring signatures

Ring signatures, odnosno prstenasti potpisi, su vrsta grupnog potpisa koja štiti anonimnost korisnika. U ovoj tehnologiji postoji skupina javnih ključeva, a potpisnik zna privatni ključ koji odgovara javnom ključu u skupini javnih ključeva (zna samo jedan). Na taj način može koristiti ovaj skup javnih ključeva i odgovarajući privatni ključ za generiranje potpisa prstena. Verifikator potpisa može samo potvrditi da potpis dolazi iz ovog skupa potpisa, ali ne zna tko je potpisao potpis. Za verifikaciju potpisa potrebni su samo javni ključevi svih članova.

4.9.1. Princip rada

Opći model prstenastih potpisa ima tri koraka. prvi je generiranje ključa, polinomijalni vremenski algoritam s ulaznim parametrom k i dva izlaza, javnim ključem pk i privatnim ključem sk. Generira se javni ključ pk_i i privatni ključ sk_i kao par ključeva za svakog potpisnika C_i. Javni i privatni ključevi različitih korisnika mogu dolaziti iz različitih sustava javnih ključeva, kao što su RSA, DLP i ECSDA. Drugi korak je algoritam potpisivanja, polinomijalni vremenski algoritam. Potpis s na poruci m generira se nakon unosa poruke m, javnih ključeva članova prstena $pk_1,pk_2,...,pk_n$ i privatnog ključa sk_i njegovog vlasnika. Neki parametri u potpisu su kružni prema određenim pravilima. Zadnji korak, odnosno algoritam u općem modelu prstenastog potpisa je provjera potpisa, deterministički algoritam. U njemu se nakon unosa potpisa, poruke i javnih ključeva članova u shemi prstenastog potpisa ispisuje ispis koji će biti istinit u slučaju da je prstenasti potpis verificiran. U suprotnom će se ispisati *false*. [57]

Uzmimo za primjer da su 4 osobe u grupi i svaka od njih ima svoj javni i tajni ključ. Jedna od osoba želi potpisati poruku. Ta osoba inicijalno generira slučajnu vrijednost v i slučajne vrijednosti x_i za svakog sudionika te uzima svoj tajni ključ s_i kojeg koristi za određivanje drugog tajnog ključa, a koji je reverzan funkciji enkripcije. Zatim uzima poruku i njezin hash i time stvara ključ k koji se koristi sa simetričnom enkripcijom za šifriranje svakog elementa prstena E_k. Svaki element prstena koristi OR funkciju iz prethodnog elementa. Svaka od nasumičnih vrijednosti za druge sudionike zatim se šifrira javnim ključem danog sudionika. Osoba koja potpisuje poruku zatim izračunava vrijednost y_s kako bi stvorio prsten (rezultat prstena mora biti jednak generiranoj slučajnoj vrijednosti v). On će zatim napraviti inverz te vrijednosti kako bi proizveo ekvivalentan privatni ključ x_s. Osoba sada oslobađa cjelokupni potpis i nasumične x vrijednosti, zajedno s izračunatim tajnim ključem, a za provjeru potpisa primatelj samo izračunava prsten i provjerava da rezultat odgovara poslanom potpisu. Ilustracija opisanog postupka nalazi se na slici 4.14. [58]



Slika 4.14. Prikaz prstenastog potpisa [58]

Osnovni koraci su:

1. Generirati ključ za šifriranje k=hash(poruka)

2. Generirati originalnu slučajnu vrijednost u od potpisnika

3. Šifrirati u da se dobije šifrirana zasljepljujuća slučajna vrijednost v = $E_k(u)$

4. Za svaku osobu osim potpisnika:

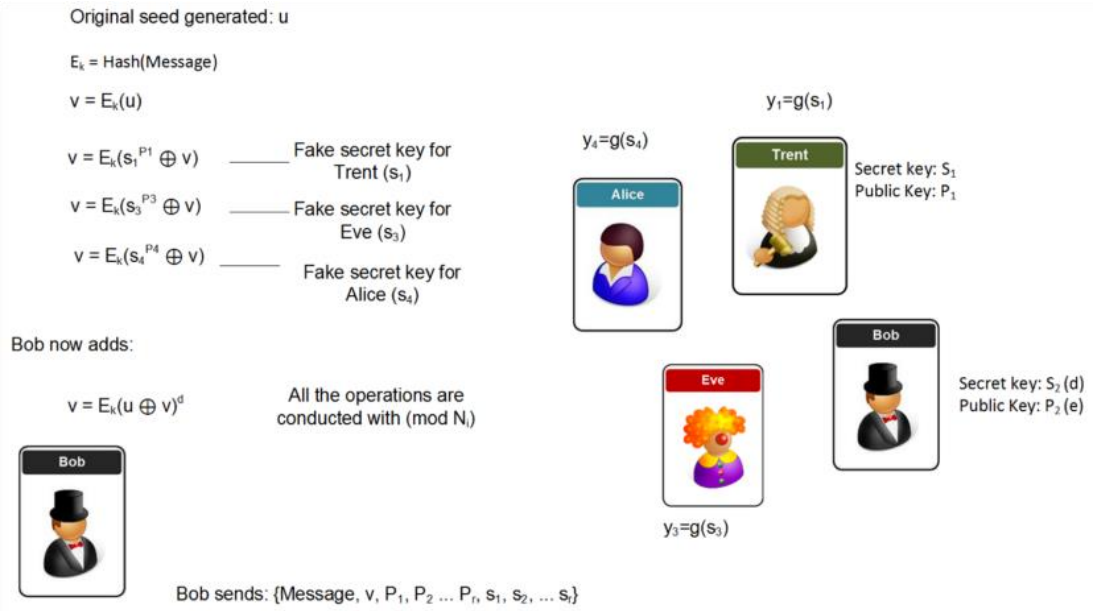
    4.1 Izračunati e = $s_i^{P_i} \ mod \ N_i$, gdje je s_i nasumični broj generiran za lažni tajni ključ i-te strane, P_i je javni ključ i-te strane, a N_i duljina ključa k i-te strane

    4.2 Izračunati v = v⊕e, odnosno napraviti operaciju XOR nad e i v

5. Za stranu koja potpisuje z, izračunati tajni ključ $s_z = (v \oplus u)^d \ mod \ N_z$, gdje je d tajni ključ potpisnika, a N_z duljina ključa k potpisnika

6. Verificirati ako je zasljepljujuća vrijednost v jednaka originalnoj slučajnoj vrijednosti potpisnika u: $v = E_k\{(v \oplus u)^d\}^e \rightarrow u$

Na slici 4.15. se nalazi prikaz osnovnih koraka algoritma prstenastih potpisa. [58]



Slika 4.15. Osnovni koraci algoritma prstenastih potpisa [58]

4.9.2. Sigurnosni zahtjevi

Pod pretpostavkom da javni ključevi članova prstena L={PK_1, PK_2,..., PK_n} i privatni ključ člana SK_i generiraju potpis R za poruku m, tada bi R trebao imati sljedeće sigurnosne zahtjeve:

1) Bezuvjetna anonimnost: čak i ako napadač nezakonito dobije sve privatne ključeve, vjerojatnost da može odrediti pravog potpisnika nije veća od 1/n, gdje je n broj članova prstena.

2) Nefalsifikat: napadač ne zna privatni ključ nijednog člana; čak i ako je potpis R dobiven od nasumičnog pogađanja, vjerojatnost da će napadač krivotvoriti legalni potpis je zanemariva.

3) Ispravnost: ukoliko je potpis potpisan strogo u skladu s procedurom potpisivanja i potpis nije falsificiran tijekom propagacije, tada prstenasti potpis zadovoljava provjeru potpisa. [57]

4.9.3. Primjena

Primjena prstenastih potpisa u blockchainu može ostvariti potrebe zaštite privatnosti obje strane interakcije. Primatelj ne može znati tko je poslao informaciju, potrebno je samo potvrditi da podaci nisu mijenjani prema potpisu.

Prva primjena prstenastih potpisa bila je na CryptoNote kako bi se sakrilo podrijetlo transakcija. CryptoNote je evolucija Bitcoina, koja može zaštititi privatnost identiteta platitelja i primatelja transakcije. U CryptoNoteu se transakcija potpisuje i verificira prstenastim potpisom, a verifikatori mogu samo osigurati da njen potpisnik pripada određenom korisničkom skupu, ali ne mogu razlikovati njegov stvarni identitet. Platitelj generira jednokratni ključ za svaku transakciju (nakon svake transakcije se generira novi javni ključ), a samo primatelj može povratiti odgovarajući privatni ključ. CryptoNote implementira to da nijedna treća strana ne može utvrditi jesu li dvije transakcije poslane istom korisniku, što rezultira vanjskom nevidljivošću adrese korisnika. [17]

Dalje, Monero je iskoristio prstenasti potpis i jednokratnu jedinstvenu adresu u CryptoNoteu kako bi proširio povjerljive transakcije na povjerljive prstenaste transakcije (engl. ring CT). RingCT je predstavio tehniku koja se zove višeslojni povezivi spontani anonimni grupni potpis kako bi kombinirao Pedersenovu obvezu s prstenastim potpisima. Monero je time postigao skrivene iznose transakcija. Konkretno, koristi prstenaste potpise i jednokratne adrese za prekid veze između ulazne i izlazne adrese u svakoj transakciji te povjerljive transakcije za skrivanje iznosa. [17]

Primjena korištenja prstenastih potpisa može biti i u Bitcoinu, gdje svatko tko zna privatni ključ iz skupine javnih ključeva može potpisati i potrošiti novčiće zaključane u pametnom ugovoru, a nitko ne zna tko je potpisao, čak ni članovi grupe.

Prstenasti potpisi mogu se koristiti u e-glasanju temeljenom na blockchain tehnologiji, gdje birač potpisuje svoj glas u ime svih ljudi s pravom glasa te pri tome dokazuje da se je registrirao i da ima pravo glasa, ali bez otkrivanja svog glasa. Također, korisnik može dokazati blockchain aplikaciji da je registriran bez otkrivanja tko je.

4.9.4. Prednosti i nedostatci

Kao što je navedeno kod sigurnosnih zahtjeva, prednosti su bezuvjetna anonimnost i nemogućnost napadača da krivotvori potpis. Također, u ovoj tehnologiji nema potrebe za pouzdanom trećom stranom.

Nedostatak je moguće otežano upravljanje i koordinacija nekoliko entiteta potpisnika. Također, veličina rezultirajućeg potpisa raste linearno s veličinom ulaza, odnosno s brojem javnih ključeva. To znači da su sheme prstenastih potpisa neizvedive za stvarne slučajeve

upotrebe s jako velikim n (na primjer, e-glasovanje s milijunima sudionika), ali za neke aplikacije s relativno malom i srednjom veličinom unosa ova tehnologija je prihvatljiva. Osim toga, iako su u razvoju neke sheme prstenastih potpisa koje bi trebale biti otporne na kvantna računala, većina postojećih shema prstenastih potpisa su ugrožene s budućim kvantnim računalima.

4.9.5. Testiranje

Testiran je kod preuzet s Interneta [59], napisan u python programskom jeziku. Za pokretanje programa potrebno je instalirati knjižnicu ecpy. Program prikazuje implementaciju prstenastog potpisa za proizvoljno odabranu poruku. U slučaju testa postoje dva prstena, svaki s 4 ključa, a poruka se potpisuje s jednim ključem iz svakog prstena. U slučaju kada se poruka potpisuje s ispravnim ključem, provjera je uspješna te program to ispisuje. U slučaju kada zamijenimo jedan od privatnih ključeva, verifikacija je negativna. Ispis pokretanja programa je na slici 4.16.

```
Message to sign:  Hello 123

public2 in ring1:  ECPublicKey:
  x: 352a46ad003617adc28d22933abc4e9a179f9cd022a3e1e3d8cc09c6adec21b0
  y: e0e648bc11f11eef74ab862be3b422e16479741f2e780951afe776a3dc098eb6
Secret2 in ring1:  ECPrivateKey:
  d: 9f09624ce19d33e57ffc30c9321930e61e85ad041979479777a0b1cf244fd915
public2 in ring2:  ECPublicKey:
  x: d7f90be66378d08b8ba1a6e4a8d918c45b4e93e384367d155aa50609df84c94
  y: 31ca4282bd7d9eda617d93c07b998f852f5177252716e3e228a73989ae16d55c
Secret2 in ring2:  ECPrivateKey:
  d: 254217cf65dc889590424d5e0c395a50e70c90b66434e409a3bb796d5219281d

Checking with valid key (Key 2 in Ring 1, Key 2 in Ring 2):  True

Now let's replace one onf the private keys (secret2):

public2 in ring1:  ECPublicKey:
  x: 352a46ad003617adc28d22933abc4e9a179f9cd022a3e1e3d8cc09c6adec21b0
  y: e0e648bc11f11eef74ab862be3b422e16479741f2e780951afe776a3dc098eb6
Secret2 in ring1:  ECPrivateKey:
  d: c2843b4a7848f801edd14c6f056f4a9d72ba0475813a3e251a098b72d58d0fa1
public2 in ring2:  ECPublicKey:
  x: d7f90be66378d08b8ba1a6e4a8d918c45b4e93e384367d155aa50609df84c94
  y: 31ca4282bd7d9eda617d93c07b998f852f5177252716e3e228a73989ae16d55c
Secret2 in ring2:  ECPrivateKey:
  d: 254217cf65dc889590424d5e0c395a50e70c90b66434e409a3bb796d5219281d

Checking with non-valid key (Key 2 in Ring 1) and valid Key 2 in Ring 2:  False
```

Slika 4.16. Testiranje prstenastog potpisa

4.10. Dandelion

Dandelion je protokol koji je rješenje za mrežni sloj blockchain tehnologije s ciljem poboljšanja privatnosti na P2P mreži. Svrha ovog protokola je spriječiti deanonimizaciju korisnika i povezivanje njihove IP adrese s pseudonimom. Izvorna verzija protokola je Dandelion, a novija verzija je Dandelion++. Izvorni Dandelion oslanjao se na tri idealizirane pretpostavke, a to su da svi čvorovi poštuju protokol, da svaki čvor generira točno jednu transakciju te da svi Bitcoin čvorovi pokreću Dandelion. Te pretpostavke u praksi nisu funkcionirale, stoga ih je Dandelion++ pokušao riješiti. [60]

4.10.1. Izvorni Dandelion

Izvorni Dandelion protokol radi u dvije faze, stem i fluff fazi.

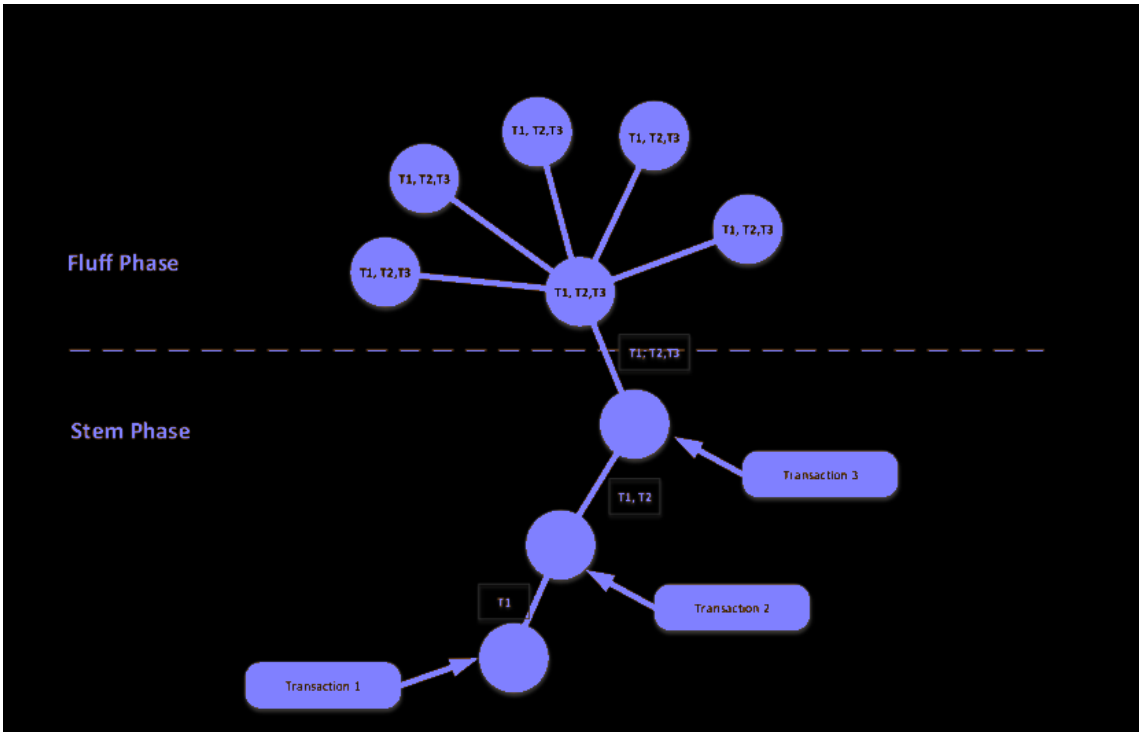
Stem faza je faza anonimnosti u kojoj je protokol dizajniran da smanji mogućnost preslikavanja na IP adresu izvornog čvora. U ovoj fazi, umjesto da čvor emitira transakciju svim svojim povezanim peerovima, on prenosi transakcijsku poruku kroz graf privatnosti do jednog slučajnog peera na temelju algoritma. Nakon toga, taj čvor samo prenosi transakcijsku poruku drugom ravnopravnom uređaju, a uzorak se nastavlja sve dok konačno (i nasumično) jedan od čvorova ne odašilje poruku u tipičnom formatu širenja ostatku mreže. Odluka o tome kada će se dogoditi korak u kojem određeni čvor nastavlja širenje u tipičnom formatu se donosi "bacanjem novčića", odnosno svaki čvor, kada prima transakciju, igra malu igru vjerojatnosti koja daje 90% šanse da transakcija ostane privatna, tj. "nastavi duž stabljike (stem faze)". To se nastavlja dok jedan od čvorova ne baci "fluff" umjesto "stem" i odmah javno emitira transakciju ili do vremenske odgode, koja je probabilistička te određena pojedinačno za svaki matični čvor koji drži transakciju. Vremenska odgoda sprječava da prvi čvor koji primi transakciju bude onaj koji ju emitira. [60]

Druga faza je fluff faza u kojoj, kao što je već napomenuto, čvor normalno širi transakciju putem difuzije i gura ju u svim smjerovima preko P2P mreže. Tako se transakcijska poruka se brzo propagira do većine čvorova u mreži. Međutim, postaje mnogo teže pratiti natrag do izvornog čvora budući da je transakcijska poruka prebačena na mnogo pojedinačnih čvorova kroz graf privatnosti prije nego što je propagirana na način koji bi omogućio promatraču da je mapira na jedan čvor. Umjesto toga, promatrač je mogao samo preslikati širenje transakcija

natrag na nekoliko čvorova gdje je poruka prenesena u stem fazi, čime se zbunjuje stvarni identitet pošiljatelja. Zapravo, ovo je na neki način slično načinu na koji prstenasti potpis prikriva stvarnog potpisnika transakcije. [60]

Primarni problemi s izvornim Dandelion protokolom proizlaze iz njegovog podcjenjivanja specifičnih vrsta protivnika zbog pretpostavki o njihovom ograničenom znanju. [60]

Kada se ovaj protokol nacrta kao dijagram, vizualni proces je sličan maslačku s dugačkom stabljikom i pahuljastom glavom, po čemu je protokol i dobio ime. Prikaz "maslačka" može se vidjeti na slici 4.16. [61]



Slika 4.17. Ilustracija Dandelion protokola kao maslačka [61]

4.10.2. Dandelion++

Dandelion++ se posebno usredotočuje na suptilne promjene izbora implementacije Dandeliona kao što su topologija grafikona i mehanizmi za prosljeđivanje poruka. Ovaj noviji protokol se oslanja na povećanje količine informacija koje protivnici moraju naučiti kako bi deanonimizirali korisnike.

Dandelion++ se značajno razlikuje od Dandeliona u svojoj matičnoj, stem fazi, u kojoj prosljeđuje transakcije preko isprepletenih putova, poznatih kao kabeli, prije nego što prenese transakcijsku poruku na mrežu. Kabeli mogu biti fragmentirani, ali njegova intuicija u odabiru čvora do kojeg će se širiti još uvijek je ograničena na njegovo lokalno susjedstvo. Ovo je važno razmatranje kada se uspoređuju rješenja anonimnosti na mrežnoj razini kao što je Tor, koji je protokol za usmjeravanje na luk, gdje klijenti trebaju globalne, trenutne informacije o mreži za određivanje putova transakcija. [60]

Obje verzije Dandeliona rade u asinkronim ciklusima, u kojima svaki čvor napreduje kada njegov unutarnji sat dosegne određeni prag. Za svaki period, Dandelion++ ima 4 primarne komponente s nekim optimizacijama:

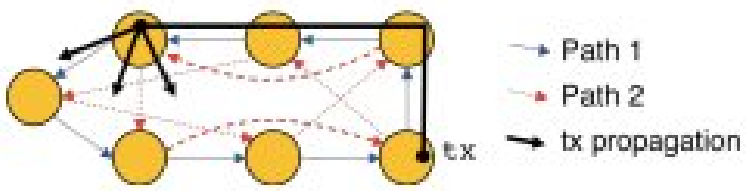
1) Grafikon anonimnosti - koristi nasumični četiri-regularni grafikon umjesto linearnog grafikona za fazu anonimnosti, a izbor Dandelion++ releja po čvorovima je neovisan o tome podržavaju li njihovi susjedi Dandelion++ ili ne

2) Prosljeđivanje vlastite transakcije - događa se svaki put kada čvor generira vlastitu transakciju te prosljeđuje transakciju duž istog izlaznog ruba u 4-regularnom grafu; ovo se razlikuje od jedne od problematičnih pretpostavki u Dandelionu gdje se pretpostavlja da čvorovi generiraju samo jednu transakciju

3) Prosljeđivanje transakcije (relej) - trenutak vjerojatnosti u stem fazi u kojem čvor prima matičnu transakciju i odabire prijenos transakcije ili je širi u mrežu; izbor širenja transakcija na mrežu je pseudoslučajan, a čvor je ili difuzor ili relejni čvor za sve relejne transakcije

4) Fail-Safe mehanizam - za svaku transakciju stem faze, svaki čvor prati vidi li se ponovno kao transakcija fluff faze; ako ne, čvor širi transakciju [60]

Prikaz ovog protokola može se vidjeti na slici 4.18. u kojoj se poruka širi po jednom (plavom) od dva iscrtana puta na grafu, a zatim se širi difuzijom. [60]



Slika 4.18. Širenje poruke u Dandelion++ protokolu [61]

4.10.3. Prednosti i nedostatci

Koristeći Dandelion++ se otežava mapiranje IP adresa promatranjem širenja transakcijskih poruka. Nadalje, Dandelion++ ne povećava značajno kašnjenje mreže, a njegova praktična izvedivost je demonstrirana na glavnoj mreži Bitcoina. Protokol je lagan, budući da ne uključuje komplicirano izračunavanje i može se implementirati bez ikakvih promjena u temeljnom Bitcoin sustavu. Pruža učinkovit te cjenovno prihvatljiv alat za anonimnost mrežnog sloja i za smanjenje mogućnosti mapiranja napada za deanonimizaciju korisnika. Također, dok druga rješenja za privatnost imaju za cilj zaštititi pojedinačne korisnike, Dandelion štiti anonimnost ograničavajući sposobnost protivnika da deanonimiziraju cijelu mrežu.

Unatoč svojim prednostima, Dandelion++ ne štiti eksplicitno od protivnika na razini ISP-a ili AS-a (autonomnih sustava) koji mogu koristiti napade usmjeravanja za deanonimizaciju korisnika. [60]

5. USPOREDBA TEHNOLOGIJA ZA POBOLJŠANJE PRIVATNOSTI NA BLOCKCHAINU

U prethodnom poglavlju opisane su, analizirane i testirane trenutno najbolje tehnologije za poboljšanje privatnosti na javnim blockchain mrežama. Kako bi usporedba opisanih tehnologija bila preglednija, u nastavku su dvije tablice koja daju prikaz usporedbe, budući da bi jedna tablica bila nepregledna s obzirom na broj opisanih tehnologija.

Tablica 5.1. Usporedba tehnologija za poboljšanje privatnosti na blockchainu

Tehnologija	**Zero-knowledge proof (ZKP)**	**Zk-SNARK**	**Zk-STARK**	**Bulletproofs**	**Povjerljive transakcije**
Opis	Jedna strana dokazuje istinitost određenih informacija drugoj strani bez otkrivanja bilo kakvih dodatnih informacija	Neinteraktivni ZKP, koristi algoritam uparivanja eliptičnih krivulja za dokazivanje znanja	Neinteraktiv-ni ZKP, sličan zk-SNARKu, uz transparentnost i bez pouzdanog postavljanja; oslanja se na hash vrijednosti	Neinteraktivni ZKP, temelji se na povjerljivim transakcijama i dokazima raspona	Protokol koji skriva iznos i adresu transakcije, temelji se na Pedersenovim obvezama i faktorima zasljepljivanja
Tipovi	Interaktivni, neinteraktivni	/	/	/	Prstenaste povjerljive transakcije (ring CT)
Svojstva	Potpunost, ispravnost, nulto znanje	Nulto znanje, sažetost, neinteraktiv-nost, argumenti	Nulto znanje, skalabilnost, transparentnost, argumenti	Nulto znanje, neinteraktiv-nost, skriva iznose transakcije	Skriva iznos i adresu transakcije
Veličina transakcije	Velika	Velika, linearno raste s porastom veličine izračuna	Srednja, srednje raste s porastom veličine izračuna	Mala, neznatno raste s porastom veličine izračuna	Povećava se s dodavanjem CT protokola

Vrijeme dokaza	Veliko	Srednje, veće od zk-STARK; do nekoliko sekundi	Jako malo, u milisekundama	Veće od zk-SNARK i zk-STARK	/
Veličina dokaza	Velika	Jako mala, u bajtima	Velika, do 250 Kb	Veća od zk-SNARK, ali manja od zk-STARK; do 1 Kb	/
Vrijeme provjere	Veliko	Jako malo, u milisekundama	Srednje, u sekundama	Veće u odnosu na zk-SNARK i zk-STARK	/
Otpornost na kvantno računarstvo	Ne	Ne	Da	Ne	Ne
Primjena	Provjera valjanosti transakcija, aplikacije za razmjenu poruka, sigurno dijeljenje dokumenata, kontrola pristupa informacijama	Zerocash – skriva se sadržaj transakcija (iznos i adrese) Zcash – provjera valjanosti transakcije	Zk-Rollups pametni ugovori	Monero i MimbleWimble – smanjenje veličine transakcije	Monero i MimbleWimble, provjera da novčići nisu stvoreni od nule Liquid Network
Prednosti	Jednostavnost, nulto znanje, privatnost	Anonimnost informacija o transakciji, učinkovitost, jeftina provjera	Ne zahtjeva pouzdane postavke, transparentnost	Učinkovitost, sigurnost, bez pouzdanih postavki, ušteda prostora i troškova	Povjerljivost podataka, privatnost, ne zahtijeva pouzdane postavke
Nedostatci	Vrijeme provjere valjanosti transakcija, troškovi provjere dokaza, količina računalne snage - neučinkovitost	Zahtjeva se pouzdan proces postavljanja	Veći troškovi	Ranjivost na kvantne napade, veće vrijeme provjere i dokaza	Problemi u kompatibilnosti s Bitcoin mrežom, problemi ukoliko iduća transakcija nije povjerljiva

Tablica 5.2. Usporedba tehnologija za poboljšanje privatnosti na blockchainu

Tehnologija	**Homomorfna enkripcija**	**Secure multi-party computation**	**Diferencijalna privatnost**	**Ring signatures**	**Dandelion**
Opis	Omogućuje izvođenje operacija nad podacima dok su šifrirani	Više strana izračunava izlaz kombinirajući svoje pojedinačne ulaze, pri čemu ne vide podatke drugih strana	Metoda koja uvodi određenu količinu nasumičnog šuma u podatke kako bi se onemogućila analiza podataka	Potpis poruke s privatnim ključem i N javnih ključeva kako bi se sakrio pravi potpis	Odašiljanje poruka čvorovima na način da se ne mogu povezati s izvornim čvorom
Tipovi	Parcijalna, donekle, potpuno	/	Globalna i lokalna	/	Dandelion, Dandelion++
Svojstva	/	Privatnost ulaza, ispravnost	Nema naknadne obrade, kompozicija	Bezuvjetna anonimnost, nefalsifikat, ispravnost	Vizualni proces sličan maslačku, Stem i fluff faza
Otpornost na kvantno računarstvo	Potpuna homomorfna enkripcija - da	Da	Ne	Ne	/
Primjena	U zk-SNARK metodi, šifriranje transakcije, prodavanje bitcoin adrese, medicina, e-glasovanje itd.	Upravljanje imovinom i privatnim ključevima, u budućnosti – dijeljenje podataka između organizacija	Dodavanje šuma podacima prije prijenosa poruke, perturbacija grafa korisnika, integracija s blockchain računarstvom u oblaku itd.	CryptoNote - skrivanje podrijetla transakcija, Monero – ring CT, e-glasanje, registracija u aplikaciju bez otkrivanja identiteta	Bitcoin - otežavanje mapiranja IP adresa promatranjem širenja transakcijskih poruka
Prednosti	Zaštita privatnosti, nema ciljanja od zlonamjernika niti nadzora (npr. od strane vlade)	Treća strana ne vidi podatke, usklađenost s GDPR-om (prekogranični prijenos), manje računalne snage od homomorfne enkripcije	Otpornost na napade povezivanja i praćenja, kompozicija	Nemogućnost krivotvorenja potpisa, anonimnost	Ne povećava kašnjenje mreže, lagan protokol, ne mijenja temeljni Bitcoin, cjenovno prihvatljiv, štiti cijelu mrežu

Nedostatci	Troškovna neučinkovitost u odnosu na nešifrirane podatke, spori algoritmi, visoki zahtjevi za pohranu podataka	Računalni troškovi, sporije vrijeme izvođenja, troškovi komunikacije, zahtijeva većinu iskrenih sudionika	Slaba dostupnost mehanizama i literature, izazov održavanja kompromisa između privatnosti i korisnosti	Teža koordinacija nekoliko entiteta potpisnika, linearan rast veličine potpisa s veličinom ulaza (brojem javnih ključeva)	Ne štiti eksplicitno od protivnika na razini ISP-a ili AS-a koji koriste napade usmjeravanja

6. ZAKLJUČAK

Blockchain je naširoko korišten u raznim područjima zbog svojih specifičnih svojstava koji su navedeni u radu. Unatoč brojnim prednostima koje blockchain tehnologija pruža sama po sebi, neka njezina svojstva otežavaju učinkovitu zaštitu privatnosti korisnika te se zbog toga nailazi na mnogo problema koji ugrožavaju korisnikovu privatnost, povjerljivost i anonimnost. U radu su predstavljeni problemi deanonimizacije, upravljanja ključevima, pametnih ugovora, napada na peer-to-peer mrežu i drugi. Iz tog razloga potrebno je imati tehnologije koje štite blockchain i otklanjaju te probleme, nalazeći rješenja za njih. Na temelju navedenih problema, u radu su opisane i analizirane trenutne tehnike za očuvanje privatnosti na blockchainu. Kako vrijeme odmiče, a istraživanja napreduju, tehnologije koje poboljšavaju privatnost na blockchainu se više ne baziraju na rješenjima za privatnost transakcija kod kriptovaluta, već se traže rješenja i smišljaju tehnologije za poboljšanje privatnosti šire primjene blockchaina. Tako su s jedne strane predstavljene različite tehnologije dokaza nultog znanja koje, po mom mišljenju, pretežito imaju potencijal za privatnost transakcija, dok su s druge strane predstavljene ostale tehnologije za zaštitu privatnosti koje rješavaju različite probleme, od deanonimizacije, upravljanja privatnim ključevima pa do problema s P2P mrežom. U tehnologijama poput homomorfne enkripcije i diferencijalne privatnosti vidim veliki potencijal za široki spektar primjene pri zaštiti privatnosti na blockchain mreži.

Iako je blockchain tehnologija koja je stvorena prvotno za trgovanje kriptovalutama, mislim da će u budućim primjenama, s novim i nadolazećim tehnologijama ili s nadogradnjama postojećih tehnologija koje korisnicima jamče još veću privatnost i anonimnost te zaštitu podataka, blockchain tehnologija biti uvelike rasprostranjena tehnologija u svim aspektima današnjeg tehnološkog društva. Iz svega navedenog, smatram da blockchain može i da hoće u budućnosti iskoristiti svoj maksimalni potencijal koji ima.

Literatura

[1] Hayes, A.: "What Is a Blockchain?", s interneta, https://www.investopedia.com/terms/b/blockchain.asp#toc-how-does-a-blockchain-workhttps://www.google.com, 18. kolovoza 2022.

[2] Nepoznati autor: "Blockchain Fundamentals", s Interneta, https://businessblockchainhq.com/blockchain-fundamentals/, 23. kolovoza 2022.

[3] Baggetta, M.: "Why Cryptography Makes Blockchain Unstoppable", s Interneta, https://blockgeeks.com/guides/blockchain-cryptography/, 22. kolovoza 2022.

[4] Poston, H.: "Blockchain and asymmetric cryptography", s Interneta, https://resources.infosecinstitute.com/topic/blockchain-and-asymmetric-cryptography/, 22. kolovoza 2022.

[5] Saravanan, P.: "Demystifying the Wallet in Blockchain", s Interneta, https://medium.com/@saravananp/demystifying-the-wallet-in-blockchain-14f3971d44db, 23. kolovoza 2022.

[6] Satapathy, S.: "Immutability in blockchain", s Interneta, https://infinite-vigilant.github.io/blockchain.html?fbclid=IwAR0u6ha7wTNrQ4bubkiA8_GB8HaWiId_PddnFfXZZ5HwlltmDjGj3JI9_9k, 24. kolovoza 2022.

[7] Chauhan, A.: "Understand How a Blockchain Peer to Peer Network Works", s Interneta, https://betterprogramming.pub/understand-how-a-blockchain-peer-to-peer-network-works-565ecd34c6d2, 23. kolovoza 2022.

[8] Peng, L. i dr.: "Privacy preservation in permissionless blockchain: A survey", s Interneta, https://www.sciencedirect.com/science/article/pii/S2352864819303827, 18. kolovoza 2022.

[9] Margaria, T. i dr.: "PETchain: A Blockchain-Based Privacy Enhancing Technology", s Interneta, https://www.researchgate.net/publication/349938632_PETchain_A_Blockchain-Based_Privacy_Enhancing_Technology, 23. kolovoza 2022.

[10] Nepoznati autor: „Razlika izmedu privatnosti i povjerljivosti“, s Interneta, https://hr.gadget-info.com/difference-between-privacy, 11. kolovoza 2022.

[11] Nepoznati autor: „Privacy or anonymity? - Which is more important in the digital era?“, s Interneta, https://www.fifosys.com/blog/security/privacy-or-anonymity-which-is-more-important-in-the-digital-era/, 12. kolovoza 2022.

[12] Sobe, P.: „Privacy and Anonymity in the Internet“, s Interneta, https://www2.htw-dresden.de/~sobe/Basoti/Lectures/1_Intro.pdf, 12. kolovoza 2022.

[13] Conti, M. i dr.: "A Survey on Security and Privacy Issues of Bitcoin", s Interneta, https://arxiv.org/pdf/1706.00916.pdf, 27. kolovoza 2022.

[14] Nepoznati autor: "A New Attack Vector To Deanonymize Bitcoin Users", s Interneta, https://medium.com/decentralize-today/a-new-attack-vector-to-deanonymize-bitcoin-users-9c6dc433d4b6, 15. rujna 2022.

[15] Juhasz, P. i dr.: "A Bayesian approach to identify bitcoin users", s Interneta, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6292573/, 12. rujna 2022.

[16] Mosakheil, J.: "Security Threats Classification in Blockchains", s Interneta, https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1093&context=msia_etds, 2. rujna 2022.

[17] Bernabe Bernal, J. i dr.: "Privacy-Preserving Solutions for Blockchain: Rewiev and Challenges", s Interneta, https://www.researchgate.net/publication/336937331_Privacy-Preserving_Solutions_for_Blockchain_Review_and_Challenges, 25. kolovoza 2022.

[18] Lake, J.: "What is a preimage attack?", s Interneta, https://www.comparitech.com/blog/information-security/what-is-preimage-attack/, 6. rujna 2022.

[19] Shah, M.: "5 blockchain security issues and how to prevent them", s Interneta, https://www.fastcompany.com/90722111/5-blockchain-security-issues-and-how-to-prevent-them, 25. kolovoza 2022.

[20] Baivab, J.: "What is Phishing Attack? Definition, Types and How to Prevent it", s Interneta, https://www.simplilearn.com/tutorials/cryptography-tutorial/what-is-phishing-attack, 15. rujna 2022.

[21] Shalvey, K.: "A hacker stole more than $55 million in crypto after a bZx developer fell for a phishing attack", s Interneta, https://www.businessinsider.com/hacker-steals-55-million-in-crypto-after-bzx-phishing-attack-2021-11, 25. kolovoza 2022.

[22] Rahman, M. i dr.: "A Rewiev on Blockchain Security Issues and Challenges", s Interneta, https://www.researchgate.net/publication/354039550_A_Review_on_Blockchain_Security_Issues_and_Challenges, 26. kolovoza 2022.

[23] Powers, B.: "This Elusive Malware Has Been Targeting Crypto Wallets for a Year", s Interneta, https://www.coindesk.com/tech/2021/01/06/this-elusive-malware-has-been-targeting-crypto-wallets-for-a-year/, 7. rujna 2022.

[24] Sayees, S.: "Smart Contract: Attacks and Protections", s Interneta, https://www.researchgate.net/publication/338926064_Smart_Contract_Attacks_and_Protections, 13. rujna 2022.

[25] Huang, Y. i dr.: "Smart Contract Security: A Software Lifecycle Perspective", s Interneta, https://www.researchgate.net/publication/336446394_Smart_Contract_Security_A_Software_Lifecycle_Perspective, 13. rujna 2022.

[26] Nepoznati autor: "What Is a Replay Attack & How Does It Affect Blockchains?", s Interneta, https://learn.bybit.com/blockchain/what-is-a-replay-attack/, 8. rujna 2022.

[27] Nepoznati autor: "What Is a replay Attack?", s Interneta, https://academy.binance.com/en/articles/what-is-a-replay-attack, 8. rujna 2022.

[28] Nepoznati autor: "What Is Replay Attack? Things You Should Know", s Interneta, https://news.coincu.com/102304-replay-attack-things-you-should-know/, 15. rujna 2022.

[29] Palley, S.: "Understanding the risk of "immutable" blockchain applications", s Interneta, http://www.rmmagazine.com/articles/article//2018/10/08/-Understanding-the-Risk-of-Immutable-Blockchain-Applications-, 26. kolovoza 2022.

[30] Enwood, D.: "Zero-knowledge proofs-a powerful addition to blockchain", https://blockheadtechnologies.com/zero-knowledge-proofs-a-powerful-addition-to-blockchain/, s Interneta, 14. rujna 2022.

[31] Goldgrabe, Y: "On Interactive Proofs and Zero-Knowledge: A Primer", s Interneta, https://medium.com/magicofc/interactive-proofs-and-zero-knowledge-b32f6c8d66c3, 21. rujna 2022.

[32] Buchannan, B.: „Schnorr identification scheme“, s Interneta, https://asecuritysite.com/zero/schnorr, 7. listopada 2022.

[33] ConsenSys: "Introduction to zk-SNARKs", s Interneta, https://consensys.net/blog/developers/introduction-to-zk-snarks/, 21. rujna 2022.

[34] Geroni, D.: "An introduction to ZkSNARKs", s Interneta, https://101blockchains.com/zksnarks-introduction/, 20. rujna 2022.

[35] https://github.com/raphaelrrcoelho/zk-snarks, s Interneta, 15. listopada 2022.

[36] https://github.com/ETHorHIL/STARK_Py, 19. listopada 2022.

[37] Panther Team: "bulletproofs In Crypto - An introduction to a non-Interactive ZK Proof", s Interneta, https://blog.pantherprotocol.io/bulletproofs-in-crypto-an-introduction-to-a-non-interactive-zk-proof/, 21. rujna 2022.

[38] "Bulletproofs and Rangeproofs", https://asecuritysite.com/zero/go_bullet, s Interneta, 16. listopada 2022.

[39] Nepoznati autor: "A primer to Confidential Transactions", s Interneta, https://medium.com/@ecurrencyhodler/a-primer-to-confidential-transactions-e6ab3dd2bf1e, 24. rujna 2022.

[40] Buchannan, B.: „Elliptic Curve Pedersen Commitment“, https://asecuritysite.com/ecc/ecc_blind, s Interneta, 17. listopada 2022.

[41] Nepoznati autor: "Homomorphic encryption", s Interneta, https://en.wikipedia.org/wiki/Homomorphic_encryption, 27. rujna 2022.

[42] Erabelli, S.: "What is Homomorphic Encryption?", s interneta, https://dualitytech.com/what-is-homomorphic-encryption/, 27. rujna 2022.

[43] Nepoznati autor: "What is homomorphic encryption, and why isn't it mainstream?", s Interneta, https://www.keyfactor.com/blog/what-is-homomorphic-encryption/, 27. rujna 2022.

[44] Munjal, K.: "A systematic review of homomorphic encryption and its contributions in healthcare industry", s Interneta, https://link.springer.com/article/10.1007/s40747-022-00756-z, 28. rujna 2022.

[45] Lindell, Y.: "Secure Multiparty Computation (MPC)", https://eprint.iacr.org/2020/300.pdf, s Interneta, 28. rujna 2022.

[46] https://github.com/acmert/bfv-python, s Interneta, 17. listopada 2022.

[47] Zagakos, A.: „What is Homomorphic Encryption?“, https://www.freecodecamp.org/news/introduction-to-homomorphic-encryption/, s Interneta, 17. listopada 2022.

[48] Nepoznati autor: "What is Secure Multiparty Computation", https://inpher.io/technology/what-is-secure-multiparty-computation/, s Interneta, 28. rujna 2022.

[49] Nepoznati autor: "What are Signatures with Treshold or Treshold Signature?", https://academy.bit2me.com/en/what-are-the-threshold-signature/, s Interneta, 28. rujna 2022.

[50] Mrvosrvic, M.: "Secure Multi-Party Computation: applications within blockchain technology", s Interneta, https://eternacapital.medium.com/secure-multi-party-computation-applications-within-blockchain-technology-e07c727281e1, 28. rujna 2022.

[51] https://github.com/Dishikagoel/Shamirs-secret-sharing-algorithm-on-RSA-key-pair, s Interneta, 18.listopada 2022.

[52] Nepoznati autor: "Differential privacy", https://en.wikipedia.org/wiki/Differential_privacy, s Interneta, 1. listopada 2022.

[53] Shaistha, F.: "Differential Privacy-Noise adding Mechanisms", https://becominghuman.ai/differential-privacy-noise-adding-mechanisms-ede242dcbb2e, s Interneta, 1. listopada 2022.

[54] Can, M.: "Investigation and Application of Differential Privacy in Bitcoin", https://ieeexplore.ieee.org/document/9714372, s Interneta, 30. rujna 2022.

[55] Muneeb, H.: "Differential Privacy in Blockchain Technology: A Futuristic Approach", https://asset-pdf.scinapse.io/prod/2980202686/2980202686.pdf, s Interneta, 30. rujna 2022.

[56] https://github.com/clin366/differential_privacy, s Interneta, 18.listopada 2022.

[57] Liu, Y.: "Enhancing anonimity of Bitcoin based on ring signature alglorithm", https://ieeexplore.ieee.org/document/8288497, s Interneta, 4. listopada 2022.

[58] Buchanan, B.: "Ring signatures and anonymisation", https://medium.com/asecuritysite-when-bob-met-alice/ring-signatures-and-anonymisation-c9640f08a193, s Interneta, 3. listopada 2022.

[59] https://asecuritysite.com/signatures/ring_b, s Interneta, 19. listopada 2022.

[60] Curran, B.: "What is the Dandelion Protocol? Complete beginner's guide", https://blockonomi.com/dandelion-protocol/, s Interneta, 5. listopada 2022.

[61] Beam Privacy: "About Dandelion and Mimblewimble", https://medium.com/beam-mw/about-dandelion-and-mimblewimble-e083597e0355, s Interneta, 5. listopada 2022.

Sažetak

Blockchain, kao jedna od najnovijih, najzanimljivijih i najperspektivnijih tehnologija sadašnjice, nudi mnoštvo tema i sadržaja za istraživanje i analiziranje. U ovom radu opisana su svojstva i temelji blockchain tehnologije, ali isto tako analizirani su i problemi na koje navedena tehnologija nailazi u smislu privatnosti, povjerljivosti i anonimnosti korisnika. Zatim su opisane, analizirane i testirane tehnologije za poboljšanje privatnosti na javnim blockchain mrežama koje pokušavaju riješiti navedene probleme privatnosti, povjerljivosti i anonimnosti. Na samom kraju rada, prikazana je tablica usporedbe opisanih tehnologija za poboljšanje privatnosti koja daje pregled najbitnijih karakteristika od svake tehnologije.

Ključne riječi: blockchain, privatnost, anonimnost, povjerljivost, tehnologije za poboljšanje privatnosti

Abstract

Blockchain, as one of the newest, most interesting and promising technologies of today, offers a multitude of topics and content for research and analysis. In this paper are described the properties and foundations of blockchain technology and analyzed the problems faced by blockchain in terms of user privacy, confidentiality and anonymity. Next, privacy-enhancing technologies on public blockchain networks are described, analyzed and tested, which try to solve the mentioned problems of privacy, confidentiality and anonymity. At the very end of the paper, a comparison table of the described technologies for improving privacy is presented, which provides an overview of the most important characteristics of each technology.

Keywords: blockchain, privacy, anonymity, confidentiality, privacy enhancing technologies