Zero-knowledge proof v praksi

Vid Keršič, Martin Domajnko, Sašo Karakatič, Muhamed Turkanović

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, Slovenija
vid.kersic@um.si, martin.domajnko@student.um.si, saso.karakatic@um.si, muhamed.turkanovic@um.si

Z vse pogostejšo uporabo interneta in migracijo storitev iz fizičnega sveta v digitalni, postajajo vprašanja o varnosti, zasebnosti in digitalnem lastništvu osebnih podatkov vse pogostejša. Ena izmed ključnih tehnologij, ki omogoča razvoj rešitev na tem področju, so ničelno spoznavni dokazi (ang. zero-knowledge proofs, ZKP). ZKP so kriptografski protokoli, pri katerih dokazovalec dokaže pravilnost poljubne trditve preveritelju, ne da bi pri tem razkril dodatne informacije ali svoje podatke. V članku predstavimo ZKP protokole in njihove razlike, s posebnim poudarkom na dveh najpogostejših družinah protokolov: zk-SNARK in zk-STARK. Njihovo uporabno vrednost prikažemo na področju samo-upravljane in decentralizirane identitete ter na področju strojnega učenja. Pri decentralizirani identiteti ZKP omogočajo deljenje podatkov brez razkritja zasebnih informacij, medtem ko pri strojnem učenju omogočajo preverljivost izhodov modelov. To pomeni, da lahko uporabnik preveri, ali je bil za generiranje napovedi dejansko uporabljen pravilno izbran model.

Ključne besede:
zero-knowledge proof
kriptografija
decentralizirana identiteta
zasebnost
overjanje

Univerzitetna založba Univerze v Mariboru

ISBN 978-961-286-894-9
DOI https://doi.org/10.18690/um.feri.4.2024.17

1 Uvod

V zadnjih letih je vse več govora o zasebnosti, varnosti, suverenosti in lastništvu podatkov na spletu, čemur priča tudi pobuda za evropske digitalne denarnice, ki temeljijo ravno na teh principih [9]. Ena izmed jedrnih tehnologij, ki omogočajo razvoj takšnih rešitev, so ničelno spoznavni dokazi ali ZKP (ang. zero-knowledge proofs) [14]. ZKP so kriptografski protokoli, pri katerih dokazovalec (ang. prover) dokaže pravilnost določene trditve preveritelju (ang. verifier), ne da bi pri tem razkril kakršno koli informacijo o trditvi, razen njene pravilnosti. Zaradi te lastnosti je uporaba ZKP smiselna v številnih aplikacijah, saj razen dokazovanja informacij in celovitosti le teh, povečujejo tudi zasebnost in varnost uporabnikov ter varujejo njihove podatke. Čeprav so bili formalno definirani že leta 1985, so se v praksi začeli uporabljati šele v zadnjih letih, zahvaljujoč razvoju učinkovitejših metod in algoritmov za generiranje kriptografskih dokazov. K hitremu razvoju je prispevala tudi njihova uporaba v tehnologiji veriženja blokov (ang. blockchain), na primer v L2 omrežjih s t. i. zero-knowledge rollup.

V članku predstavljamo, definiramo in opisujemo ZKP protokole, s poudarkom na dvema najpomembnejšima družinama ZKP protokolov zk-SNARK in zk-STARK. Dodatno se osredotočamo na dva primera njihove uporabe, in sicer na področju decentralizirane identitete ter strojnega učenja. Pri prvem primeru je glavna uporaba deljenje podatkov, kjer uporabniki dokažejo veljavnost in vrednost določenih atributov, ne da bi pri tem razkrili svoje zasebne podatke ali celo konkretne vrednosti atributa. Eden najbolj razširjenih ZKP platforma za decentralizirano identiteto je Privado ID. Pri drugem primeru se ZKP uporabljajo za preverljivost izhodov modelov strojnega učenja, kjer se z uporabo ZKP omogoči preverjanje, če je bil za napovedovanje uporabljen določen model. Ta primer uporabe je še posebej aktualen v časih, ko vse bolj uporabljamo in plačujemo za storitve, ki vključujejo strojno učenje, pri čemer je nemogoče preveriti, ali je bil uporabljen model, za katerega plačujemo.

2 Zero-knowledge proofs

Področje kriptografije se je v zadnjem desetletju, še posebej v zadnjih letih, zelo razvilo in veliko kriptografskih protokolov, ki so v preteklosti bili le izvedljivi v teoriji, se danes lahko že učinkovito uporablja v sklopu dejanskih realnih aplikacijah. Medtem, ko se digitalno podpisovanje, simetrično in asimetrično šifriranje že vsakdan uporabljajo v digitalnem svetu, na primer pri obisku spletnih strani in podpisovanju elektronskih dokumentov, se naprednejši kriptografski protokoli kot so homomorfno šifriranje (ang. homomorphic encryption) in ničelno spoznavni dokazi (ZKP), ki so jedrna tema tega prispevka, šele začenjajo uporabljati v realnem svetu.

2.1. Splošno

ZKP je kriptografski protokol, ki poteka med dvema entitetama: *dokazovalcem* in *preveriteljem* [14]. S pomočjo protokola lahko dokazovalec prepriča preveritelja o posedovanju določenih podatkov/informacij brez razkritja podrobnosti le teh. Eden izmed najbolj znanih primerov je dokazovanje polnoletnosti, pri čemer ni razkrita dejanska starost osebe, temveč le trditev *starost osebe je večja od 18*. Čeprav so bili ZKP defninirani že v 80. letih prejšnjega stoletja, se je njihova uporabnost razširila v zadnjih letih, čemur je botrovalo več faktorjev, kot so finančne investicije, optimizacije orodij za generiranje kriptografskih dokazov in razvoj novih algoritmov ter pristopov.

Protokoli ZKP so definirani s tremi lastnostmi:

- **celovitost** (ang. completeness): preveritelj sprejme dokaz le, če ima dokazovalec res informacijo, in je bil protokol izveden pravilno;
- **trdnost** (ang. soundness): obstaja le zelo majhna verjetnost, da lahko dokazovalec prepriča preveritelja z napačno informacijo;

- **ničelno znanje** (ang. zero-knowledge): čeprav je preveritelj prepričan o veljavnosti dokaza, ne izve ničesar o dejanskih podatkih/informaciji.

Skozi leta raziskav je bilo razvitih več različnih sistemov oz. protokolov ZKP, ki se med seboj razlikujejo po načinu delovanja. Najpogosteje se delijo na (a) **interaktivne** in (b) **ne-interaktivne**, pri čemer se slednji pogosteje uporabljajo, saj ne zahtevajo komunikacije oz. izmenjave več sporočil med dokazovalcem in preveriteljem. Ker imajo ne-interaktivni precej več želenih značilnost kot interaktivni, pri čemer je najbolj želena nepotreba po izmenjavi več sinhronih sporočil med dokazovalcem in preveriteljem, se interaktivni protokoli pogosto s Fiat-Shamirjevo hevristiko preslikajo v ne-interaktivne [11].

Tudi znotraj kategorije ne-interaktivnih protokolov je bilo razvitih več podsistemov ZKP, pri čemer sta najbolj razširjena zk-SNARK in zk-STARK, ki sta predstavljena podrobneje v nadaljevanju. Protokoli ZKP se lahko uporabijo tudi za različne namene, na primer za selektivno razkritje podatkov v primeru deljenja elektronskih dokumentov, ali za delegiranje zahtevnejših programov na oddaljen računalnik, kjer želimo ohraniti preverljivost in pravilnost izvedbe programa (ang. verifiable computing), na primer izvajanje transakcij na L2 omrežjih pri tehnologiji veriženja blokov (ang. blockchain) (Ethereum).

2.2. zk-SNARK

zk-SNARK (ang. zero-knowledge succinct non-interactive argument of knowledge) so najbolj znana in razširjena družina protokolov ZKP [13]. Kot je že iz njihovega imena razvidno sta eni izmed najpomembnejših lastnosti **ne-interaktivnost** in **zgoščenost** (ang. succinctness). Slednja karakteristika izvira iz tega, da so dokazi "majhni" in njihova verifikacija hitra. Na praktičnem primeru to pomeni, da je preverjanje dokaza precej hitrejše kot generiranje samega dokaza oz. izvedba samega programa v primeru preverljivega računalništva.

Eden izmed pomembnejših korakov pri zk-SNARK je zaupanja vredna namestitev (ang. trusted setup), ki se izvede pred jedrnim delom protokola - dokazovanjem in preverjanjem [13]. Tekom zaupanja vredne namestitve se izračuna oz. definira skupen referenčni niz (ang. common reference string, CSR), ki se nato uporabi pri generiranju dokaza in preverjanju. CSR se navadno izračuna z večstranskim računanjem (ang. multi-party computation, MPC), saj le-ta omogoča varnejši način generacije, saj je manjša verjetnost, da bi vse osebe, ki sodelujejo v MPC, razkrile parametre uporabljene za generacijo CSR.

Skozi leta je bilo predstavljenih več zk-SNARK sistemov kot so ***Halo2***, ***Plonky2***, ***Aurora*** in ***Sonic***. Ti sistemi se med seboj razlikujejo v lastnostih CSR, na primer univerzalnosti in pouporabi niza za različne programe, učinkovitosti, uporabljenih matematičnih konstruktih itd.

2.3. zk-STARK

zk-STARK (ang. zero-knowledge scalable transparent argument of knowledge) so druga najbolj znana družina protokolov ZKP [3]. Razviti so bili nekaj let po zk-SNARK-ih z namenom rešitve njihovih pomanjkljivosti, predvsem potrebe po zaupanja vredni namestitvi. Za dokazovanje in preverjanje zk-STARK dokazov tako ni potrebe po CSR, zaradi česar so bolj transparentni [3]. zk-STARK sistemi prav tako vsebujejo druge kriptografske primitive zaradi česar so tudi odporni na kvantne računalnike in imajo boljše teoretične skalabilne lastnosti kot zk-SNARK-i (glej tabelo 1). Kljub temu so zaradi manjše razširjenosti in tega, ker so mlajši, trenutno še manj optimizirani in pogosto manj učinkoviti na realnih primerih.

V zadnjih letih je bilo razvitih tudi več zk-STARK sistemov, kot so ***Winterfell***, ***Stone*** od ***Starkware*** in ***RISC Zero***.

2.4. Primerjava

Čeprav zk-SNARK-i in zk-STARK-i omogočajo enake funkcionalnosti, se v določenih karakteristikah precej razlikujejo. Kot je bilo že omenjeno, prvi potrebujejo zaupanja vredno namestitev, čeprav so raziskovalci pred kratkim predstavili tudi zk-SNARK-e brez zaupanja vredne namestitve [22]. Kljub temu imajo zk-STARK-i več teoretičnih boljših lastnosti, kot so boljša časovna zahtevnost dokazovanja, kar se pozna še posebej pri večji količini vhodnih podatkov. Povzetek primerjave je predstavljen v Tabeli 1.

Tabela 1: Primerjava zk-SNARK in zk-STARK.

Lastnost	zk-SNARK	zk-STARK
Potrebna zaupanja vredna namestitev	Da (nekateri ne)	Ne
Odpornost na kvantne računalnike	Ne (nekateri da)	Da
Velikost dokaza	Majhna (več sto B)	Velika (več sto KB)
Zahtevnost dokazovanja	Kvazi-linearno	Logaritemska
Zahtevnost preverjanja	Logaritemska	Logaritemska

Aplikacije in sistemi temelječi na ZKP se razvijajo s pomočjo ogrodij (ang. framework), ki omogočajo definiranje tako imenovanih vezij (ang. circuit). Vezja so programi, ki kodirajo proces, ki ga želimo dokazati. Vnaprej je potrebno definirati vhodne podatke, izhodne podatke in sam program. Prav tako je potrebno definirati vidnost podatkov, na primer kateri vhodni oz. izhodni podatki naj bodo javni ali zasebni, kar pomeni, da niso razkriti preveritelju. Ogrodja, ki omogočajo razvoj sistemov ZKP so navadno poimenovana glede na to, kateri zk-SNARK ali zk-STARK protokol je uporabljen, na primer Halo2, Plonky2, itd.

3 Primeri uporabe

ZKP so našli uporabo na različnih področjih, kjer omogočajo izboljšano zasebnost, varnost in učinkovitost v digitalnih sistemih. Sledi nekaj pomembnih primerov uporabe:

- **Zasebnost pri kriptovalutah**: Zerocash, protokol za anonimne transakcije s kriptovalutami, uporablja ZKP za skrivanje ključnih informacij o transakcijah: znesek, identiteto pošiljatelja in prejemnika ter zgodovino transakcij. Sistem omogoča potrditev veljavnosti transakcije brez razkritja podrobnosti, podobno kot bi dokazali plačilno sposobnost brez vpogleda v celoten bančni izpisek. ZKP v primeru Zerocash protokola dokazuje sledeče štiri ključne elemente: zadostnost sredstev pošiljatelja, veljavnost transakcije, odsotnost ustvarjanja novega denarja ter prejem pravilnega zneska s strani prejemnika [2].
- **Overjanje v internetu stvari (IoT)**: V ekosistemih interneta stvari se lahko ZKP uporabljajo za anonimno komunikacijo in varno overjanje naprav, na primer za dokazovanje in verifikacijo identitete senzorjev. Zaradi računske kompleksnosti večina tradicionalnih ZKP algoritmov ni primernih za uporabo na majn zmoglivih IoT napravah. [5] predstavlja različne primere učinkovite in varne aplikacije ZKP algoritmov v IoT sistemih z anonimno komunikacijo.
- **Povečana prepustnost transakcij verig blokov**: ZKP igra ključno vlogo v rešitvah za povečanje zmogljivosti tehnologije veriženja blokov, še posebej pri tehnologiji Zero-knowledge rollups (ZK-rollups). Pri tehnologiji ZK-rollup se ZKP uporablja za preverjanje in združevanje večih transakcij izven verige, nato pa v glavno verigo pošljejo en sam dokaz o njihovi veljavnosti. Ta pristop bistveno poveča prepustnost transakcij, s čimer obravnava enega ključnih izzivov v skaliranju tehnologije veriženja blokov, hkrati pa ohranja varnostna jamstva osnovne verige blokov [8].

- **Verifikacija identitete**: ZKP omogočajo varno verifikacijo identitete brez razkrivanja občutljivih osebnih podatkov. Ta pristop posameznikom omogoča, da dokažejo specifične vidike svoje identitete, brez da s preveriteljem delijo dejanske vrednosti podatkov [10].
- **Varni sistemi glasovanja**: Sistemi elektronskega glasovanja lahko uporabljajo ZKP za zagotavljanje integritete glasovanja in zasebnosti volivcev. Ta tehnologija omogoča volivcem, da preverijo, ali je bil njihov glas pravilno preštet, ne da bi razkrili svojo izbiro, s čimer ohranja tako tajnost kot tudi preglednost glasovanja [1].
- **Zmanjšanje velikosti transakcij verig blokov**: Protokol Bulletproof bistveno izboljša velikost območnih dokazov, ki so linearno odvisni od števila elementov, v obstoječih predlogih za zaupne transakcije v Bitcoinu in drugih kriptovalutah. S tem zmanjša velikost transakcij, kar vodi do izboljšane učinkovitosti in razširljivosti v omrežjih verig blokov [4].

Tehnologija ZKP ima širok spekter uporabe, vendar smo se v tem delu posvetili dvema specifičnima področjema: decentralizirani identiteti in strojnemu učenju. Izbor teh dveh področij temelji na naših praktičnih izkušnjah in preverjenih rezultatih.

3.1. Decentralizirana identiteta

V hitro razvijajočem se okolju upravljanja digitalne identitete so se decentralizirane rešitve izkazale kot obetaven pristop k reševanju vprašanj glede zasebnosti, in posameznikom omogočajo večji nadzor nad njihovimi osebnimi podatki. To podpoglavje obravnava uporabo tehnologije ZKP v kontekstu decentralizirane identitete, s posebnim poudarkom na njeni integraciji in uporabi v naši lastni decentralizirani denarnici Masca. Za ta namen smo uporabili platformo ZKP, znano kot Privado ID [21]. Ta tehnologija, prej poznana pod imenom PolygonID [17], predstavlja nabor razvojnih orodij, ki omogočajo učinkovito implementacijo zaupanja vredne in zasebne izmenjave preverljivih poverilnic [7]. Privado ID temelji na napredni kriptografiji in tehnologiji verige blokov, kar zagotavlja visoko raven varnosti in zasebnosti pri upravljanju digitalne identitete v okviru naše rešitve. Pri tem smo se osredotočili na naslednje tri ključne komponente teh orodij:

- Osnovni protokol Iden3 [15] in njegova vloga pri upravljanju identitete.
- Postopek izmenjave sporočil, ki omogoča varno preverjanje identitete.
- Uporaba ZKP v tem procesu za večjo varnost in zasebnost.

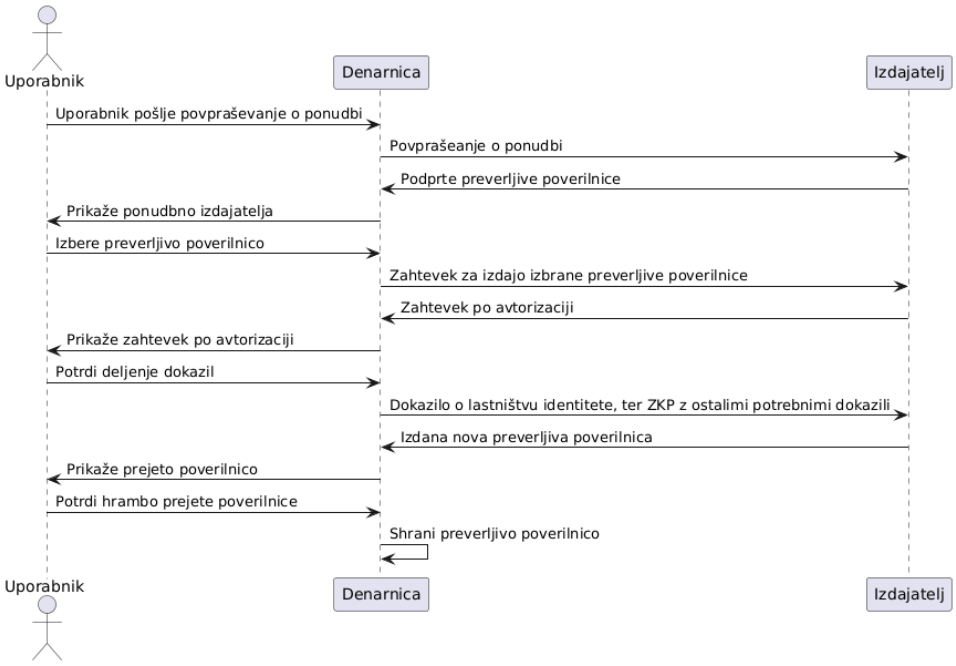
Glavna naloga Iden3 protokola je skrb za upravljanje uporabnikove identitete. Uporabnik lahko ima eno ali več identitet, ki so na omrežju Ethereum predstavljene kot računi (ang. accounts) ali pametne pogodbe (ang. smart contracts). Identitete lahko tako ustvarjajo kot tudi prejemajo izjave o identitetah, npr. izjava o starosti, državljanstvu ali izobrazbi. Te izjave služijo kot povezave med akterji in so lahko javne ali zasebne. Uporabniki lahko tudi kreirajo dokazila o prejetih izjavah. Ta dokazila se med drugim pri Iden3 protokolu ustvarjajo s pomočjo tehnologije ZKP, kjer se specifično uporablja sistem zk-SNARK. Sporočila in izjave v protokolu so tudi digitalno podpisana, kar omogoča preverjanje izvora, torej izdajatelja, in zagotovi, da ni prišlo do sprememb podatkov v tranzitu. Zraven omenjenih akterjev in konceptov so v Iden3 protokolu ključnega pomena tudi merklova drevesa (ang. Merkle trees). Ta zagotavljajo skalabilnost protokola tako, da zmanjšujejo število potrebnih transakcij za posodabljanje stanja na verigi blokov in omejujejo količino podatkov, ki jih je treba hraniti na verigi. Uporabljajo se za dvoje: prvič, za vodenje stanja identitete, kar vključuje evidenco prejetih in izdanih izjav, in drugič, za spremljanje veljavnosti izdanih izjav – torej ali so te izjave še vedno veljavne ali pa so bile preklicane. Pri tem je pomembno poudariti, da v primeru digitalno podpisanih izjav ni potrebno posodabljati stanja na verigi blokov. Digitalni podpis sam po sebi namreč zagotavlja dovolj visoko raven verodostojnosti in avtentičnosti izdane izjave.

Platforma Privado ID zraven protokola Iden3 uporabljajo tudi protokol Iden3comm [16]. Ta definira pravila za komunikacijo med agenti ter denarnicami, in je grajen na protokolu za sporočanje DIDComm [6]. Izjave v

protokolu so predstavljene v obliki W3C preverljivih poverilnic (ang. verifiable credentials) [24]. Pravila definirajo obliko sporočil za izdajanje preverljivih poverilnic, avtorizacijo, izmenjavo dokazov in povpraševanja o preklicu izdanih poverilnic.

Življenjski cikel izdajanja preverljive poverilnice na aplikacijskem nivoju, pri uporabi Privado ID orodij je prikazan na Sliki 1 in je sledeč:

- Na začetku uporabnik preko svoje digitalne denarnice pošlje povpraševanje o ponudbi izdajatelju. S tem si pridobi informacije o poverilnicah, ki jih le ta lahko izdaja.
- Nato uporabnik izbere določeno preverljivo poverilnico in preko denarnice pošlje zahtevek za izdajo le te.
- Strežnik oz. agent se odzove z zahtevkom za avtorizacijo, kjer mora uporabnik dokazati lastništvo identitete s katero želi prevzeti poverilnico. Ta zahtevek lahko vsebuje tudi druge pogoje, ki jih uporabnik mora dokazati, npr. da je polnoletna oseba, da ima opravljeno določeno stopnjo izobrazbe ipd.
- Denarnica uporabniku prikaže avtorizacijski zahtevek, na primer za dokazovanje polnoletnosti, in uporabnik odobri deljenje dokazil.
- Uporabnik se nato odzove s potrebnimi dokazili, pri čemer lastništvo identitete dokaže z digitalnim podpisom, za ostala dokazila pa uporabi ZKP. Za ustvarjanje podpisov in dokazil je odgovorna uporabnikova denarnica. V primeru dokazovanja polnoletnosti denarnica ustvari ZKP iz osebnega dokumenta v obliki preverljive poverilnice. Ta vsebuje le dokaz, da je oseba starejša od 18 let, pri čemer ne razkrije nobenih dodatnih osebnih informacij, niti točne starosti.
- Strežnik nato preveri prejeta dokazila in v primeru, da so bili vsi pogoji izpolnjeni, izda uporabniku preverljivo poverilnico. V nasprotnem primeru se odzove z napako.
- Denarnica prikaže prejeto poverilnico in jo po potrditvi uporabnika shrani.



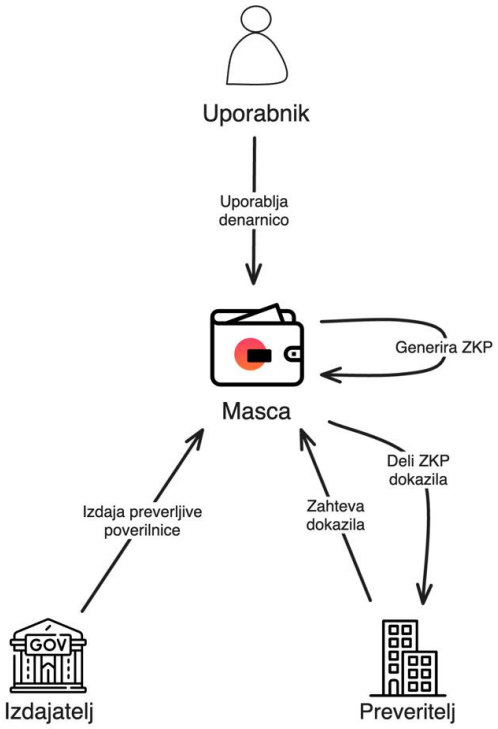
Slika 1: Življenjski cikel izdajanja preverljive poverilnice.

Po pregledu življenjskega cikla izdajanja preverljivih poverilnic z orodji Privado ID in uporabljenih osnovnih protokolov bomo predstavili še integracijo te tehnologije v lastno razvito digitalno denarnico Masca. Masca je

MetaMask Snap [20] (vtičnik), ki razširi delovanje kripto denarnice MetaMask s funkcionalnostmi samo-upravljane identitete (ang. self-sovereign identity, SSI) oz. decentralizirane identitete [19]. Obstoječim funkcionalnostim smo s pomočjo Privado ID knjižnic dodali podporo za protokola Iden3 in Iden3comm, ter podporo za ustvarjanje ZKP, ki temeljijo na sistemu zk-SNARK. Podpora za procesiranje sporočil Iden3comm protokola je bila dodana tako, da smo razširili naš seznam podprtih QR kod s pravili definiranih v protokolu. S tem smo omogočili uporabnikom denarnice Masca, da uspešno prevzamejo zahtevke za izdajo preverljivih poverilnic, kot tudi zahtevke po avtorizaciji. Procesiranje prevzetih zahtevkov smo opravili s pomočjo integriranih knjižnic, pri čemer smo v sam proces dodali tudi varen in uporabniško prijazen vmesnik. Za ustvarjanje ZKP s sistemom zk-SNARK je bilo potrebno v denarnico Masca uvoziti potrebna vezja. Ta vezja so zaradi svoje splošne opredelitve pogosto precej obsežna, njihova velikost pa lahko presega tudi 200 MB. Služijo kot specializiran programski jezik in nekakšen poizvedovalni jezik, ki določa obseg dokazljivih trditev znotraj določenega sistema ZKP. Obenem zagotavljajo prilagodljiv okvir za izražanje in poizvedovanje kompleksnih trditev v obliki, primerni za kriptografsko preverjanje. Zaradi svoje velikosti ta vezja povzročajo precejšnjo upočasnitev delovanja sistema in podaljšajo čas, potreben za prvo namestitev denarnice. Poleg tega je treba vezja ob vsaki inicializaciji denarnice znova naložiti v pomnilnik, kar dodatno vpliva na hitrost in odzivnost sistema. Za namen izboljšanja delovanje smo implementirali sledeče optimizacije:

- Spremenili smo način, kako so vezja kodirana, s čimer smo zmanjšali začetni čas nalaganja.
- Vezja smo shranili v predpomnilnik (ang. cache), da se izognemo ponovnemu nalaganju v primeru procesiranja zaporednih zahtevkov.
- Vezja se naložijo po potrebi, kar pomeni, da ne vplivajo na delovanje ostalih funkcionalnosti denarnice, le na ustvarjanje ZKP.

Slika 2 predstavlja celoten model zaupanja SSI. Prikazuje uporabnika, ki upravlja svojo digitalno identiteto s pomočjo denarnice, v tem primeru je to Masca. Uporabnik lahko v to denarnico prejema preverljive poverilnice od različnih izdajateljev, kot so državne institucije ali univerze. Prav tako lahko prejema zahtevke po avtorizaciji, na primer ko podjetje želi dokaz o uporabnikovi izobrazbi. Denarnica Masca je odgovorna za obdelavo teh zahtevkov, pripravo ZKP dokazil ter njihovo deljenje s preveritelji.



Slika 2: Model zaupanja SSI.

3.2. Strojno učenje

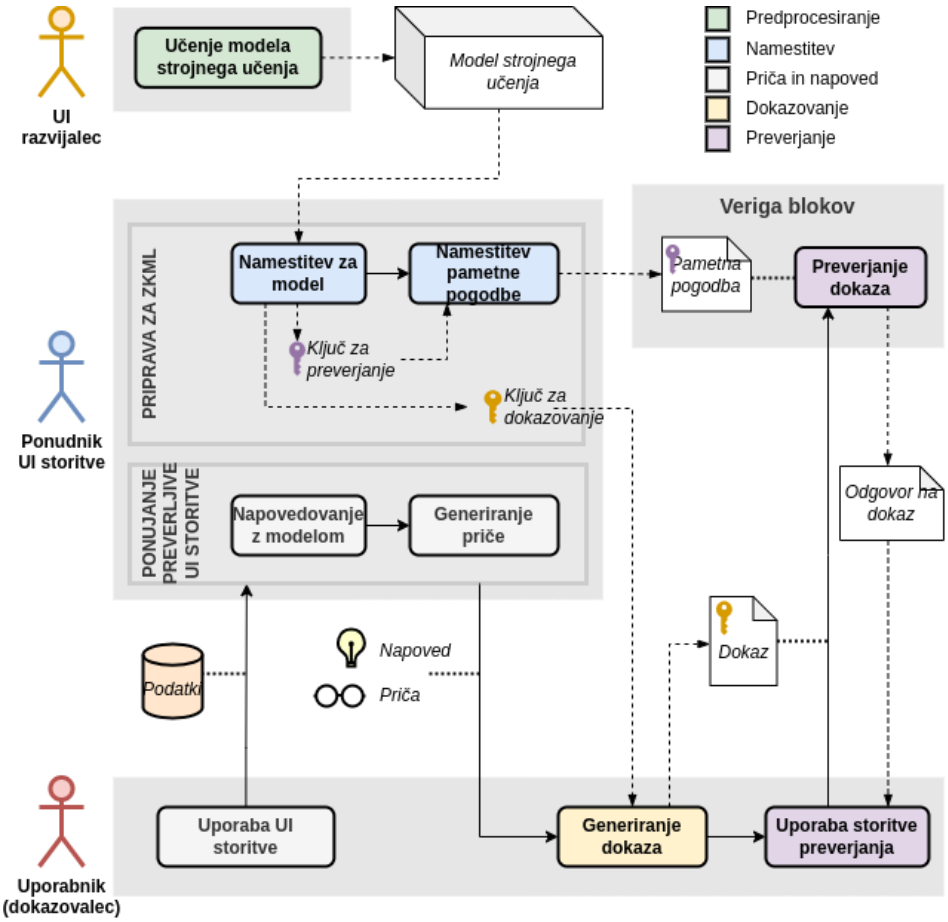
ZKP so se na začetku uporabljali za dokazovanje preprostih trditev in manjše količine podatkov, kot je bilo predstavljenih v prejšnjih poglavjih. Toda zaradi svoje generičnosti se lahko prav tako uporabijo za dokazovanje splošnih računalniških programov napisanih v poljubnem Turingovo polnem jeziku. Eden izmed takšnih primerov uporabe je uporaba ZKP na modelih strojnega učenja, tj. za dokazovanje napovedovanja modelov [12]. To področje se imenuje zero-knowledge machine learning (ZKML) ali pogosto v literaturi preverljivo strojno učenje (ang. verifiable machine learning) [23, 18]. Ker modeli stojnega učenja zelo računsko zahtevni programi, je bila uporaba ZKP na tem področju zelo dolgo praktično nemogoča, vendar je v zadnjih letih s teoretičnimi napredki in razvojem ogrodij postala izvedljiva, vendar le za napovedovanje in ne celotno učenje.

Dodatna vrednost, ki jo prinese ZKML, je preverljivost napovedovanja modelov strojnega učenja. Zaupanje v poštenost operaterjev in ponudnikov API storitev oz. modelov strojnega učenja se nadomesti z matematično/kriptografsko preverljivostjo. Z uporabo ZKML je mogoče odgovoriti na naslednja vprašanja oz. dokazati sledeče trditve:

- Uporabljen je bil točno določen model strojnega učenja M.
- Napoved y je bila pri uporabi modela M dobljena pri uporabi vhodnih podatkov x.
- Pri določenem vhodu x (ni razkrit preveritelju) smo dobili izhod y (razkrit preveritelju).

Medtem ko ZKML sam po sebi definira le način generiranja in preverjanja dokazov, sta se razvila dva prevladujoča načina uporaba: na verigi blokov (ang. on-chain) in izven (ang. off-chain). Pri slednjem potekata generiranje in preverjanje dokazov izven verige blokov tj., na strežniku, medtem ko pri prvem načinu poteka preverjanje dokaza na verigi blokov tj., s pomočjo pametnih pogodb. ZKML se lahko tako uporabi znotraj obstoječih informacijskih sistemov, ki ponujajo storitve strojnega učenja kot tudi pri decentraliziranih aplikacijah (ang. dapp).

Na sliki 3 je prikazan postopek ZKML na verigi blokov, v katerem smo definirali štiri tipe akterjev - razvijalec umetne inteligence (UI), ponudnik UI storitve, uporabnik, ki je v vlogi dokazovalca, in veriga blokov s pametno pogodbo, ki je v vlogi preveritelja. UI razvijalec poskrbi za definicijo in učenje modela strojnega učenja, kar spada pod pred-procesiranje in ni neposredno del ZKML. Ponudnik UI storitve izvede namestitev za izbran model, kjer se generirata ključa za dokazovanje in preverjanje, pri čemer se slednji namesti kot del pametne pogodbe na verigo blokov. Uporabnik nato uporablja UI storitve, pri čemer prejme poleg same napovedi tudi podatke potrebne za generiranje dokazov t. i. pričo (ang. witness). Po generaciji dokaza lahko uporabnik sam dokaz uporabi v decentraliziranih aplikacijah, kjer verifikacija dokaza poteka s pametnimi pogodbami.



Slika 3: Postopek uporabe ZKML na verigi blokov.

Trenutno najbolj razviti ogrodji za ZKML sta ezkl in Orion. Prvo ogrodje omogoča generiranje zk-SNARK dokazov, medtem ko drugo zk-STARK. ezkl temelji na ZKP sistemu Halo2, pri čemer omogoča pretvorbo modelov strojnega učenja v razširjenem formatu ONNX (Open Neural Network Exchange) v Halo2 vezja. Ogrodje Orion medtem vsebuj orodja in knjižnice za implementacijo modelov strojnega učenja v programskem jeziku Cairo, tj. programski jezik za programiranje preverljivih programov in pametnih pogodb za L2 omrežju Starknet. Ogrodji se razlikujeta tudi po različnih načinih zasebnosti, saj Orion ponuja le javen način, pri katerem so razkrite vse vrednosti (vhodi, izhod in uteži), medtem ko ezkl omogoča tudi zasebne načine, pri čemer se razkrijejo le določeni podatki. Primerjava med ezkl in Orion je povzeta v tabeli 2.

Tabela 2: Primerjava ezkl in Orion.

Lastnost	ezkl	Orion
ZKP sistem	zk-SNARK	zk-STARK
Zaupanja vredna namestitev	Da	Ne
ZKP ogrodje	Halo2	Cair – Stone, Platinum ...
Podprti modeli	Nevronske mreže, odločitvena drevesa, naključni gozdovi ...	Nevronske mreže, odločitvena drevesa, naključni gozdovi, SVM ...
Programski jeziki	Rust, Python, JavaScript	Cairo, Python
Načini zasebnosti	Zasebno, javno, šifrirano	Javno

ZKML lahko na več načinov poveča funkcionalnosti decentraliziranih aplikacij, pri čemer se kompleksnejše računanje (napovedovanje modela) izvede izven verige blokov, medtem ko se preverjanje dokaza in uporaba izhoda zgodi na verigi blokov v pametnih pogodbah. Eden izmed takšnih primerov uporabe je diverzifikacija portfelja

kriptožetonov decentraliziranih avtonomnih organizacij (ang. decentralized autonomous organization, DAO). V finančni industriji se modeli strojnega učenja pogosto uporabljajo v te namene, medtem ko se v portfeljih na verigi blokov ne, saj so modeli prekompleksni, da bi se neposredno izvedli v pametnih pogodbah. Če pa uporabimo samo napoved kot parameter funkcije, ne vemo, kako je bila napoved o prerazporejanju sredstev pridobljena - torej, če je bil uporabljen res točno izbran model strojnega učenja. ZKML omogoča, da se model napovedovanja izvede izven verige blokov, bodisi na lokalnem računalniku ali na strežniku, pri čemer se dodatno generira ZKP dokaz o napovedovanju/inferenci modela. Ko se nato kliče funkcija na pametni pogodbi, ki izvede prerazporejanje kriptožetonov od DAO-ta, sprejme kot vhod ZKP dokaz in izvede preverjanje dokaza, pri čemer se prerazporejanje zgodi le, če je bila dobljena napoved/strategija za prerazporejanje res izhod tistega modela strojnega učenja.

4 Diskusija

ZKP se danes aktivno in uspešno uporabljajo za različne primere, pri čemer smo se v članku osredotočili na primer decentralizirane identitete in strojnega učenja. Pri prvem omogočajo varnejšo in bolj zasebno razkritje osebnih podatkov, kjer je mogoče le razkrite nujno potrebne vrednosti oz. v določenih primerih le predikate, na primer da je določena vrednost večja od določene številke. Pri strojnem učenju dodajo zmožnost preverjanje, kar omogoči verifikacijo izvedbe modela za napovedovanje. Na ta način lahko uporabnik preveri, da je bil uporabljen izbran model ob določenem vhodu, pri čemer se zaupanje v ponudnika storitve nadomesti z matematičnimi zakoni.

Kljub vsem prednostim imajo ZKP še danes določene pomanjkljivosti. Ena izmed pomanjkljivosti so kompleksnejši sistemi, saj je za generiranje ZKP dokazov potrebno imeti dostop do ZKP vezij, ki so pogosto velika več sto MB, kar je lahko za mobilne naprave z nekoliko slabšimi procesorji zahtevno. Samo generiranje ZKP dokazov je tudi zahteven proces, ki lahko v določenih primerih traja več sekund oz. v primeru strojnega učenja več minut/ur.

5 Zaključek

ZKP so ena izmed tehnologij, ki je v zadnjih letih prešla iz teorije v praktično uporabo. Zaradi svojih značilnosti lahko obstoječe in nove aplikacije, na primer decentralizirane, nadgradi z novimi funkcionalnostmi, kot so večji nadzor nad zasebnimi podatki, povečana zasebnost in varnost. V članku smo predstavili njeno uporabnost na primeru decentralizirane identitete in strojnega učenja.

Čeprav se ZKP lahko že danes uporabljajo v praksi, je odprtih še veliko izzivov, ki jih je potrebno nasloviti. Eden izmed takšnih izzivov je skalabilnost tehnologije na primere z večjo količino podatkov, kar se še posebej pozna pri strojnem učenju, kjer modeli oz. nevronske mreže v praksi vsebujejo več milijonov uteži/nevronov. Raziskovalci se prav tako ukvarjamo z razvojem novih ZKP protokolov, ki naslavljajo odprte probleme trenutnih zk-SNARK in zk-STARK sistemov, kot so zaupanja vredna namestitev, velikost dokazov in odpornost na kvantne računalnike.

Literatura

[1] Ben Adida. "Helios: web-based open-audit voting". In: Proceedings of the 17th Conference on Security Symposium. SS'08. San Jose, CA: USENIX Association, 2008, pp. 335–348.

[2] Eli Ben Sasson et al. "Zerocash: Decentralized Anonymous Payments from Bitcoin". In: 2014 IEEE Symposium on Security and Privacy. 2014, pp. 459–474. doi: 10.1109/SP.2014.36.

[3] Eli Ben-Sasson et al. "Scalable, transparent, and post-quantum secure computational integrity". In: Cryptology ePrint Archive (2018).

[4] Benedikt Bünz et al. "Bulletproofs: Short Proofs for Confidential Trans- actions and More". In: 2018 IEEE Symposium on Security and Privacy (SP). 2018, pp. 315–334. doi: 10.1109/SP.2018.00020.

[5] Zhigang Chen et al. "A Survey on Zero-Knowledge Authentication for Internet of Things". In: Electronics 12.5 (2023). issn: 2079-9292. doi: 10.3390/electronics12051145. url: https://www.mdpi.com/2079- 9292/12/5/1145.

[6] DIDComm messaging protocol. url: https://identity.foundation/didcomm-messaging/spec/v2.1/ (obiskano 16.7.2024).

[7] Martin Domajnko, Vid Keršič, and Muhamed Turkanović. "OID4VC: izdajanje in deljenje preverljivih poverilnic na osnovi OpenID Connect". In: Nasl. z nasl. zaslona. Univerza v Mariboru, Univerzitetna založba; Fakulteta za elektrotehniko, računalništvo in informatiko, 2023, pp. 149–163. url: https://press.um.si/index.php/ump/catalog/book/804.

[8] Ethereum developer documentation: Zero-knowledge rollups (ZK-rollups). url: https://ethereum.org/en/developers/docs/scaling/zk- rollups/ (obiskano 15.7.2024).

[9] EU Digital Identity Wallet Home. url: https://ec.europa.eu/digital-building-blocks/sites/display/EUDIGITALIDENTITYWALLET/EU+Digital+Identity+Wallet+Home (obiskano 19.7.2024).

[10] Uriel Feige, Amos Fiat, and Adi Shamir. "Zero-knowledge proofs of identity". In: Journal of Cryptology 1.2 (June 1988), pp. 77–94. issn: 1432- 1378. doi: 10.1007/bf02351717. url: http://dx.doi.org/10.1007/ BF02351717.

[11] Amos Fiat and Adi Shamir. "How to prove yourself: Practical solutions to identification and signature problems". In: Conference on the theory and application of cryptographic techniques. Springer. 1986, pp. 186–194.

[12] Rosario Gennaro, Craig Gentry, and Bryan Parno. "Non-interactive verifiable computing: Outsourcing computation to untrusted workers". In: Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30. Springer. 2010, pp. 465–482.

[13] Rosario Gennaro et al. "Quadratic span programs and succinct NIZKs without PCPs". In: Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings 32. Springer. 2013, pp. 626–645.

[14] S Goldwasser, S Micali, and C Rackoff. "The knowledge complexity of interactive proof-systems". In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. STOC '85. Providence, Rhode Island, USA: Association for Computing Machinery, 1985, pp. 291–304. isbn: 0897911512. doi: 10.1145/22145.22178. url: https://doi. org/10.1145/22145.22178.

[15] Iden3 protocol documentation. url: https://docs.iden3.io/ (obiskano 16.7.2024).

[16] Iden3comm protocol documentation. url: https://iden3-communication. io/ (obiskano 16.7.2024).

[17] Introducing Privado ID. url: https://www.privado.id/blog/introducing-privado-id-moving-beyond-polygon-to-deliver-independent-privacy-preserving-identity-solutions (obiskano 16.7.2024).

[18] Vid Kersic and Muhamed Turkanovic. "A review on building blocks of decentralized artificial intelligence". In: arXiv preprint arXiv:2402.02885 (2024).

[19] Vid Keršič et al. "Dodajanje poljubnih funkcionalnosti digitalni kripto denarnici MetaMask". In: Nasl. z nasl. strani. Univerza v Mariboru, Univerzitetna založba; Fakulteta za elektrotehniko, računalništvo in informatiko, 2022, pp. 141–154. url: https://dk.um.si/IzpisGradiva. php?id=82880.

[20] MetaMask Snaps. url: https://metamask.io/snaps/.

[21] Privado ID. url: https://www.privado.id/ (obiskano 16.7.2024).

[22] Srinath Setty. "Spartan: Efficient and general-purpose zkSNARKs without trusted setup". In: Annual International Cryptology Conference. Springer. 2020, pp. 704–737.

[23] Tobin South et al. "Verifiable evaluations of machine learning models using zkSNARKs". In: arXiv preprint arXiv:2402.02675 (2024).

[24] Verifiable Credentials Data Model v2.0. url: https://www.w3.org/TR/vc-data-model-2.0/ (obiskano 16.7.2024).