ŠOLSKI CENTER CELJE

Srednja šola za kemijo, elektrotehniko in računalništvo

USTVARJANJE IN UPORABA LASTNE VERIGE BLOKOV ZA DELJENJE NECENZURIRANIH NOVIC

raziskovalna naloga

Avtorji:

Maj Zabukovnik, R-4. b

Aney Vaishnav, R-4. b

Luka Bombek, R-4. b

Mentor:

mag. Boštjan Resinovič

Mestna občina Celje, Mladi za Celje

Celje, marec 2024

ŠOLSKI CENTER CELJE

Srednja šola za kemijo, elektrotehniko in računalništvo

USTVARJANJE IN UPORABA LASTNE VERIGE BLOKOV ZA DELJENJE NECENZURIRANIH NOVIC

raziskovalna naloga

Avtorji:

Maj Zabukovnik, R-4. b

Aney Vaishnav, R-4. b

Luka Bombek, R-4. b

Mentor:

mag. Boštjan Resinovič

Mestna občina Celje, Mladi za Celje

Celje, marec 2024

IZJAVA*

Mentor Boštjan Resinovič v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Ustvarjanje in uporaba lastne verige blokov za deljenje necenzuriranih novic, katere avtorji so Maj Zabukovnik, Aney Vaishnav in Luka Bombek:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 27. 3. 2024

žig šole

ŠOLSKI CENTER CELJE
SREDNJA ŠOLA ZA KEMIJO, ELEKTROTEHNIKO IN RAČUNALNIŠTVO
CELJE, Pot na Lavo 22
1

Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

Zahvala

Zahvaljujemo se vsem, ki so kakorkoli pomagali pri izdelavi raziskovalne naloge. Brez vaše pomoči raziskovalna naloga ne bi nastala, pa naj je šlo le za spodbudne besede, majhne ideje ali nasvete in predvsem kritike pri izdelavi, saj smo se iz njih naučili največ ter je tako nastal še boljši končni izdelek.

Najprej bi se iskreno zahvalili profesorju mag. Boštjanu Resinoviču za motivacijo, da smo se lotili izdelave raziskovalne naloge in za vse strokovne nasvete glede izdelave. Za idejno zasnovo in oblikovanje uporabniškega vmesnika pa bi se radi zahvalili tudi sošolcu Marku Sadniku.

Zahvalili bi se tudi profesorici Tjaši Verdev, ki je poskrbela za lektoriranje raziskovalne naloge in svetovanje pri oblikovanju besedila ter profesorici Rosani Breznik, ki je prevedla in lektorirala angleški povzetek.

Povzetek

Uporaba tehnologije veriženja blokov je v razmahu že vse od njene prve uporabe za izdelavo kriptovalute imenovane bitcoin in je danes uporabljena na raznolikih področjih, denimo v zdravstvu, finančnem sektorju, energetiki ter logistiki.

Cilj raziskovalne naloge je bil podrobno raziskati delovanje tehnologije veriženja blokov in s pomočjo pridobljenega znanja izdelati lastno implementacijo verige blokov. Ker trenutno na svetu obstaja več kot 10.000 različnih kriptovalut, smo se posvetili niši, ki poleg monetarnih transakcij omogočajo še objavo novic. S tem smo zagotovili popolno integriteto in preprečili kakršno koli obliko cenzure vsebine. Poskrbeli smo tudi za uporabo čim močnejšega kriptografskega algoritma in uporabo sistema nagrajevanja, ki bosta v prihodnosti skrbela za samozadostnost ter stabilnost omrežja. Tekom izdelave raziskovalne naloge smo izdelali tudi priročno spletno aplikacijo, ki uporabniku omogoča tako ogled že objavljenih novic kot tudi njihovo objavo.

Ključne besede: *kriptografija, kriptovalute, necenzurirane novice, omrežje enakovrednih partnerjev, tehnologija veriženja blokov.*

Abstract

The use of blockchain technology has been on the rise ever since its initial application in creating the cryptocurrency known as Bitcoin and is now applied in diverse fields such as healthcare, finance, energy, and logistics.

The aim of the research project was to thoroughly investigate the functioning of blockchain technology and, with the acquired knowledge, create our own implementation of a blockchain. Given that there are currently more than 10,000 different cryptocurrencies worldwide, we focused on a niche that allows not only monetary transactions but also the publication of news. This ensured complete integrity and prevented any form of content censorship. We also ensured the use of robust cryptographic algorithms and implemented a reward system to ensure the future self-sufficiency and stability of the network. Throughout the research project, we have developed a user-friendly web application that enables users to view already published news and contribute their own publications.

Keywords: *cryptography, cryptocurrencies, uncensored news, peer-to-peer network, blockchain technology.*

Kazalo vsebine

Kazalo slik

Kazalo tabel

1 UVOD

1.1 Opredelitev področja in raziskovalnega problema

Leta 2008 se je svet moral spopasti z do tedaj nepredstavljivim propadom nekaterih izmed največjih bank v ZDA in reševanje le-teh s strani ameriške vlade. Ta dogodek je med navadnimi posamezniki povzročil veliko nezaupanje v sistem bančništva z delnimi rezervami, kot tudi razkril težnjo vladajočih po ohranitvi status quo. Eden izmed nasprotnikov te ekonomske politike, pod psevdonimom Satoshi Nakamoto, je kot odziv leta 2009 javno objavil prvo širše prepoznavno tehnologijo veriženja blokov imenovano bitcoin.

Ta je prvič v zgodovini omogočila izvajanje varnih in nespremenljivih monetarnih transakcij brez nadzora s strani centralne entitete. S tem se je Satoshi prav tako popolnoma znebil potrebe po zaposlovanju mediatorjev, ki pri centraliziranih plačilnih sistemih, denimo PayPal, predstavljajo dodatne stroške, katere posredno pokrijejo uporabniki s plačevanjem bančnih provizij. Sprva so se s to tehnologijo spoznali le kriptografi in računalniški navdušenci, vendar je z naraščanjem tržne kapitalizacije kriptovalute zanimanje prešlo tudi med laike. Medtem ko je večina iskala razne načine pridobitve hitrega bogastva s pomočjo izkoriščanja kratkoročne volatilnosti na trgu, so izjeme med njimi prepoznale širšo vrednost tehnologije in pričele z objavo lastnih ambicioznih decentraliziranih projektov, ki so omogočili vse od pametnih pogodb do decentraliziranih alternativ borzam.

Nenazadnje je manjša skupina razvijalcev prišla do spoznanja, da je prihod interneta informacije močno centraliziral in s tem vladajočim omogočil lažjo cenzuro nad znanjem. V zadnjih letih je ravno iz tega razloga vedno težje najti nepristranske, verodostojne, kvalitetne vire, ki (še) niso podlegli korporativnim ali vladnim pritiskom. S tem namenom smo se odločili izdelati raziskovalno nalogo, s katero želimo narediti lastno implementacijo verige, ki poleg običajnih monetarnih transakcij omogoča še hranjenje tekstovnih novic direktno v verigi blokov, s čimer vsaka objava postane imuna kakršnimkoli modifikacijam, zanikanju avtorstva ter izbrisom. Raziskovalna naloga prav tako predstavlja kontinuiteto lanskoletne raziskovalne naloge [1], v kateri smo izdelali lastno knjižnico za kriptografijo z eliptičnimi krivuljami, ki nam bo letos v pomoč pri implementaciji kredibilnega digitalnega podpisovanja transakcij.

1.2 Cilji in hipoteze

Cilj raziskovalne naloge je izdelava lastne verige blokov, ki hkrati poleg monetarnih transakcij omogoča še transakcije s poljubnim besedilom oziroma novico. Ker je vsaka transakcija v verigi blokov shranjena trajno, bo takšna novica v povsem nespremenjeni obliki objavljena večno in posledično odporna pred kakršnokoli obliko cenzure.

V raziskovalni nalogi smo postavili naslednje hipoteze:

1. Naša implementacija verige blokov je hkrati primerna za objavo necenzuriranih novic in varen prenos monetarne vrednosti, kar ji prinaša prednost v primerjavi z obstoječimi rešitvami.
2. Izdelana veriga blokov je ustrezno kriptografsko zavarovana in varna pred večino napadov.
3. Nagrajevanje rudarjev in omejitev velikosti blokov omogočata obstoj omrežja tudi v realnem svetu, saj bo vsaj del operativnih stroškov rudarjev pokrit.

1.3 Metode raziskovanja

Med snovanjem idejne osnove raziskovalne naloge smo se osredotočili predvsem na podrobno preučevanje literature in temeljito analizo konkurenčnih sistemov na osnovi tehnologije veriženja blokov. Odkrili smo tržno vrzel, saj navkljub obstoju številnih tehnologij za decentralizirano hranjene podatkov ne obstaja tehnologija prilagojena specifično objavi publicističnih besedil.

Po izoblikovani ideji smo ocenili zahtevnost in izvedljivost projekta. Zaradi obsežnosti tematike, ki jo obravnavamo v raziskovalni nalogi, smo tudi med samim pisanjem morali poseči po strokovni literaturi in podrobneje proučevati strukture kriptovalut, ki delujejo na osnovi dokaza o delu.

Ob koncu raziskovalnega dela je bilo končni izdelek potrebno temeljito preizkusiti, zato smo izvedli tudi številne teste, ki so simulirali delovanje v resničnem svetu. Izdelek smo ponudili tudi izbrani skupini, ki je sodelovala v omrežju in poganjala lastna vozlišča, kar nam je omogočilo dodatno testiranje zanesljivosti delovanja omrežja.

1.4 Struktura raziskovalne naloge

Raziskovalna naloga v grobem sestoji iz dveh delov, torej predstavitve teoretične osnove in praktičen prikaz uporabe pridobljenega teoretičnega znanja. Teoretično osnovo zajemata predvsem prvo poglavje, ki je uvod v delo in drugo poglavje, ki na poljuden način predstavi zasnovo ter delovanje tehnologije veriženja blokov. Tretje poglavje je bolj strokovno usmerjeno, saj je v njem predstavljena naša implementacija te tehnologije. V četrtem poglavju je predstavljeno omrežje enakovrednih partnerjev (*ang. P2P network*), sledita poglavji, ki predstavita uporabljene programske jezike in orodja ter izdelano uporabniško aplikacijo za interakcijo z verigo blokov. V sedmem poglavju so predstavljeni rezultati raziskovalne naloge in analiza hipotez postavljenih ob začetku raziskovanja. Raziskovalna naloga se zaključi z osmim poglavjem, ki vsebuje zaključek, v devetem pa so navedeni uporabljeni viri in literatura.

2 TEHNOLOGIJA VERIŽENJA BLOKOV

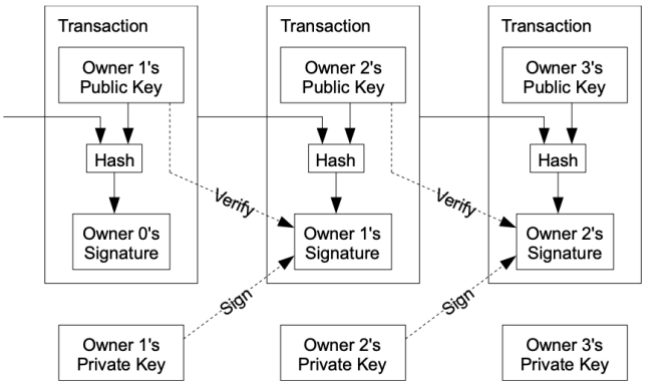
Tehnologija veriženja blokov (*ang. blockchain*) predstavlja decentralizirano bazo podatkov oziroma porazdeljeno javno knjigo, ki s pomočjo kriptografije omogoča trajno in zanesljivo hranjenje podatkov. Njena uporaba se je sprva razširila med raznimi kriptovalutami, čeprav jo dandanes podjetja med drugim uporabljajo tudi za zagotavljanje integritete pri dobavni verigi, izvedbi pametnih pogodb, implementaciji povsem digitalnih volitev in dokazovanju lastništva avtorskega dela. Ravno zaradi vsestranskosti te tehnologije je pomembo razumevanje njenega delovanja. V nadaljnjih poglavjih so podrobneje predstavljeni osnovni koncepti tehnologije veriženja blokov. Čeprav so ti večino zaradi preprostosti predstavljeni z vidika monetarnih prenosov vrednosti, se je pomembno zavedati, da v splošnem transakcija predstavlja le podatkovno strukturo za zapis nekih informacij, medtem ko podatkovni blok trajno v verigo zapiše do tedaj še nepotrjene transakcije. [2]

2.1 Transakcije[1]

Transakcija je po slovarski definiciji poslovno dejanje, pri katerem dobi vsaka stran določeno vrednost. V fizičnem svetu so transakcije trivialne, sploh pri poslovanju z gotovino, saj denimo kupec za določene dobrine prodajalcu na licu mesta izroči ustrezno monetarno vrednost. Trgovcu je tako omogočena takojšna overitev veljavnosti izročene gotovine, kupec pa istega bankovca ne bo mogel ponovno porabiti pri drugem trgovcu. Z razvojem interneta so se pojavile spletne trgovine, ki so za poslovanje morale implementirati alternativne plačilne sisteme primerljive z gotovino, ki bi bili ne le preprosti za uporabo, ampak tudi zanesljivi. Problematiko so razrešila velika plačilna podjetja, denimo Visa, MasterCard, PayPal, Stripe in drugi, ki so prek centralnih strežnikov sprejemali zahtevke za plačilo, izvedli poizvedbo o plačilni sposobnosti potrošnika in se na podlagi teh informacij odločili za odobritev oziroma zavrnitev transakcije. S prevzemom vloge posrednika med potrošniki in prodajalci se je zaupanje o zanesljivosti plačil preneslo na njih, posledično pa so za svojo storitev zaračunali določeno provizijo (do 3 % zneska transakcije). Dandanes vse spletne in kartične transakcije slonijo na kredibilnosti plačilnih podjetji, ki morajo skrbeti za verodostojnost transakcij ter posledično preprečiti dvojno porabo sredstev.

[1] Poglavje je povzeto po [17].

Pri implementaciji tehnologije veriženja blokov so ravno zanesljive in verodostojne transakcije predstavljale največji izziv, saj brez centralne entitete na videz ni bilo mogoče preprečiti zapravljanja neobstoječih sredstev. Problem je bil v celoti razrešen z iznajdbo tehnologije bitcoina, ki je zajela del inovativnih rešitev njenih predhodnikov, predvsem ecash, b-money, bitgold in hashcash. Satoshijeva rešitev je zajemala verigo digitalnih podpisov, saj je vsak plačnik moral zgoščeno vrednost[2] prejšnje transakcije in javni ključ[3] naslednjega prejemnika podpisati s svojim zasebnim ključem. S tem lahko vsakdo potrdi veljavnost podpisa in verigo lastništva vse do izvorne transakcije, ko so bili kovanci ustvarjeni. Veriga podpisov je grafično prikazana na Sliki 1. Kljub sledljivosti izvora dvojno zapravljanje še vedno ni razrešeno, saj bi lahko denimo nekdo z zasebnim ključem ustvaril dva veljavna podpisa z različnima prejemnikoma. To bi mu omogočilo neomejeno porabo prvotno omejenih sredstev. Edina rešitev tega problema je v hranjenju celotne zgodovine transakcij v vsakem vozlišču omrežja. Tako lahko vozlišče ob prejemu transakcije preveri veljavnost digitalnega podpisa, kot tudi pregleda zgodovino vseh transakcij. Če je transakcija navkljub veljavnemu popisu v preteklosti že bila opravljena, vozlišče takšno transakcijo preprosto ovrže. Vsa poštena vozlišča v omrežju bodo storila enako, torej takšna transakcija ne bo sprejeta v omrežju kot veljavna.



Slika 1: Shema verige podpisov transakcij [3, p. 2]

[2] Zgoščeno vrednost dobimo z uporabo zgoščevalne funkcije, tj. vsaka funkcija, ki jo lahko uporabimo za enolično preslikavo podatkov poljubne velikosti v njihovo zgoščeno vrednost, katere velikost je konstanta. Zaradi enoličnosti preslikave vsaka sprememba v vhodu take funkcije drastično vpliva na končno vrednost. Bitcoin denimo uporablja algoritem SHA256, ki vrača zgoščene vrednosti velikosti 256 bitov. [10]

[3] Javni ključ prejemnika je redkokdaj naveden direktno in je najpogosteje podan posredno, kot naslov, ki je izpeljan iz javnega ključa. Bitcoinov naslov denarnice predstavlja zgoščena vrednost javnega ključa.

2.2 Blok transakcij[4]

V zgornjem poglavju smo definirali transakcijo in s pomočjo verige podpisov potrdili njeno veljavnost. Prav tako smo preprečili dvojno porabo sredstev s pomočjo pregleda zgodovine transakcij, vendar s tem pridemo do problema enoličnega hranjenja zaporedja transakcij, ki je v nadaljevanju predstavljen na konkretnem primeru.

Denimo, da imamo na voljo 10 enot neke kriptovalute in z njimi želimo pretentati sistem ter izvršiti dve transakciji po 10 enot. Sprva ustvarimo prvo transakcijo in jo posredujemo v omrežje, nato ustvarimo še drugo transakcijo in jo prav tako posredujemo v omrežje. Zaradi decentralizirane narave tehnologije veriženja blokov vsako vozlišče hrani lastno zgodovino vseh transakcij. Prav tako bo v decentraliziranem omrežju do nekaterih akterjev v omrežju najprej prišla naša prva transakcija, nato pa bodo ti drugo transakcijo zavrgli kot neveljavno, medtem pa bodo nekateri najprej prejeli našo drugo transakcijo ter posledično kasneje zavrnili prvo transakcijo. S tem pride do neskladnosti v omrežju glede zgodovine veljavnih transakcij. Takšne neskladnosti si ne moremo privoščiti, saj v omrežju ni glavnega vozlišča, ki bi delovalo kot mediator in razrešil dvoumnost.

S tem namenom vsako vozlišče posluša za transakcijami v omrežju in jih združuje v večjo entiteto imenovano podatkovni blok. Poleg informacij o vseh transakcijah blok sestoji tudi iz svoje zgoščene vrednosti, ki ga enolično določa, kot tudi vozlišču predstavlja kriptografski izziv. Rešitev le-tega vozlišču omogoči proizvesti veljaven blok. Kriptografski izziv je v bloku prisoten, saj moramo zagotoviti enotno zgodovino transakcij skozi vsa vozlišča v omrežju. Če bi lahko vsako vozlišče po svoje združevalo transakcije v poljubne bloke, bi prišli do podobnega problema, kot smo ga imeli pred uvedbo blokov transakcij. Zaradi prisotnosti kriptografskega izziva pa lahko v omrežju zagotovimo veliko večjo stabilnost, saj bodo vozlišča označila bloke kot veljavne, če bodo poleg veljavnih transakcij imeli še pravilno rešen kriptografski izziv. Ker rešitev tega izziva ni zagotovljena v razumnem času, bodo druga vozlišča z veseljem sprejela kopijo veljavnega bloka, ki ga je morebiti proizvedlo drugo vozlišče. Več o tem kriptografskem izzivu je predstavljeno v naslednjem poglavju.

[4] Poglavje je povzeto po [11].

2.3 Dokaz o delu[5]

Z uvedbo blokov transakcij je bilo potrebno uvesti nekakšen matematični izziv. Iskanje primerne funkcije vsekakor ni bila preprosta naloga, saj mora biti postopek rešitve dane funkcije zelo težaven, obenem pa mora biti potrjevanje pravilnosti le-te preprosto in hitro opravilo. Takšne funkcije so v literaturi poimenovane kot enosmerne funkcije. Nenazadnje je takšna funkcija moral omogočati tudi prilagoditev težavnosti reševanja.

Ker so računalniki že od samega začetka primarno namenjeni manipulaciji podatkov, lahko pozabimo na uvedbo kakršnekoli matematične operacije nad realnimi števili (seštevanje, odštevanje, množenje, deljenje, ...), saj so takšni izzivi za računalnik povsem trivialni. Hkrati odpade tudi vsaka funkcija, s katere bi se dalo razpoznati zaporedje. To bi bilo sprva morda težko razpoznavno, vendar bi se z napredkom hitrosti delovanja računalnikov kot tudi umetne inteligence takšno zaporedje sčasoma prepoznalo, posledično bi iz težko rešljive enosmerne funkcije prešli v preprosto dvosmerno funkcijo. Nenazadnje se je Satoshi Nakamoto odločil uporabiti zgoščevalno funkcijo, saj je njen izhod povsem nepredvidljiv glede na najmanjšo spremembo v vhodnih podatkih, preverjanje obstoječe rešitve je preprosto in omogoča prilagajanje težavnosti.

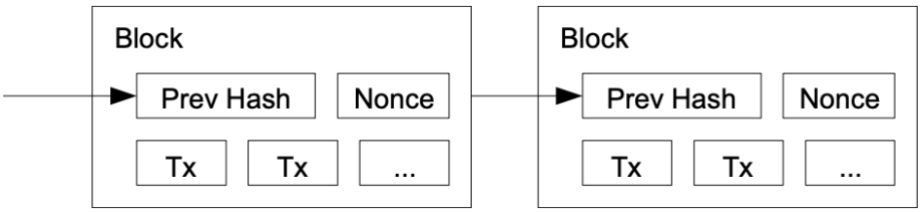
S tem si je Satoshi zamislil sistem dokaza o delu (*ang. proof-of-work*), ki je temeljil na uporabi zgoščevalne funkcije. Tako bi rudar[6] sprva vse podatke o bloku podal vhodu v zgoščevalno funkcijo, ta pa bi mu vrnila neko veliko število oziroma zgoščeno vrednost. Da bi otežil postopek ustvarjanja novih veljavnih blokov, je v izvorni kodi določil, da lahko vozlišče kot veljaven blok sprejme le tiste, katerih zgoščena vrednost na začetku vsebuje določeno število ničel. S tem je omogočil tudi spreminjanje težavnosti ustvarjanja novih blokov, saj se v primeru prehitrega generiranja novih, zahtevano število ničel na začetku zgoščene vrednosti poveča. To seveda velja tudi obratno. Ker pa zgoščevalna funkcija za iste podatke vedno vrne isto zgoščeno vrednost, je bloku bilo potrebno dodati še poseben atribut, ki ga vozliče lahko

[5] Poglavje je povzeto po [12].

[6] Rudar je vozlišče v omrežju, ki aktivno sodeluje pri procesu ustvarjanja novih blokov in potrjevanju transakcij. Način potrjevanje se razlikuje od omrežja do omrežja. Rudar je za svoje delo običajno nagrajen s kriptovaluto omrežja, če uspešno proizvede veljavni blok.

poljubno spreminja in s tem iz zgoščevalne funkcije pri istih podatkih o transakcijah dobi drastično različen izhod.

Navkljub uvedbi dokaza o delu še vedno nismo povsem razrešili problematike vrstnega reda blokov transakcij. V trenutni implementaciji je namreč mogoča poljubna razvrstitev veljavnih blokov. Da se povsem znebimo potencialnih dvoumnosti o ustreznem vrstnem redu, je med podatke trenutnega bloka potrebno dodati še zgoščeno vrednost prejšnjega bloka. S tem smo dosegli verigo blokov, saj se vsak blok poleg sebe sklicuje na svojega predhodnika, ki se sklicuje na svojega predhodnika, ki se sklicuje ... vse do prvega bloka v verigi, kar je razvidno s Slike 2. S tem nismo samo zagotovili enotnega vrstnega reda, vendar smo tudi preprečili kakršne koli spremembe v že proizvedenih blokih, saj bi sprememba v bloku z zaporedno številko *n* povsem spremenila njegovo zgoščeno vrednost, torej bloki od vključno *n + 1* ne bi bili več veljavni. Za uveljavitev takšne spremembe bi posameznik moral samostojno popraviti vse podatkovne bloke, katerih zaporedna številka je večja od zaporedne številke spremenjenega bloka. To bi s časoma postal skoraj nemogoč izziv, saj bi posameznik moral hitreje od omrežja popravljati neveljavne bloke kot tudi proizvajati nove.



Slika 2: Prikaz povezave blokov v verigo [3, p. 3]

2.4 Varnost in zanesljivost omrežja[7]

Uporaba tehnologije veriženja blokov za izvedbo transakcij, še posebej monetarnih, ne bi bila mogoča, če ta tehnologija ne bi bila zanesljiva in varna. Vsi do sedaj predstavljeni koncepti v drugem poglavju pripomorejo k varnosti in integriteti samih transakcij, vendar tudi uporaba najboljših varnostih elementov in standardov ni smiselna, če je samo omrežje ranljivo.

Ker tehnologija veriženja blokov predstavlja porazdeljeno bazo podatkov, je edino smotrno, da takšno tehnologijo uporabimo v omrežju enakovrednih partnerjev (*ang. Peer To Peer oz.*

[7] Poglavje je povzeto po [15] in [13].

krajše P2P newtork) in se s tem znebimo kakršne koli centralne točke, ki bi bila lahka tarča za napad. V takšno omrežje se torej lahko vključi vsakdo, zato je potrebno predpostaviti, da bodo v omrežju tudi nepoštena vozlišča, ki ne bodo sledila pravilom omrežja in se skušala okoristiti ob ponujeni priložnosti. Takšno aktivnost moramo onemogočiti brez posegov v decentralizirano naravo omrežja, kar pa je mogoče le s predpostavko, da bo večina v omrežju poštena in zatorej sprejemala pravilne odločitve. Večino v takšnem omrežju ne predstavlja dejansko število vozlišč, vendar je tukaj pomembna njihova računska moč, saj lahko le tako izvedejo tako imenovani 51 % napad.

To je napad, ko nepoštena vozlišča nadzirajo več kot 50 % računske moči in si tako na dolgi rok zagotovijo hitrejše generiranje novih blokov kot tudi potencialno spreminjanje že obstoječih. S tem si pridobijo moč zavračanja veljavnih transakcij, dvojnega zapravljanja že porabljenih sredstev, neupoštevanja trenutnih pravil težavnosti, ... Tak napad bi torej povzročil prisiljen razcep verige (*ang. hard fork*), kjer bi poštena vozlišča še vedno hranila lastno verzijo verige blokov in sledila starim pravilom v omrežju, medtem ko bi napadalci gradili lastno verzijo verige. Vse neveljavne transakcije, ki so jih izvedla nepoštena vozlišča v verigi blokov poštenih vozlišč, sploh ne bi obstajale, zato bi si lahko omrežje po takšnem napadu dokaj hitro opomoglo, vendar bi nedvomno bil na udaru njihov ugled. Prav tako se je pomembno zavedati, da je takšen napad pri velikih in uveljavljenih verigah blokov zelo malo verjeten, saj bi zahteval ne samo ogromno količino zmogljive strojne opreme, ampak bi bil tudi neekonomičen zaradi ostrega padca cene kriptovalute po takšnem napadu.

Obstaja še nekaj napadov, ki lahko vplivajo na omrežje, denimo Sybilov napad, ko poštena vozlišča obkrožimo z zlonamernimi, ta pa poštenemu vozlišču onemogočijo uspešno povezavo do drugih vozlišč in ga tako osamijo. Napadalec se lahko posluži tudi DoS napada, kjer bi določeno vozlišče s prevelikim pošiljanjem podatkov preobremenil in ga tako onesposobil. Takšen napad bi imel le lokalen in začasen vpliv, vozlišče se lahko tudi ustrezno zaščiti z omejevanjem količine prejetih podatkov, velikosti podpisov, zavračanjem nestandardnih transakcij in blokiranjem vozlišč, ki ne sledijo pravilom omrežja. Nenazadnje obstaja tudi verjetnost razvoja kvantnega računalnika, ki dokazano lahko zlomi uporabljeno asimetrično kriptografijo in s tem ustvari veljavne podpise transakcij brez lastništva zasebnega ključa. Ta

problematika je splošno znana, zato večina velikih omrežji že išče rešitve, ki bodo odporne pred napadi kvantnih računalnikov.

Poleg malo verjetno uspešnih napadov na omrežje se je pomembno zavedati, da so do sedaj bili najuspešnejši tisti napadi, ki so se osredotočili na (ne)zmožnost posameznika po ustreznem shranjevanju in varovanju lastnih ključev. Ker tehnologija omogoča izvedbo transakcije vsakomur, ki lahko s pomočjo podpisa dokaže lastništvo zasebnega ključa, so se napadalci osredotoči na razne napade, ki od uporabnika pridobijo njihov zasebni ključ. Da bi se izognili takšnemu napadu, je pomembno, da zasebnega ključa nimamo shranjenega na kakršnem koli mediju povezanem v svetovni splet, znan je samo nam in ga pod nobenim pogojem ne posredujemo tretji osebi. Nenazadnje je pomembno tudi, da ključ hranimo na varnem, a znanem mestu, saj je v primeru izgube ključa okrevanje sredstev skoraj nemogoče.

Uporaba omrežja je torej zelo varna, saj je večina napadov skoraj neizvedljivih. Največ za varnost omrežja lahko nenazadnje naredi vsak posameznik, tako da pravilno rokuje z zasebnim ključem in krepi omrežje s pomočjo vzpostavitve lastnega vozlišča.

2.5 Zasebnost[8]

Eden izmed privlačnih vidikov uporabe kriptovalut predstavlja tudi nivo zasebnosti, ki jih takšna omrežja ob pravilni uporabi lahko nudijo. Tehnologija veriženja blokov od uporabnika ne zahteva nikakršnega razkritja identitete za uporabo, vendar so se zaradi številnih primerov pranja denarja razvila podjetja namenjena analizi zgodovine transakcij. Ta lahko ob zadostni količini podatkov in nepremišljeni rabi s strani uporabnika povežejo naslov z dejansko osebo. Sprva se takšen nadzor ne zdi problematičen, morda celo upravičen, vendar ob pogledu na širšo sliko kaj kmalu ugotovimo, da kriptovalute v represivnih državah državljanom predstavljajo edini način ohranitve osnovne finančne avtonomije.

Predpostavimo, da živimo v totalitarni državi, kjer želimo podpreti organizacijo, ki se zavzema za človekove pravice. Donacija prek nakazila ni mogoča, saj bi takšna transakcija bila preprosto izsledljiva. S tem namenom se odločimo uporabiti eno izmed kriptovalut. Ko izvedemo nakazilo, se takšna transakcija trajno shrani v verigo blokov in je prosto dostopna. Sprva nam

[8] Poglavje je povzeto po [14].

takšna transakcija ne poda skoraj nobenih podatkov o identiteti plačnika, dokler ta isti plačnik ne izvrši nakazila svojemu prijatelju. S tem je prijatelj pridobil njegov naslov, s pomočjo katerega si lahko ogleda celotno zgodovino njegovih transakcij. Tako je posameznik z le eno samo transakcijo povezal svojo identiteto s svojim naslovom, kar posledično razkrije vse transakcije, ki jih je kadarkoli opravil. Tej problematiki bi se lahko preprosto izognili z uporabo novega naslova za izvedbo posamezne transakcije.

Navkljub uporabi novega naslova za izvedbo transakcije še vedno nismo povsem varni pred razkritjem identitete, saj mora biti vsaka transakcija poslana v omrežje, kar razkrije naš IP naslov. Posledično je mogoče s pomočjo IP naslova povezati identiteto osebe kot tudi njene denarnice. Našemu ponudniku internetnih storitev je omogočeno tudi blokiranje dostopa do omrežja kriptovalute, s čimer bi nas lahko izolirali in nam onemogočili izvedbo transakcije. V takšnem primeru je vredno uporabiti povezavo prek omrežja Tor, saj je razkritje naše identitete zelo oteženo.

3 LASTNA IMPLEMENTACIJA TEHNOLOGIJE VERIŽENJA BLOKOV

Za strukturo naše implementacije tehnologije veriženja blokov smo se odločili zgledovati po tehnologiji bitcoina, saj nam je bila ta najbolj poznana. Prav tako je delovanje v primerjavi z Etheriumom preprostejše, verjetno tudi varnejše in zanesljivejše, saj ne omogoča pametnih pogodb ter kompleksnejših transakcij, ki so lahko potencialno nevarne. Nenazadnje je struktura bitcoina povsem zadostna za potrebe izdelave lastne verige blokov za decentralizirano objavo novic. V nadaljevanju poglavja je podrobneje predstavljena tehnična zgradba uporabljenih osnovnih struktur. Celotna zgradba je zgrajena po načelu objektno usmerjenega programiranja, zato vsak del predstavlja enega ali več objektov. S tem smo zagotovili modularnost, kar nam je omogočilo sočasno pisanje programske kode kot tudi povečalo preglednost in nadgradljivost kode v prihodnosti.

3.1 Transakcije

Transakcijo smo razdelali na tri dele. Sprva smo izdelali glavo transakcije, ki vsebuje reference, ki kažejo na vhode in izhode transakcije ter jo kvantificirajo. Struktura glave je razvidna iz Tabele 1.

Tabela 1: Struktura glave transakcije

Atribut	Podatkovni tip v C#
Število vhodnih transakcij	int
Polje vhodnih transakcij	List<Transaction_Input>
Število izhodnih transakcij	int
Polje izhodnih transakcij	List<Transaction_Output>

Kvantifikacija se morda zdi nesmiselna, saj je ta podatek enak dolžini polja vhodov oziroma izhodov, vendar je s tem še dodatno zmanjšana verjetnost naknadnega spreminjanja podatkov transakcije. Oba kvantifikatorja sta podatkovnega tipa int, kar teoretično omogoči transakcije z več kot 4 milijardami vhodov in izhodov. Za shranjevanje dejanskih vhodov in izhodov transakcij smo uporabili List, ki je po delovanju zelo podoben poljem (*ang. array*), vendar je

prijaznejši za uporabo pri pogostem spreminjanju podatkov, saj dinamično dodeljuje pomnilnik. V njih hranimo objekte tipa vhodna oziroma izhodna transakcija, katerih zgradba je predstavljena v nadaljevanju.

Vsaka transakcija je sestavljena iz vsaj enega izhoda. Izhod transakcije vsebuje podatke o vrednosti, dolžini skript, skript za porabo sredstev in neobvezno polje za vnos poljubnega besedila, kar je v naši verigi predvideno za objavo novice. Ker izhod predstavlja prenos vrednosti iz ene denarnice v drugo, je pomembno, da je vsota vrednosti vseh izhodov manjša ali enaka vsoti vseh vrednosti vhodov v transakcijo, kar predstavlja še eno varovalko za preprečevanje porabe neobstoječih sredstev.

Atribut skript vsebuje zahteve, ki jih mora naslednji pošiljatelj izpolniti, če želi sredstva porabiti. Zaradi same preprostosti in varnosti bo naše omrežje omogočalo izključno preproste standardne transakcije, ki zahtevajo le, da pošiljatelj ustvari podpis, katerega veljavnost se preveri s pomočjo navedenega javnega ključa. V prihodnosti je atribut skript mogoče nadgraditi, tako da omogoča tudi kompleksnejše transakcije, predvsem transakcije, ki zahtevajo več podpisov za porabo sredstev. Atribut dolžina skripta v bajtih predstavlja dodatno varnostno varovalko, ki še dodatno oteži spreminjanje skripta.

Atribut besedilo omogoča vnos poljubnega besedila, vendar je zaradi lepšega oblikovanja same novice po objavi priporočena objava besedila skupaj z oznakami HTML5. Označevalni jezik omogoča določanje strukture dokumenta, naslovov, odstavkov, metapodatkov, pisave itd. Ker bo aplikacija za prikazovanje novic delovala kot spletna aplikacija, je omogočeno dodatno oblikovanje tudi s pomočjo CSS, ki omogoča še dodatne modifikacije v izgledu objavljene novice. Izbira glede načina zapisa novice je povsem prepuščena samemu avtorju novice in ne bo imela vpliva na vsebine, saj bo novica ne glede na izgled trajno objavljena, vendar obstaja velika verjetnost, da bodo lepše oblikovane novice že zaradi same preglednosti dobile več ogledov. Prav tako se je potrebno zavedati, da bo novica napisana v HTML in potencialno CSS za isto količino besedila novice zasedla več prostora, kar pomeni, da bo takšna novica za prostor v bloku morala plačati sorazmerno več glede na svojo velikost. Nenazadnje bi bilo nemogoče uvesti kakršnekoli omejitve glede objavljene vsebine, zato je omogočeno pisanje tudi dinamičnih aplikacij z uporabo denimo JavaScripta. Takšna uporaba verige blokov

primarno ni namen izdelane verige blokov, vendar se je potrebno zavedati možnosti, da je vsako objavo povsem nemogoče cenzurirati brez uvedbe centralne entitete.

Tabela 2: Struktura izhoda transakcije

Atribut	Podatkovni tip v C#
Vrednost transakcije	double
Dolžina skripta v bajtih	int
Skript	List<string>
Besedilo	string

Vsaka transakcija mora biti sestavljene iz vsaj enega vhoda. Ta se mora zaradi zaščite pred porabo neobstoječih sredstev sklicevati na do sedaj še neuporabljen izhod že obstoječe transakcije – UTXO (*ang. Unspent Transaction Output*). S tem je omogočena popolna sledljivost izvora kovancev do njihovega nastanka v izvorni transakciji, ko jih je rudar prejel kot nagrado za uspešno rešitev dokaza o delu. Sklic na prejšnjo transakcijo je mogoč s pomočjo reference zgoščene vrednosti izvorne transakcije, kot tudi navedena zaporedna številka izhoda. Ta je pomembna, saj lahko ima ena transakcija več kot en izhod.

Za veljavnost transakcije je zelo pomemben tudi podpis, saj iniciator transakcije z njim dokaže lastništvo zasebnega ključa, ki mu daje izključno pravico do porabe sredstev. Pošiljatelj transakcije podpis ustvari s pomočjo funkcije algoritma za kriptografijo eliptičnih krivulj, ki s pomočjo zgoščene vrednosti transakcije kot tudi zasebnega ključa ustvari podpis. Ob pošiljanju transakcije pošiljatelj poleg podpisa poda še celoten javni ključ. Sprva se v omrežju preveri ujemanje zgoščene vrednosti podanega javnega ključa z navedenim v skripti prejšnje transakcije. V primeru ujemanja se s pomočjo javnega ključa preveri še podpis. Če funkcija za preverjanje podpisa vrne zgoščeno vrednost trenutne transakcije, se takšen vhod v transakcijo šteje kot veljaven, saj je pošiljatelj dokaz lastništvo monetarne vrednosti.

Ker se vhod transakcije navezuje na prejšnji izhod, ki lahko vsebuje novico, lahko na takšen način pričnemo nit novic. To se izkaže še posebej uporabno, ko želi uporabnik poročati o

nečem, kar še ni dokončno oziroma želi popraviti ali dodatno objasniti preteklo objavo. Nit novice se lahka uporabi tudi v smislu dopolnjevanja s strani drugih uporabnikov. Za takšno uporabo se mora prvotni avtor objave odločiti sam, saj mora v izhodu transakcije z novico atribut skript pustiti prazen. S tem omogoči vsakomur ustvarjanje nove transakcije, ki vsebuje njegovo novico kot vhod.

Tabela 3: Struktura vhoda transakcije

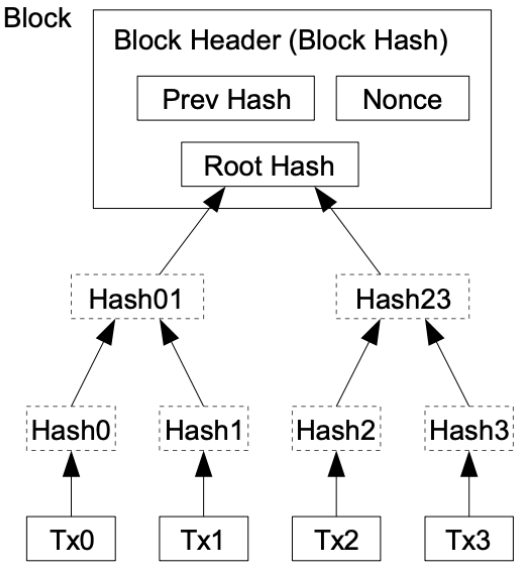
Atribut	Podatkovni tip v C#
Zgoščena vrednost prejšnje transakcije	string
Zaporedna številka izhoda v prejšnji transakciji	int
Dolžina podpisa v bajtih	int
Podpis	string

3.2 Blok

Vsak blok sestoji iz dveh delov, in sicer glave, ki vsebuje vse potrebne podatke za enolično določitev vsebine bloka, in telesa, ki vsebuje natančne podatke o vsaki izvedeni transakciji. Skupna velikost podatkov shranjenih v bloku je omejena na 1 MB, kar pri ustvarjenem novem bloku na 10 min pomeni največjo letno rast velikosti verige blokov za 52 GB. S tem je omogočena participacija tudi posameznikom, katerih internetne hitrosti ne presegajo nekaj 100 kb/s. Za takšno omejitev smo se odločili, saj želimo v čim večji meri spodbujati decentralizacijo, ki je mogoča le v primeru, da se lahko kar se da veliko vozlišč pridruži omrežju.

Natančna struktura glave bloka je razvidna iz Tabele 4. Prvi potreben atribut iz glave je zgoščena vrednost prejšnjega bloka. Ta je pomembna, če želimo doseči verigo med seboj povezanih blokov. Zgoščena vrednost prejšnjega bloka je enaka dvakrat zgoščeni vrednosti glave prejšnjega bloka z algoritmom SHA256.

Naslednja pomembna vrednost je zgoščena vrednost vseh transakcij bloka oziroma Merkle root zgoščena vrednost. Ta vrednost enolično določa vse transakcije v bloku brez potrebe po hranjenju vsebine transakcij. S tem prihranimo predvsem čas in procesorsko moč, ko želimo izračunati zgoščeno vrednost bloka, saj transakcije zasedejo večino velikost bloka, medtem ko Merkle root zgoščena vrednost zavzame le 32 bajtov. To vrednost dobimo s pomočjo združevanja parov zgoščenih vrednosti transakcij v drevesni strukturi, dokler nam ne ostane le ena zgoščena vrednost. V primeru lihega števila transakcij se transakcija združi sama s seboj. Zaradi drevesne strukture je dodajanje nove transakcije zelo učinkovito, saj ni potrebe po ponovnem izračunu celotnega drevesa, vendar moramo spremeniti le tisto vejo, na katero nova transakcija dejansko vpliva. Grafični prikaz strukture za izračun Merkle root zgoščene vrednosti je razviden iz Slike 3.



Slika 3: Postopek izračuna Merkle root zgoščene vrednosti [3, p. 4]

Naslednji podatek iz glave je unix čas. Ta predstavlja število sekund, ki so pretekle od polnoči 1. januarja 1970. Podatek o času je potreben, saj lahko le tako omrežje neodvisno izračuna prilagoditev težavnosti. Ravno iz tega razloga je potrebno poskrbeti za čim večjo točnost tega podatka, zato mora biti v veljavnem bloku podatek o času striktno večji od mediane časov prejšnjih 11 blokov, hkrati pa ti ne smejo biti za 2 uri večji od trenutnega časa, ki ga ima nastavljeno vozlišče.

Atribut nBits v glavi bloka podatkov nam pove, kakšna je težavnost omrežja oziroma postavlja zgornjo mejo, od katere mora biti zgoščena vrednost bloka manjša. Ta vrednost je zaradi učinkovitosti predstavljena kot uint, ki v programskem jeziku C# zavzame 4 bajte, kar je občutno manj od 32 bajtne velikosti zgoščene vrednosti bloka. nBits spremenimo v 32 bajtno velikost s pomočjo naslednje formule $T = k \cdot 256^{e-3}$, kjer k predstavljajo zadnji 3 bajti vrednosti nBits, medtem ko e predstavlja prvi bajt te vrednosti. Za veljavnost bloka je prav tako pomembno, da je vrednost nBits enaka vrednosti prejšnjega bloka, razen če zaporedna številka bloka večkratnik števila 2016. V takšnem primeru se težavnost spremeni glede na hitrost ustvarjanja novih blokov v prejšnjem obdobju. Pri spremembi težavnosti je pomembno, da ta sledi pravilom omrežja in za izračun uporabi formulo

$$new\ nBits = old\ nBits \cdot \frac{potreben\ čas\ za\ rudarjenje\ zadnjih\ 2016\ blokov}{2016 \cdot 10}.$$

Zadnji potreben atribut v glavi je nonce oziroma enkratna vrednost, ki je namenjena spreminjanju zgoščene vrednosti bloka. Pomembna je predvsem za rudarje, ki se trudijo proizvesti veljaven blok, saj lahko to vrednost povsem poljubno spreminjajo brez nulifikacije veljavnosti bloka. Ker je podatkovnega tipa uint, je njen razpon od 0 pa vse do več kot 4 milijard. V primeru, da bi rudar preizkusil vse mogoče vrednosti nonca in še vedno ne bi ustvaril veljavnega bloka, se lahko v takem primeru odloči na novo razporediti transakcije, kar povsem spremeni zgoščeno vrednost Merkle root ter ima ponovno na voljo novih še nepreizkušenih 4 milijard noncev. Trenutna zasnova glave rudarju omogoča $(n-1)! \cdot 2^{32}$ različnih kombinacij na sekundo, kjer n predstavlja število transakcij bloka, in ob predpostavki, da nobena od transakcij v bloku ni odvisna ena od druge. S trenutno strojno opremo se v eni sekundi ni mogoče približati izračunu vseh možnih kombinacij. Nenazadnje v resničnem svetu rudar stalno prejema nove transakcije, občasno je transakcij tudi preveč, da bi jih bilo mogoče spraviti v en sam blok, s čimer se število mogočih kombinacij še samo povečuje.

Tabela 4: Struktura glave bloka

Atribut	Podatkovni tip v C#
Zgoščena vrednost prejšnjega bloka	string

Merkle root zgoščena vrednost	string
Čas	uint
nBits	uint

3.3 Razporeditev sredstev

Zaradi decentralizacije je varnost, zanesljivost in neodvisnost verige blokov povsem odvisna od števila vozlišč in od njihove procesorske moči. Ker se akterji v omrežju soočajo z raznoraznimi stroški, predvsem s stroški nakupa strojne opreme, porabljene električne energije in vzdrževanjem internetne povezave, je potrebno uvesti sistem nagrajevanja, ki bo najaktivnejše nagrajeval in posledično privabljal vedno večjo množico k sodelovanju v omrežju. Nagrajevanje v obliki uradnih državnih valut ni mogoče, saj je odvisno od finančnih sredstev centralne avtoritete, ki bi si s tem utrdila položaj glavnega odločevalca v omrežju, katerega bi bilo potrebno zadovoljiti za izplačilo nagrade. Iz tega razloga se je zelo velika večina obstoječih verig blokov odločila nagrade izplačevati v nativni valuti verige blokov. Takšno nagrajevanje ima poleg neodvisnosti nagrajevanja od razvijalcev verige blokov prednost tudi v tem, da spodbuja aktivnejšo uporabo in delno lastništvo, kar spodbuja pošteno ustvarjanje blokov zaradi korelacije med monetarno vrednostjo nagrade in zanesljivostjo delovanja tehnologije.

Pri nekaterih projektih so si razvijalci v izvornem bloku (*ang. genesis block*) zagotovili začetni delež ustvarjene valute, ki jo kasneje lahko prodajo na prvih dražbah imenovanih tudi ICO (*ang. initial coin offering*). Pri sistemih zagotavljanja varnosti verige blokov, kot je denimo dokaz lastništva (*ang. proof of stake*), je takšna začetna prerazporeditev nujna za začetek delovanja omrežja, vendar v nobenem pogledu ni pravična, kaj šele neodvisna od centralne avtoritete. Nenazadnje so si razvijalci zagotovili večni odločevalni delež v omrežju, s čimer imajo zelo velik vpliv nad nadaljnjim razvojem. Sami smo se želeli oddaljiti od ustaljenih praks, zato smo se kljub pomislekom o učinkovitosti delovanja dokaza o delu odločili za njegovo implementacijo. S tem je participacija od samega začetka omogočena vsakomur, z izjemo izvornega bloka, ki ga za samo inicializacijo verige blokov moramo zrudariti sami.

Nagrade v omrežju so izplačane vedno v prvi transakciji bloka (*ang. coinbase transaction*). Izplačana nagrada je sestavljena iz novo ustvarjenih kovancev in potencialnih nagrad, ki so jih uporabniki dodali transakcijam in si s tem zagotovili prioriteto pri vključevanju transakcije v novo nastali blok. Od začetka delovanja bodo novo ustvarjeni kovanci predstavljali veliko večino nagrade, vendar se bo to razmerje sčasoma obrnilo, saj se količina na novo ustvarjenih kovancev čez čas zmanjšuje. S tem želimo zagotoviti čim manjšo inflacijo, ko se omrežje razširi, hkrati pa v večji meri nagraditi tiste, ki so v omrežju sodelovali že od začetka. Količina nagrade se izračuna s pomočjo naslednje formule $r = 50 \cdot (\frac{1}{2})^{\left\lfloor \frac{h}{210\,000} \right\rfloor}$, kjer r predstavlja količino na novo ustvarjenih kovancev, h pa število že ustvarjenih blokov. Število 210.000 je izbrano, saj predstavlja približno število na novo ustvarjenih blokov v obdobju 4 let. Takšno ustvarjanje novih kovancev zagotavlja dolgoročno deflacijo, katere posledica je pogosto dvig cene valute na enoto. Če v omrežju postavimo pravila, da je najmanjša vrednost, ki jo je mogoče prenesti, enaka vrednosti 10^{-8}, to pomeni, da v omrežju nikoli ne bo več kot 21 milijonov enot. Do te številke pridemo tako, da najprej izračunamo $50 \cdot \left(\frac{1}{2}\right)^{n-1} > 10^{-8}$, s čimer ugotovimo, da se nagrada lahko prepolovi le 33-krat, če želimo, da je vrednost nagrade še vedno večja od najmanjše dovoljene vrednosti v omrežju. Za vsoto vseh ustvarjenih kovancev to vrednost vstavimo še v formulo $\sum_{n=1}^{33} 210\,000 \cdot 50 \cdot \left(\frac{1}{2}\right)^{n-1} = 2{,}1 \cdot 10^{7}$.

Ena izmed slabosti deflacijskega modela so vedno večji dodatki, ki jih bodo uporabniki morali dodajati transakcijam, če bodo želeli, da je njihova transakcija vključena v blok. S tem se bo sčasoma zagotovila večja kvaliteta objavljenih novic, saj finančno ne bo več smotrno objavljati brezpredmetnih in nekvalitetnih novic, saj prostor za hranjenje verigo blokov ni neskončen.

3.4 Uporabljena kriptografija[9]

V prejšnjem poglavju je bilo omejeno, da je vsak vhod v transakcijo potrebno kriptografsko podpisati tako, da zadovolji kriterije, ki so bili podani v izhodu starševske transakcije. S kriptografskim podpisom lahko v zelo veliki meri preprečimo nepooblaščeno porabo sredstev

[9] Poglavje je povzeto po 5. poglavju [1].

z drugega računa, saj lahko veljaven podpis poustvari le tisti, ki je lastnik ustreznega zasebnega ključa.

Takšno podpisovanje je del asimetrične kriptografije, t. i. kriptografija, kjer se za šifriranje vsebine uporablja javni ključ in za dešifriranje zasebni ključ. Zasebni ključ omogoča generiranje javnega ključa, zato zaradi varnosti kriptografija pomembna tajnost le-tega. Poznamo več različnih asimetričnih kriptografskih algoritmov, med drugim RSA, Diffie-Hellman, mrežna kriptografija in nenazadnje eliptične krivulje. Slednje smo uporabili kot primarno kriptografijo v naši verigi blokov, saj so od dandanes zelo razširjene hkrati pa ponujajo visoko stopnjo varnosti tudi nekaj desetletji v prihodnost.

Kriptografija z uporabo eliptičnih krivulj deluje na osnovi končnih cikličnih grup, ki jih tvorijo te krivulje in točka na njej, imenovana generator, pod posebnimi pogoji. Zasebni ključ predstavlja poljubno število $k \in \mathbb{N}$, da bo veljalo $1 \leq k < n$, kjer je n red eliptične krivulje. Javni ključ R izračunamo s pomočjo zasebnega, tako da generator grupe G množimo s skalarjem k. Rezultat takšne operacije je prav tako točka, ki je element iste končna ciklične grupe. Vrednost javnega ključa lahko še dodatno poenostavimo, tako da vrednost ordinate preprosto ovržemo, saj jo lahko po potrebi vedno ponovno izračunamo. Izračunajmo še vrednost $r = G\ (mod\ n)$, ki bo potrebna pri podpisovanju.

S pomočjo ustreznega para ključev lahko ustvarimo veljaven podpis naše transakcije, s čimer dokažemo lastništvo zasebnega ključa in si posledično zagotovimo pravico zapraviti sredstva. Najprej moramo izračunati zgoščeno vrednost transakcije h, ki jo dobimo tako, da v funkcijo za izračun SHA256 vrednosti kot vhod podamo vse podatke transakcije razen podatkov o podpisih. Podpis dobimo s pomočjo formule[10] $s \equiv k^{-1} \cdot (h + r \cdot k)\ (mod\ n)$ in ga dodamo vhodu transakcije skupaj z našim javnim ključem. Takšna transakcije je ob uporabi ustreznega zasebnega ključa veljavna, zato jo lahko posredujemo v omrežje. Ko vozlišče prejme našo transakcijo, bo med drugim preverilo tudi veljavnost vseh podpisov.

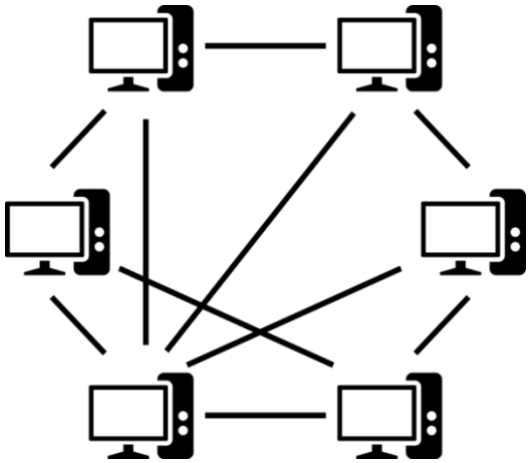
Podpis preveri tako, da najprej izračuna zgoščeno vrednost podanega javnega ključa in jo primerja s tisto navedeno na prejšnjem izhodu. Če se vrednosti ujemata, vozlišče nadaljuje s

[10] Pri operacijah znotraj končnih cikličnih grup vrednost a^{-1}, kjer je $a \in \mathbb{N}$, predstavlja multiplikativni inverz vrednosti a, torej bo veljalo $a \cdot a^{-1} \equiv 1\ (mod\ n)$.

preverjanjem veljavnosti podpisa. Najprej izračuna zgoščeno vrednost transakcije h. Nato s pomočjo formule $(h \cdot s^{-1}\ (mod\ n)) \cdot G + (r \cdot s^{-1}\ (mod\ n))$ preverimo veljavnost podpisa. Če je dobljena vrednost enaka vrednosti javnega ključa (natančneje vrednosti r), se podpis smatra kot veljavnega.

4 OMREŽJE ENAKOVREDNIH PARTNERJEV[11]

Eden izmed aspektov, ki zagotavlja decentralizacijo in odpornost omrežja pred napadi na centralno točko, je gotovo tudi uporaba omrežja enakovrednih parterjev. V večini dandanes uporabljenih omrežjih se poslužujemo modela odjemalec-strežnik, kjer se nam nadrejeni strežniki odzivajo na naše zahteve in nam posredujejo podatke. Takšen način delovanja omogoča bolj optimalno delovanje, saj podatke hrani le nekaj naprav in tako ne pride do shranjevanja redundantnih kopij. Največja slabost takšnega modela predstavlja centraliziranost, saj se ti strežniki odločajo kaj in kdaj nam bodo posredovali podatke ter hranijo zapise o napravah, ki so želele dostop do podatkov. Nenazadnje je razpoložljivost podatkov povsem odvisna od delovanja teh centralnih entitet, zato je takšno omrežje veliko bolj ranljivo na napade.



Slika 4: Prikaz povezav med napravami v omrežju enakovrednih partnerjev [5]

Temu se pri uporabi tehnologije veriženja blokov za objavo necenzuriranih novic želimo v čim večji meri izogniti, zato smo se poslužili omrežja enakovrednih partnerjev (*ang. peer-to-peer network oz. P2P*). Takšno omrežje še vedno sestavljajo med seboj povezane naprave, vendar v takšnem omrežju ni mogoče najti centralne, nadrejene točke, saj njeno nalogo opravlja vsaka naprav priključena v omrežje. S tem dobimo veliko omrežje, ki je odporno pred napadi na centralno točko, saj se v primeru prekinitve povezave z enim računalnikom podatki preusmerijo na drugega, sicer bolj oddaljenega, a še vedno dostopnega. Razvejena struktura

[11] Opis delovanja TCP protokola je povzet po [16].

omrežja je razvidna s Slike 4, ki prikazuje povezave med več med seboj enakovrednimi napravami.

Največji problem omrežji enakovrednih partnerjev, poleg redudantnega shranjevanja podvojenih podatkov, predstavlja vzpostavitev povezave med napravami, saj se morajo naprave predhodno med seboj poznati. V javnem omrežju, ki lahko vključuje več tisoč povezav, ukaza za razpršeno komunikacijo (*ang. broadcast*) ni mogoče poslati, tako da odkrivanje drugih naprav v omrežju na ta način ni mogoče. S tem namenom imamo v izvorni kodi vnaprej določen primarni IP naslov računalnika (77.38.109.221:1201), na katerega se lahko obrnejo nove naprave, saj jim bo ta vrnil nekaj drugih vozlišč v omrežju, na katera se lahko povežejo. Poleg primarnega vozlišča se lahko določi še nekaj rezervnih vozlišč, na katere se lahko nova naprava v omrežju zanese za zagotavljanje začetnih povezav. Seveda je takšna začetna povezava v omrežje dokaj centralizirana, vendar omrežje od nove naprave ne zahteva prve povezave v točno določeno točko. Če ima administrator seznam ostalih naprav v omrežju, lahko primarni IP naslov naprave spremeni na drugo vozlišče, ki pri njem uživa večje zaupanje. Takšno ravnanje pozitivno vpliva na delovanje samega omrežja, saj se s tem ne samo razbremeni primarno vozlišče, ampak se tudi spodbuja decentralizacija.

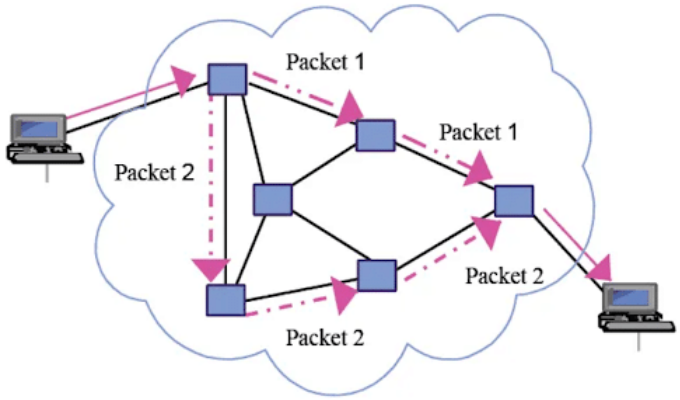
Ko nova naprava enkrat prejme seznam drugih vozlišč v omrežju, z njimi poskusi vzpostaviti povezavo. Če je povezava uspešna, se izmenjava podatkov lahko prične. V primeru, da je povezava z napravo počasna, lahko vozlišče napravo vedno zaprosi za seznam njenih vozlišč, prekine povezavo in se poskusi povezati z drugim vozliščem, ki ponuja ugodnejšo povezavo. Vsaka naprava vedno hrani seznam vseh naprav, s katerimi ima vzpostavljeno povezavo. Za preverjanje aktivnosti povezave si morata napravi izmenjati podatke na 30 minut. V primeru, da se naprava po tem času ne odzove, se povezava smatra kot prekinjena.

V omrežju se lahko znajdejo tudi številna škodljiva vozlišča, ki po omrežju širijo neveljavne bloke, s čimer zmanjšujejo pretočnost omrežja in po nepotrebnem zapravljajo procesorsko moč ostalih vozlišč. V primeru, da naprava zazna, da ji določeno vozlišče večkrat posreduje neveljavne transakcije ali bloke, lahko takšno napravo premakne na seznam neverodostojnih naprav in z njo prekine vsakršno povezavo.

Z vsemi zgoraj opisanimi postopki je nova naprava uspešno vzpostavila povezavo z nekaj vozlišči v omrežju, zato se postopek sinhronizacije verige blokov lahko prične. Naprava izbranemu vozlišču pošlje poizvedbo po že obstoječih blokih, pri čemer poda kot argument zadnji zaporedno številko bloka, ki ga še ima shranjenega v bazi. Pri večini novih naprav bo to zaporedna številka 0, saj je izvorni (*ang. genesis*) blok že umeščen v bazo podatkov. Vozlišče bo nato napravi poslalo prvih 500 blokov od podane zaporedne številke bloka. Ko naprava prejme nov blok, preveri njihovo veljavnost s pomočjo validatorjev in ga v primeru ugodnega izida shrani. Pomembno je omeniti, da se pri prenosu v omrežju lahko zgodi, da paket z blokom, ki ima večjo zaporedno številko, naprava prejme pred blokom z manjšo. Validacija bloka z večjo seveda ni možna, saj bloki tvorijo verigo, zato se ga shrani na začasni seznam še ne potrjenih blokov. Njegova validacija se izvede, ko veriga že validiranih in shranjenih blokov doseže njegovo zaporedno številko. Takšen postopek se ponavlja, dokler novi napravi nihče več ne more posredovati novih blokov. Takrat se izvede še sinhronizacija tabele trenutno še ne potrjenih transakcij. S tem je nova naprava v omrežju postala enakovreden partner vsem drugim in lahko opravlja polno funkcionalnost, denimo posluša za novimi transakcijami in bloki, jih validira, posreduje, ustvarja lastne transakcije, sinhronizira druga vozlišča in najpomembneje izvaja proces rudarjenja, s katerim se poteguje za nagrade v omrežju.

Za vso zgoraj opisano funkcionalnost je uporabljen protokol TCP (*ang. Transmission Control Protocol*), ki omogoča zanesljiv in urejen prenos podatkov v omrežju. Dandanes ga ravno zaradi zanesljivost uporabljajo denimo protokoli aplikacijske plasti, kot so HTTP, SMTP, FTP, SSH in drugi. Pred izmenjavo zahtev in podatkov se morata napravi med seboj vzpostaviti povezavo, kar dosežeta s pomočjo trosmernega usklajevanja (*ang. three way handshake*). Ko sta napravi uspešno povezani med seboj, se izmenjava podatkov lahko prične. Pogosto se zgodi, da so podatki namenjeni pošiljanju preveliki, da bi jih lahko poslali v le enem podatkovnem paketu, zato TCP poskrbi za ustrezno fragmentacijo podatkov na pakete ustrezne velikosti. V glavi prav tako nastavi parametra za zaporedno številko paketa, ki prejemniku omogoča pravilno defragmentacijo podatkov, kot tudi kontrolno vsoto, ki prejemniku omogoča preverjaje celovitosti podatkov. Nenazadnje protokol TCP/IP omogoča izbiro najugodnejše poti za potovanje paketa po omrežju, zato lahko vsak paket na poti do cilja ubere drugačno pot. Potovanje paketov po omrežju je razvidno s Slike 5. V primeru

manjkajočih podatkov prejemnik od pošiljatelja zahteva ponovno pošiljanje le točno določenega neprejetega paketa, zato pravimo, da protokol zagotavlja zanesljiv in celovit prenos podatkov, ki je pri prenosu podatkov o transakcijah ter blokih ključnega pomena. Po končani izmenjavi podatkov se povezave prekine. Za prenos podatkov bi se načeloma lahko poslužili tudi protokola UDP (*ang. User Datagram Protocol*), ki sicer omogoča hitrejši prenos podatkov med napravami, vendar s tem izgubi na zanesljivosti prenosa, saj ne zagotavlja prejema podatkovnih paketov.



Slika 5: Prikaz potovanja podatkovnega paketa po omrežju [6]

Pri izbiri ustreznega protokola za uporabo pri tehnologiji veriženja blokov menimo, da zanesljivost in zagotavljanje celovitosti prenosa podatkov odtehtata slabosti, t. i. počasnejši prenos podatkov. Nenazadnje naša izbira TCP protokola nikogar ne zavezuje k njeni uporabi, saj se lahko brez spreminjanja pravil verige blokov uporabi tudi UDP. Hitrejši prenos v neki meri doprinese k uspešnejšemu rudarjenju, saj lahko pri ustvarjanju blokov štejejo tudi milisekunde. V takšnem primeru seveda pride upoštev kar se da hiter prenos podatkov in čim večja mreža vzpostavljenih povezav z drugimi vozlišči.

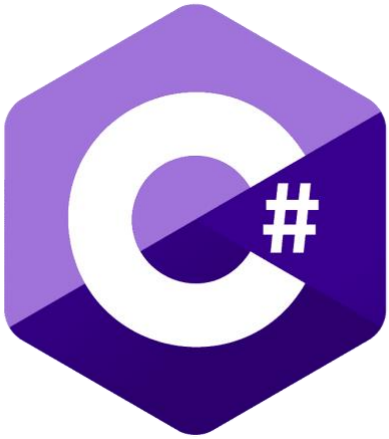
5 UPORABLJENI PROGRAMSKI JEZIKI IN ORODJA

Tekom izdelave verige blokov smo se poslužili raznoraznih programskih jezikov in orodji, ki so nam omogočila uspešno izvedbo projekta. Pri izdelavi zalednega dela, ki vključuje funkcije za preverjanje veljavnosti blokov in transakcij, za pošiljanje in prejemanje podatkovnih paketov, za ustvarjanje transakcij, za zgoščevanje vrednosti itd., smo primarno uporabljali programski jezik C#. Ker novice zaradi lažjega oblikovanja izgleda uporabljajo HTML in CSS, smo aplikacijo za ogled in upravljanje s transakcijami postavili kot spletno aplikacijo. Za dinamično postavitev elementov, kot tudi upravljanje z njimi, smo se poslužili še programskega jezika JavaScript.

Nenazadnje smo tekom razvoja projekta različni razvijalci spisali več tisoč vrstic programske kode. Ročno primerjanje različnih verzij kode in zagotavljanje njenega delovanja bi bila praktično nemogoča, zato smo za verziranje kode uporabili GitHub.

5.1 C#

C# je objektno usmerjen programski jezik, ki ga je javnosti predstavilo podjetje Microsoft leta 2000. Jezik predstavlja najnovejšo verzijo v procesu evolucije jezikov razvitih iz programskega jezika C. Uporabljena sintaksa je napram predhodnikom veliko preprostejša in berljivejša, navkljub dodatni funkcionalnosti. Kombinacija naprednih funkcij in preproste sintakse je očitno na trgu zaželena, saj je se v le nekaj letih zavihtel na vrh lestvice najbolj uporabljenih programskih jezikov. Nenazadnje jezik omogoča relativno hitro izvajanje, sploh v primerjavi s sodobnimi tekmeci, ki se poslužujejo tolmačenja namesto kombinacije delnega prevajanja v vmesni jezik in kasnejšega tolmačenja programske kode.



Slika 6: C# [7]

5.2 Programski jeziki za razvoj spletnih aplikacij

HTML (*ang. HyperText Markup Language*) je eden izmed najbolj razširjenih označevalnih jezikov, ki s pomočjo značk in atributov omogoča postavitev osnovnega ogrodja spletnih dokumentov. Večina značk deluje v paru, torej je vsako začetno značko potrebno tudi ustrezno zapreti. Prva verzija je bila razvita v CERN-u in objavljena leta 1991, vendar se razvoj nadaljuje še danes.

Kmalu je eden od zaposlenih v CERN-u spoznal pomanjkljivost uporabe le HTML, zato je W3C leta 1996 izdal še jezik za pisanje stilskih predlog imenovan CSS (*ang. Cascading Style Sheets*). CSS je še vedno pogosto uporabljen za oblikovanje spletnih strani, saj ponuja preprosto integracijo s HTML ogrodjem. Uporabimo ga lahko za spreminjanje barv, odmikov, stila pisave, postavitve elementov, odzivanje na spremembo lokacije miške, ...

Z razvojem svetovnega spleta se je skozi leta tipična zgradba spletnih strani spremenila iz statičnih v dinamične, saj so razvijalci izkušnjo in prikazano vsebino v čim večji meri želeli prilagoditi vsakemu uporabniku posebej. Pri tem so se poslužili programskega jezika JavaScript, ki se s pomočjo dogodkov, kot so kliki miške, pritiski gumbov ali spremembe vsebine elementa, ustrezno odzivajo na uporabnikovo interakcijo. To je nenazadnje omogočilo izdelavo iger za brskalnike, spletno nakupovanje in razbremenitev strežnikov, saj so se lahko določene zahteve procesirale na računalniku odjemalca.

Slika 7: HTML, CSS in JavaScript [8]

5.3 GitHub

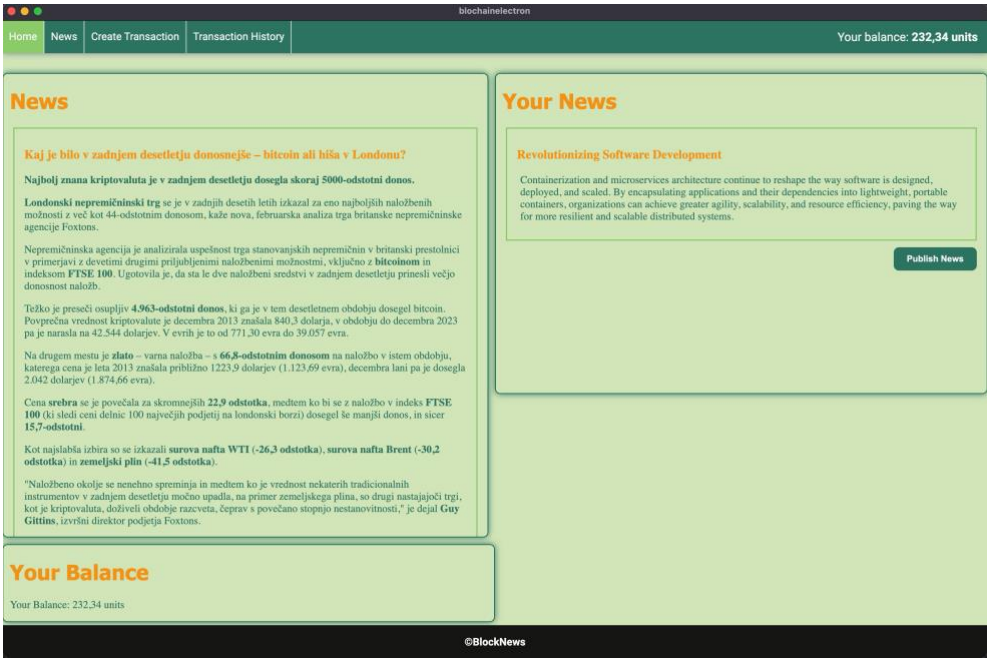
GitHub je pogosto uporabljeno sodelovalno razvojno orodje za razvijalce programske opreme, saj omogoča preprost in efektiven način deljenja kode med več razvijalci, ki želijo sodelovati na istem projektu. Kot osnovo za svoje delovanje uporablja odprtokodni Git, ki je bil razvit leta 2005 z namenom podpore nelinearnim tokovom dela. Poleg verziranja kode med drugim omogoča še denimo shranjevanje kode v oblačnih storitvah, lažjo komunikacijo med razvijalci, razprave o težavah v kodi in pregled kode drugih razvijalcev. Vsa omenjena funkcionalnost je bila zelo pomembna za uspešno dokončanje projekta, saj je zvišala našo produktivnost in učinkovitost.

Slika 8: GitHub [9]

6 PREDSTAVITEV IZDELANE APLIKACIJE

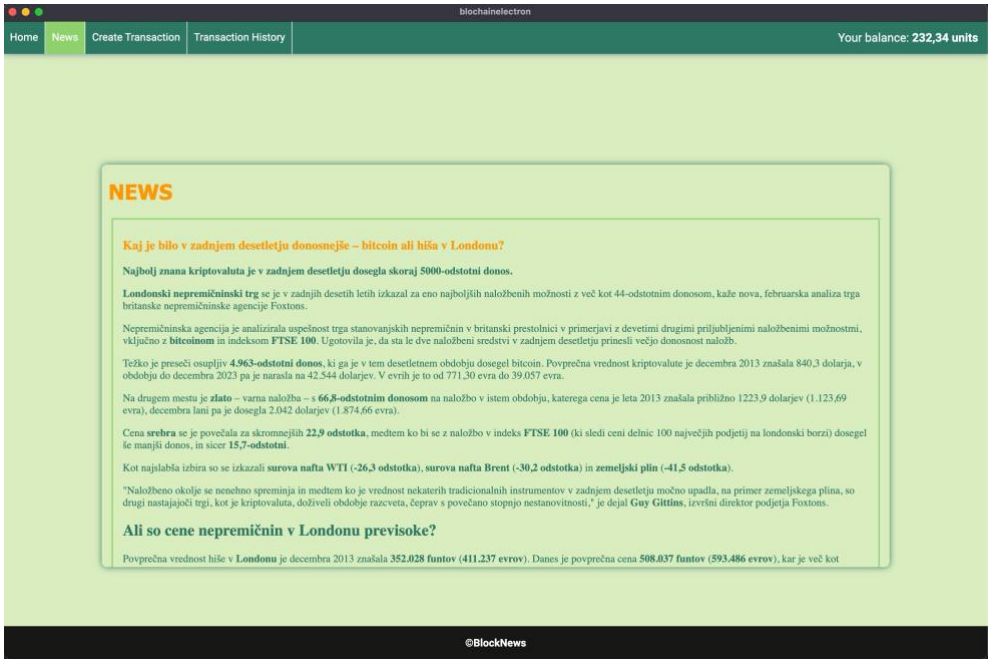
Izdelali smo spletno aplikacijo, ki omogoča osnovno funkcionalnost za interakcijo uporabnika z verigo blokov. Uporabnik aplikacijo zažene, tako da odpre datoteko index.html. Domači zaslon aplikacije je razviden s Slike 9 in je uporabniku viden takoj ob zagonu aplikacije v privzetem spletnem brskalniku. Na vrhu okna ima uporabnik na voljo navigacijsko vrstico, ki omogoča spremembo trenutnega pogleda. Spodaj so na voljo tri okna, ki uporabnika informirajo z najpomembnejšimi informacijami.

Na levi strani najprej vidimo okno imenovano News oziroma Novice, ki prikazuje 10 najnovejših novice drugih uporabnikov. Če je uporabnik novico oblikoval po smernicah za oblikovanje, se bo naslov označen z značko *<h3>* obarval rumeno, prav tako se bo izpisal prvi odstavek vsebine označen z značko *<p>.* V spodnjem desnem kotu okvirčka novice je nenazadnje izpisan datum in čas objave novice. Desno od okvirčka za novice je postavljen še okvirček imenovan Your news oziroma Vaše novice, ki vsebuje 10 najnovejših novic, ki jih je objavil uporabnik. Izpisane so na enak način kot zgoraj opisane novice. Pod seznamom novic je na voljo še gumb Objavi novico, ki uporabnika preusmeri na stran za izvedbo transakcije. Nenazadnje nas v spodnjem levem delu domači zaslon obvešča tudi o trenutnem stanju.

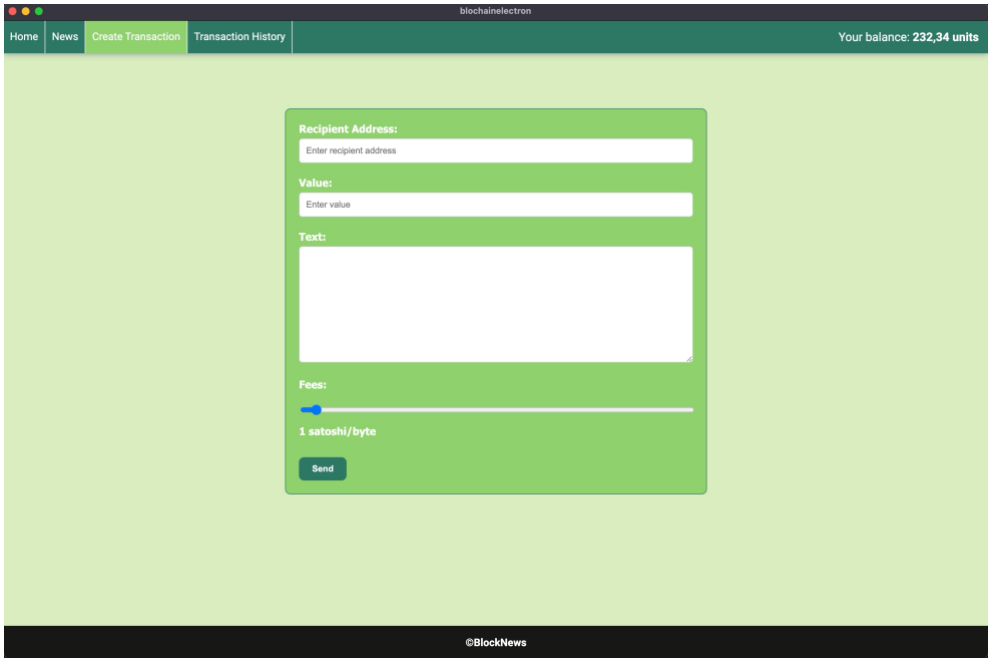


Slika 9: Prikaz primarnega pogleda aplikacije

Naslednji uporabniški pogled razviden s Slike 10 je namenjen prikazovanju objavljenih novic po kronološkem vrstnem redu. Ob kliku na novico se uporabniku prikaže celotna vsebina novice.



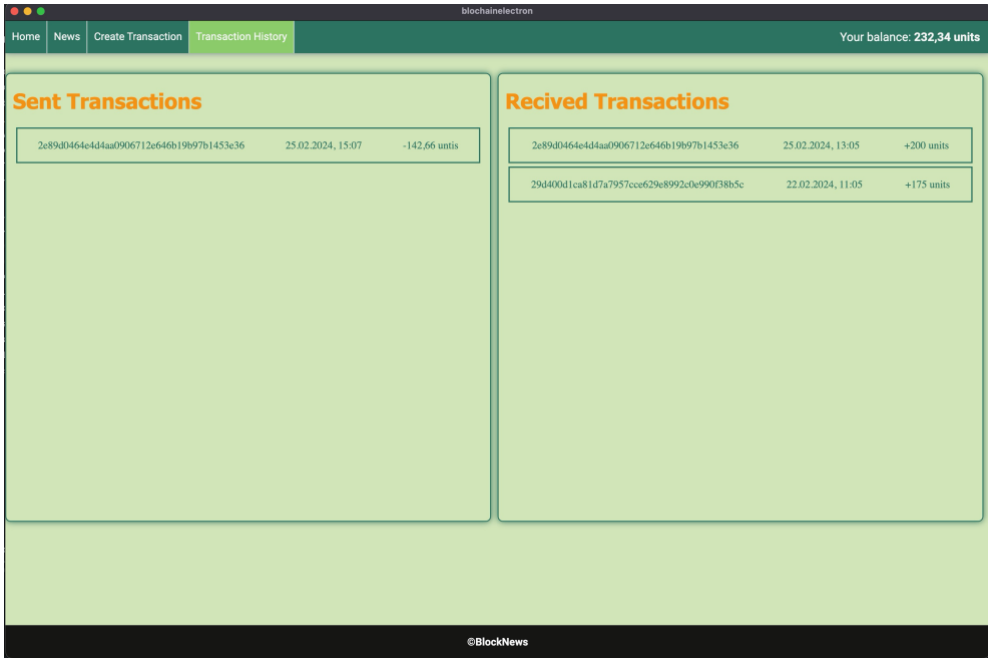
Slika 10: Prikaz pogleda za ogled novic



Slika 11: Uporabniški pogled za pošiljanje transakcije ali objavo novice

Tretji uporabniški pogled razviden s Slike 11 uporabniku omogoča ustvarjanje in pošiljanje transakcij. Za pošiljanje nove transakcije mora uporabnik vnesti podatke o prejemnikovem naslovu, ki predstavlja zgoščeno vrednost njegovega javnega ključa, vrednosti transakcije, ki mora biti večja ali enaka 0, in nastaviti vrednost provizije, ki jo uporabnik rudarju nameni za vključitev transakcije v blok. Enota za določanje provizije je satoshi[12]/bajt, kar omogoča sorazmerno plačilo provizije glede na velikost poslane transakcije oziroma povedano drugače, večjo provizijo plača, kdor objavi daljšo novico. Podatek o besedilu ni obvezen, saj veriga omogoča pošiljanje tudi samo monetarnih transakcij. Za pošiljanje novice uporabnik uporabi gumb Submit, pri čemer se je potrebno zavedati, da kasnejša sprememba transakcije ni mogoča.

Zadnji uporabniški pogled razviden s Slike 12 uporabniku omogoča pregled nad njegovimi transakcijami. Na levi strani lahko uporabnik vidi vse poslane transakcije skupaj s podatki o naslovu prejemnika, časovnem žigu in vrednosti. Na desni strani so uporabniku prikazane prejete transakcije, ki vsebujejo podatke o izvornem naslovu, časovnem žigu in vrednosti.



Slika 12: Uporabniški pogled za pregled lastnih preteklih transakcij

[12] Satoshi je najmanjša enota, ki jo posameznik lahko posreduje v omrežju, in predstavlja 10^{-8} enot kriptovalute.

7 PREDSTAVITEV REZULTATOV RAZISKOVALNE NALOGE

Prva zastavljena hipoteza se nanaša na zmožnost uporabe izdelane verige blokov tako za prenos monetarne vrednosti kot objavo novic, kar ji prinaša konkurenčno prednost v primerjavi z obstoječimi rešitvami. To hipotezo smo delno potrdili. Menimo, da izdelana veriga blokov izpolnjuje začrtane cilje brez sprejemanja varnostnih kompromisov, vendar v tem trenutku še ne more tekmovati z večjimi tekmeci, predvsem zaradi slabše prepoznavnosti.

Na trgu že slabih 10 let obstajajo rešitve s podobnim ciljem, denimo Civil, zgrajen na Etheriumovem omrežju, ki je zaradi neprofitabilnosti pred kratkim prenehal z delovanjem in Minds, ki je družabno omrežje podobno družabnemu omrežju Tweeter brez možnosti cenzure vsebine. Zaenkrat še nobena od rešitev ni pridobila širše prepoznavnosti v javnosti, kar lahko v veliki meri pripišemo trenutni monopolni poziciji, ki jo do neke mere izkoriščajo velika družabna omrežja in medijske hiše. Prav tako večina ljudi ni dovolj osveščena glede tehnologije veriženja blokov, zato se jim pogosto alternative trenutnemu statusu quo zdijo nesmiselne, hkrati od uporabnika zahtevajo več truda, saj bi za popolno neodvisnost uporabnik moral postati vozlišče v omrežju. Nenazadnje so rešitve, ki uporabljajo verige blokov, pogosto počasnejše od centraliziranih kot tudi zaradi svoje narave anonimnosti privabljajo manj zaupljive vire.

Menimo, da je navedena problematika premostljiva, vendar bi to zahtevalo množično željo uporabnikov po obstoju tehnologije za objavo necenzuriranih novic. Kot izboljšavo trenutne implementacije izdelane verige blokov prav tako vidimo sistem za razvrščanje novic, ki bi v praksi dobre in zaupanja vredne novice prikazal pri vrhu. Takšen sistem bi s seboj prinesel tudi številne težave, saj bi bilo potrebno razviti decentraliziran sistem rangiranja. Ena izmed dobrih rešitev je zmožnost uporabnika po monetarni nagradi kvalitetne novice, vendar bi se kaj hitro znašli v podobni situaciji kot danes, kjer je najvplivnejše mnenje tisto, ki ima za sabo največ denarja.

Druga hipoteza se dotakne varnostno-kriptografskega aspekta, ki mora za zanesljivo delovanje ostati nedotaknjen. Za implementacijo sistema podpisovanja smo uporabili shemo ECDSA (*ang. Elliptic Curve Digital Signature Algorithm*), ki je izvedba DSA (*ang. Digital Signature Algorithm*) s pomočjo eliptičnih krivulj. Uporabili smo krivuljo secp256k1, ki je standardizirana

na področju kriptografije, med drugim jo uporablja tudi bitcoin. Ta krivulja ponuja 256 bitno zaščito, ki je v današnjem in bo še v prihodnjem svetu ena varnejših. Največjo nevarnost asimetričnih kriptografskih algoritmov predstavlja prihod zmogljivih kvantnih računalnikov, ki lahko s pomočjo Shorovega algoritma zlomijo celotno kriptografijo. Takšen napredek na področju kvantnega računalništva ni moč pričakovati še vsaj nekaj let, vendar bi hiter napredek negativno vplival na celotno računalniško infrastrukturo, saj alternative, odporne pred kvantnimi računalniki, denimo kriptografija na osnovi mreže (*ang. lettice based cryptography*), še niso pogosto uporabljene. Kasnejša sprememba uporabljenega kriptografskega algoritma v omrežju je mogoča, vendar bi gotovo povzročila razcep verige, kar bi negativno vplivalo na omrežje, ki bi se čez noč prepolovilo. Podrobnejša razlaga delovanja in odpornosti pred kvantnimi računalniki je na voljo v naši lanskoletni raziskovalni nalogi [1]. Na podlagi zgoraj navedenih razlogov drugo hipotezo ovržemo, saj navkljub zagotavljanju visokega nivoja varnosti še obstaja nevarnost napada kvantnih računalnikov, ki bi povsem uničili integriteto transakcij v omrežju.

Tretja hipoteza se nanaša na trajnostni vidik obstoja verige blokov v realnem svetu, saj se bodo tako rudarji kot vozlišča srečevala z raznoraznimi operativnimi stroški. Porabo električne energije za poganjanje vozlišča v omrežju je težko oceniti, saj je v veliki meri odvisna od uporabljene strojne opreme in cene električne energije, vendar znesek ne bi smel presegati 10 evrov. Znesek se lahko poveča tudi za nekajkrat, če vozlišče aktivno sodeluje tudi pri procesu rudarjenja. Poraba rudarja je povsem odvisna od računalniške moči, ki mu jo nameni, uporabljene strojne opreme, nivoja optimizacija programa za rudarjenje, itd. Del stroškov predstavlja tudi nakup in vzdrževanje strojne opreme ter stroški vzdrževanja dostopa do svetovnega spleta.

Z namenom minimiziranja operativnih stroškov smo v protokol vgradili sistem izplačevanja nagrad v obliki novo ustvarjenih kovancev kriptovalute. Ti kovanci pripadajo rudarju, ki je uspešno ustvaril nov veljaven blok v omrežju. Sprva je pričakovati skoraj ničvrednost kovancev, vendar bo cena narasla, če se seveda pojavi zanimanje za uporabo verige blokov. To bi omogočilo, da rudarji prodajo novo ustvarjene kovance in si tako zagotovijo pokrivanje stroškov. Če si pogledamo profitabilnost rudarjenja bitcoina, lahko ugotovimo, da se je do leta 2022 razmerje med ceno rudarjenja in ceno bitcoina gibalo med 0,5–0,9, kar zagotavlja

pokrivanje operativnih stroškov, možnost vlaganja v nadgrajevanje opreme in izplačilo profita. Zaradi zgornjih argumentov tretjo hipotezo dolgoročno potrjujemo.

8 ZAKLJUČEK

V raziskovalni nalogi smo obravnavali področje tehnologije veriženja blokov, ki jo je dokončno razvil in objavil Satoshi Nakamoto leta 2009. Kljub relativno kratki dobi obstoja je tehnologija veriženja blokov požela veliko pozornosti, zato smo priča številnim novim, ambicioznim projektom, ki skušajo konkurirati že uveljavljenim rešitvam ali jih vsaj spodbuditi k izboljšavam. Eden takšnih projektov je gotovo predstavljen tudi v raziskovalni nalogi, saj poskuša znova vrniti kredibilnost in ugled medijem, saj tehnologija zagotavlja nespremenljivost objavljenih novic in spodbuja objavo verodostojnih novic.

Ker je razumevanje delovanja tehnologije veriženja blokov nujno potrebno za razumevanje dela, smo najprej poljudno predstavili delovanje tehnologije in se šele nato osredotočili na tehnično specifiko lastne implementacije. Pomemben del dejanske rabe tehnologije predstavlja tudi delovanje omrežja enakovrednih partnerjev, ki je razloženo v četrtem poglavju. Za konec smo se osredotočili še na predstavitev delovanja izdelane aplikacije za interakcijo z verigo blokov in analizo hipotez.

Ugotovili smo, da je izdelana veriga blokov primerna za objavo novic kot tudi varen prenos monetarne vrednosti, vendar le-ta aspekta ne zagotavljata uspeha na tržišču. Velik del uspeha določene verige blokov predstavlja dobro trženje in ponudba odlične uporabniške izkušnje, ki je naša aplikacije trenutno še ne ponuja. Prav tako smo prišli do spoznanja, da uporabljena kriptografija eliptičnih krivulj ne zagotavlja varnosti pred napadi kvantnih računalnikov, zato bi bila nadgradnja omrežja na alternativne asimetrične algoritme v naslednjih desetletjih nujna. Nenazadnje smo raziskali tudi trajnostni aspekt samostojnega obstaja verige, ki smo ga potrdili, saj menimo, da bo sistem nagrajevanja rudarjev v omrežju le-tem pokril vsaj del operativnih stroškov.

Tekom izdelave projekta smo se srečevali s številnimi ovirami in izzivi, ki smo jih po našem mnenju uspešno razrešili. Menimo, da končen izdelek zagotavlja vso osnovno in pričakovano funkcionalnost brez (znanih) varnostnih kompromisov, vendar so izboljšave vsekakor mogoče. Pri izdelavi nam je v veliki meri bilo v pomoč lanskoletno znanje s področja asimetrične kriptografije.

9 VIRI IN LITERATURA

[1] M. Zabukovnik, A. Vaishnav in L. Stokavnik, "Kriptografija z eliptičnimi krivuljami," marec 2023. [Elektronski vir]. Dostopno na: https://www.knjiznica-celje.si/raziskovalne/4202306879.pdf. [Poskus dostopa 20. oktober 2023].

[2] R. Pavlović, „Zagotavljanje transparentnosti oskrbovalne verige s tehnologijo veriženja podatkovnih blokov," 2018. [Elektronski vir]. Dostopno na: https://repozitorij.uni-lj.si/Dokument.php?id=105022. [Poskus dostopa 22. november 2023].

[3] S. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Elektronski vir]. Dostopno na: https://bitcoin.org/bitcoin.pdf. [Poskus dostopa 21. november 2023].

[4] W. Xu, Z. Xuan, N. Wei in L. Ren Ping, „Survey on Blockchain for Internet of Things," januar 2019. [Elektronski vir]. Dostopno na: https://www.researchgate.net/publication/330351295_Survey_on_Blockchain_for_Internet_of_Things. [Poskus dostopa 16. december 2023].

[5] Wikipedia, „Peer to peer network," 14. marec 2022. [Elektronski vir]. Dostopno na: https://en.wikipedia.org/wiki/Peer-to-peer#/media/File:P2P_network.svg. [Poskus dostopa 27. februar 2024].

[6] Ring Central, „Packet," 18. avgust 2021. [Elektronski vir]. Dostopno na: https://www.ringcentral.com/gb/en/blog/wp-content/uploads/2021/08/Packet-switching.png.webp. [Poskus dostopa 27. februar 2024].

[7] A. Maher, „Writing High-Quality C# Code: Key Principles and Essential Tools," 2. marec 2023. [Elektronski vir]. Dostopno na: https://maherz.medium.com/writing-high-quality-c-code-key-principles-and-essential-tools-47dafacabc28. [Poskus dostopa 27. februar 2024].

[8] J. D., „Logo HTML5, CSS, JavaScript," 3. maj 2021. [Elektronski vir]. Dostopno na: https://www.freepnglogos.com/images/javascript-39422.html. [Poskus dostopa 28. februar 2024].

[9] AllVectorLogo, „GitHub Logo,“ 30. december 2021. [Elektronski vir]. Dostopno na: https://allvectorlogo.com/github-logo/. [Poskus dostopa 27. februar 2024].

[10] A. Jurišić, „Tečaj iz kriptografije in računalniške varnosti,“ 2008. [Elektronski vir]. Dostopno na: http://lkrv.fri.uni-lj.si/~ajurisic/kirv08/folije/f08.pdf. [Poskus dostopa 15. november 2023].

[11] „Block,“ 13. maj 2019. [Elektronski vir]. Dostopno na: https://en.bitcoin.it/wiki/Block. [Poskus dostopa 5. december 2023].

[12] S. Lin, „Proof of work vs. proof of stake in cryptocurrency,“ [Elektronski vir]. Dostopno: https://www.researchgate.net/publication/369870684_Proof_of_Work_vs_Proof_of_Stake_in_Cryptocurrency. [Poskus dostopa 5. december 2023].

[13] J. Groopman, „Top blockchain attacks, hacks and security issues explained,“ Techtarget, 7. junij 2023. [Elektronski vir]. Dostopno na: https://www.techtarget.com/searchsecurity/tip/Top-blockchain-security-attacks-hacks-and-issues. [Poskus dostopa 28. december 2023].

[14] „Privacy,“ BitcoinWiki, 2022 september 26. [Elektronski vir]. Dostopno na: https://en.bitcoin.it/wiki/Privacy. [Poskus dostopa 29. december 2023].

[15] „Weaknesses,“ BitcoinWiki, 30. oktober 2023. [Elektronski vir]. Dostopno na: https://en.bitcoin.it/wiki/Weaknesses. [Poskus dostopa 28. december 2023].

[16] Wikipedia, „Transmission Control Protocol,“ 20. februar 2024. [Elektronski vir]. Dostopno na: https://en.wikipedia.org/wiki/Transmission_Control_Protocol. [Poskus dostopa 25. februar 2024].

[17] R. Sarode, M. Poudel, S. Shrestha in S. Bhalla, „Blockchain for committing peer-to-peer transactions using distributed ledger technologies,“ 1. januar 2021. [Elektronski vir]. Dostopno na: https://www.researchgate.net/publication/352455679_Blockchain_for_committing_p

eer-to-peer_transactions_using_distributed_ledger_technologies. [Poskus dostopa 28. december 2023].