ELEKTROTEHNIŠKI VESTNIK 86(3): 130-137, 2019
IZVIRNI ZNANSTVENI ČLANEK

Analitika blokovnih verig in iskanje povezanih transakcij

Franc Drobnič, Urban Sedlar, Andrej Kos, Matevž Pustišek

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: franc.drobnic@fe.uni-lj.si

Povzetek. Blokovne verige v zadnjem desetletju zbujajo čedalje večje zanimanje splošne in tudi znanstvene javnosti predvsem po zaslugi kriptovalut, še zlasti prve med njimi, Bitcoina. Blokovne verige v izvedbi, ki so jo razvijalci namenili za kriptovalute, sicer niso prilagojene shranjevanju večjih količin podatkov, na njihovi podlagi pa poteka živahen razvoj rešitev, ki bi bile bolj primerne za ta namen. Blokovne verige se širijo v dveh smereh. Nekatere omogočajo shranjevanje podatkov tako, da so ti dostopni samo tistemu uporabniku, ki jih je shranil. Velika pozornost pa je namenjena tehnologijam, ki bi omogočile vsaj delno dostopnost teh podatkov tudi upravljavcem shramb v oblaku za agregirane analize, seveda pod pogojem, da uporabniki ohranijo zasebnost in po možnosti nadzor nad tem, kdo in kdaj sme uporabljati njihove podatke.

Poleg analize podatkov, ki bi bili dodatno shranjeni v povezavi z blokovnimi verigami, je danes aktualna znanstvena tema tudi analiza podatkov same blokovne verige. Ker pa blokovne verige nimajo vgrajenih orodij za analizo, je treba uporabiti zunanja orodja. Za analizo povezanih udeležencev v transakcijah smo izbrali grafovsko podatkovno zbirko, podatke iz blokovne verige Ethereum prenesli v dva podatkovna modela in na njih odkrivali povezanost z iskanjem obhodov v grafu.

Ključne besede: blokovne verige, velepodatki, analitika, teorija grafov, Neo4j

Big Data Analytics in Blockchains and Search for Associated Transactions

Blockchains are gaining a significant interest in general public as well as in scientific community, especially after the successful implementation of cryptocurrencies, particularly Bitcoin as the first of them. As the blockchains in the form implemented for the cryptocurrencies are not suitable for storing large quantities of data, there is a vivid development of solutions for this purpose going on. Blockchains are extended in two ways. Some extensions allow access only to the user that has stored the data originally. On the other hand, much attention is paid to technologies that would allow the cloud-storage vendors to perform an aggregated analysis, of course with a due respect for privacy of data owners and possibly enabling their control of the data usage, i.e. those who are permitted to use their data when such use is allowed.

Analysis of the basic data constituting a blockchain is of a considerable scientific interest as well. Since a blockchain does not provide an analytic tool itself, it is necessary to use an external one. In order to find closely related transaction participants, we chose a graph database, ingested the Ethereum blockchain data into two different data models and found such communities by searching for tours in the graph.

Keywords: blockchain, big data, analytics, graph theory, Neo4j

1 UVOD

Razvoj elektronskih in internetnih storitev omogoča vedno nove storitve, ki nam lajšajo vsakdanja opravila in ponujajo nove načine izvajanja teh opravil. Klasični vzorec internetne komunikacije je, da je mogoče pošiljati podatke od koderkoli in kamorkoli v omrežju. Stranski učinek teh možnosti je, da se poslana vsebina lahko poljubno razmnoži in avtor od trenutka pošiljanja naprej nima več nadzora nad razmnoževanjem vsebine.

Raziskovalci iščejo rešitev te težave na področju kriptografije, znotraj tega pa trenutno največ obljubljajo blokovne verige. Razvoj teorije in izvedbe blokovnih verig poteka že več desetletij, v praksi pa so se razširile šele z uvedbo verige Bitcoin [1].

Blokovne verige ponujajo funkcionalnosti, ki so najbolje prilagojene mehanizmom delovanja kriptovalut, torej obravnavajo finančne transakcije. Če bi želeli njihove lastnosti porabiti za drugačne vrste vsebin, bi jih bilo treba nadgraditi. Razvoj takšnih nadgradenj že poteka, kar bo predstavljeno kasneje.

V tem delu bomo pregledali nabor rešitev, ki obljubljajo možnosti za shranjevanje velikih količin podatkov v sisteme na podlagi blokovnih verig in tudi možnosti, ki jih ponujajo za analizo tako shranjenih podatkov, kot jo poznamo na splošno na področju velepodatkov (angl. Big Data).

V drugem razdelku bomo predstavili obe področji, ki se stikata na tej temi: značilnosti blokovnih verig in možnosti uporabe metod s področja analitike velepodatkov. V tretjem razdelku bodo predstavljeni sistemi, ki jih razvijajo na področju shranjevanja podatkov v navezavi z blokovnimi verigami. Dva možna načina analize blokovnih verig in pridruženih shranjenih

Prejet 23. julij, 2018
Odobren 7. februar, 2019

podatkov bosta podana v četrtem razdelku. V petem razdelku pa bomo predstavili praktičen primer analize podatkov v blokovni verigi Ethereum z uporabo velepodatkovnih tehnologij, natančneje z uporabo grafovske podatkovne zbirke.

2 PREGLED OBRAVNAVANIH PODROČIJ

2.1 Blokovne verige

Blokovne verige (angl. Blockchain) imajo več pomembnih značilnosti. Naslovi udeležencev in podpisi transakcij so zasnovani po sistemu javnih in zasebnih ključev, s čimer je zagotovljena anonimnost (psevdonimnost) uporabnikov. Več transakcij je združenih v bloke, ki so med seboj povezani tako, da naslednji blok v verigi vsebuje tudi zgoščeno vrednost prejšnjega bloka in s tem je zagotovljena nespremenljivost prejšnjih blokov. Blokovna veriga je shranjena na več računalnikih v omrežju (vozliščih) tako, da vsak od teh vsebuje popolno kopijo vse verige. Taki računalniki so poimenovani popolno vozlišče (angl. Full Node). Tako lahko vsako popolno vozlišče preveri veljavnost vse verige. Z razpršenostjo kopij verige je zagotovljena varnost pred izgubo podatkov ob odpovedi posameznih vozlišč. Nekatere blokovne verige omogočajo, da lahko del računalnikov shranjuje samo novejše dele verig, kar jim omogoča samo izvajanje transakcij, ne pa potrjevanja blokov. Vsak nov blok je treba po dodajanju potrditi, pri čemer lahko sodeluje vsako popolno vozlišče, kar doseže z izvajanjem »dokaza o opravljenem delu« (angl. Proof of Work – PoW) ali t.i. »rudarjenjem« (angl. Mining). Cilj tega postopka je najti naključno enkratno polnilo (angl. »nonce«), ki ustreza pravilu, da ga je težko najti, zelo preprosto pa je preveriti njegovo ustreznost. Pri potrjevanju bloka več popolnih vozlišč tekmuje, katero bo v krajšem času našlo pravo vrednost polnila. S tem je zagotovljena neodvisnost verige od osrednjega potrjevalca. Sodelovanje pri takem iskanju spodbuja pravilo, da vozlišče, ki najhitreje najde polnilo, dobi nagrado v obliki določenega zneska v isti kriptovaluti.

Lastnosti blokovnih verig, kot so anonimnost, nespremenljivost, varnost pred izgubo podatkov in neodvisnost od osrednjega skrbnika, so zanimive tudi za uporabo na drugih področjih, ne samo pri kriptovalutah. Zaslužki, ki se obetajo sodelujočim pri uporabi in razvoju na področju kriptovalut, spodbujajo razvoj tehnologije, posledično pa tudi zanimanje za razvoj uporabe blokovnih verig na drugih področjih.

Lastnosti, da je podatkovna shramba razmnožena[1] po vozliščih v omrežju in da večje število transakcij potrjujejo hkrati, sta omejujoči, če bi želeli z blokovnimi verigami izvajati plačilni promet v obsegu, v kakršnem ga izvajajo veliki plačilni sistemi (npr. Visa), kjer sta število transakcij in njihova frekvenca bistveno večja kot pri današnjih blokovnih verigah. Izboljšava načina shranjevanja verige s prijemi, ki so znani iz velepodatkovnih rešitev, je predlagana npr. v [2].

Blokovne verige nastopajo v dveh vrstah izvedbe: javne, brez upravljanja s pravicami (angl. Permissionless), kamor spadajo trenutno najbolj razširjene kriptovalute (npr. Bitcoin, Ethereum, Ripple, Litecoin, …), in zasebne, z dodanim upravljanjem pravic (angl. Permissioned), kjer vozlišča v omrežju niso enakovredna v smislu, da za potrjevanje veljavnosti blokov skrbi določena podmnožica vozlišč.

2.2 Velepodatki

Velepodatki (angl. Big Data) so področje računalništva (ali širše informacijsko-komunikacijskih tehnologij), kjer je predmet obdelave tako velika količina podatkov, da jih ni mogoče shraniti na posamezen računalnik. Za obdelave velepodatkov se uporabljajo velike gruče (angl. Cluster) računalnikov, ki so med seboj povezani z zmogljivimi omrežnimi povezavami. Take konfiguracije računalnikov se uporabljajo tudi v oblaku (angl. Cloud), kjer je več gruč razmeščenih po različnih geografskih lokacijah in s tem omogočajo shranjevanje poljubnih podatkov zelo velikemu številu uporabnikov.

Shranjevanje podatkov v gruči ali v oblaku pa ima nekaj pomanjkljivosti. Glavna ovira pri večjem razmahu uporabe javnih oblakov je nezaupanje uporabnikov do ponudnikov oblaka. To nezaupanje sicer ni tako, da bo upravitelj oblaka shranjene podatke zlorabil ali uničil, ampak da lahko do podatkov neupravičeno pride zaradi radovednosti. Rešitev, da uporabniki kriptirajo vsebino, preden jo shranijo v oblak, je sicer za te uporabnike zadovoljiva, ne omogoča pa velepodatkovne analize teh podatkov, kar bi bil lahko dodaten vir dohodka za ponudnike storitve shranjevanja v oblaku in s tem zniževanja stroškov za uporabnike.

Novejši predlogi za izboljšanje te funkcionalnosti gredo v smeri enkripcijskih shem, ki bi vsaj deloma omogočale pridobivanje podatkov iz kriptiranih vsebin, ob obveznem varovanju zasebnosti lastnikov teh vsebin. Predlagane enkripcijske sheme so: diferencialna zasebnost (angl. Differential Privacy) [3], homomorfna enkripcija [4] in iskalna simetrična enkripcija (angl. Searchable Symmetric Encryption – SSE) [5]. Nastala je tudi relacijska podatkovna zbirka CryptDB [6], ki omogoča izvajanje nekaterih relacijskih operacij v jeziku SQL nad kriptiranimi podatki, ne da bi bili nekriptirani podatki razkriti strežniku, kjer so shranjeni. Pri uporabi sistemov, ki omogočajo operacije nad kriptiranimi podatki brez razkrivanja izvirnih podatkov, pa so mogoči napadi s kriptoanalizo, zato je treba te sisteme uporabljati zelo previdno, na kar so avtorji sistema CryptDB opozorili v [7].

Poleg varovanja zasebnosti je v nekaterih primerih uporabe zaželena lastnost tudi nespremenljivost shranjenih podatkov. Za ta namen bi lahko uporabili

[1] Podatkovna zbirka kriptovalut ni porazdeljena (distribuirana) v pravem pomenu besede, ker vsa vozlišča v omrežju vsebujejo enako kopijo celotne zbirke. Porazdeljeno je samo pridobivanje konsenza o gradnji verige.

blokovne verige, ki pa bi jih bilo treba razširiti tako, da bi bilo mogoče z njimi zaščititi večje količine podatkov. Blokovna veriga Ethereum [8], ki je nastala po vzoru verige Bitcoin, omogoča različne razširitve, kar ji omogoča dodatna zmožnost »pametnih pogodb« (angl. Smart Contract). Trenutne izvedbe blokovnih verig, ki so zasnovane samo z upoštevanjem zahtev kriptovalut, pa izkazujejo omejitve pri frekvenci potrjevanja transakcij in količini shranjenih podatkov, kar so razčlenili v [9] in pokazali, kateri deli zasnove sistemov z blokovnimi verigami lahko povzročijo te težave in jih bo treba v prihodnosti izboljšati.

Pomembno področje, ki je primerno za uporabo blokovnih verig, so informacijski sistemi v zdravstvu. Tu so blokovne verige zanimive predvsem zaradi velikih zahtev po varovanju podatkov o boleznih in zdravljenju [10], kjer naj bo bolnik tisti, ki odloča o dostopnosti podatkov vsem drugim udeležencem v procesu zdravljenja. Tako je nastalo nekaj pobud: [11], [12], kjer z uporabo blokovne verige z upravljanjem pravic omogočajo bolniku, da učinkovito dostopa do vseh svojih podatkov, ne glede na to, v kateri zdravstveni ustanovi so ti podatki nastali, poleg tega pa tudi kadarkoli določi, kdo sme vpogledovati v te podatke ali jih spreminjati. Raziskovanje je spodbudilo tudi ameriško ministrstvo, pristojno za zdravje [13].

Lastnosti blokovnih verig bi bile lahko koristne tudi pri elektronski izvedbi glasovanj [14], vendar pri tem ostajajo nerešena številna pomembna vprašanja.

3 SHRANJEVANJE VELIKIH KOLIČIN PODATKOV IN BLOKOVNE VERIGE

Na podlagi verige Ethereum je nastalo več predlogov rešitev za shranjevanje velikih količin podatkov. Skoraj vse delujejo na omrežju enakopravnih vozlišč (angl. Peer-to-Peer, P2P), kar pomeni, da nobeno vozlišče nima drugačne vloge kot preostala. Z uporabo porazdeljene zgoščene tabele (angl. Distributed Hash Table – DHT), ki omogoča preverljivo nespremenljivost vsebin, v prihodnosti pa tudi jamstvo dosegljivosti, in omrežne topologije Kademlia so nastale decentralizirane datotečne shrambe. Swarm [15] je del osnovnega sklada programske opreme verige Ethereum. Storj [16] namesto dokaza o opravljenem delu uporablja dokazovanje zmožnosti pridobivanja shranjene vsebine (angl. Proof of Retrievability). Analitika je pri tem sistemu mogoča samo na javno objavljenih vsebinah (ki so zaščitene pred spreminjanjem), ne pa na zasebnih, ker so kriptirane po sistemu javnih ključev (PKI). IPFS [17] uporablja ideje iz uspešnih v praksi izvedenih sistemov, kot so BitTorrent (izmenjava datotek), Git (vodenje različic datotek), Self-Certifying File Systems (porazdeljeni globalni imenski prostor) in drugi. IPFS kot plačilno sredstvo lahko uporablja Filecoin [18]. To je kriptovaluta, pri kateri za potrjevanje blokov tekmujejo ponudniki (rudarji) s količino prostora, ki ga namenjajo shranjevanju podatkov. Avtorji sistemov Swarm in IPFS priznavajo, da sta si po zmožnostih in načinu uporabe zelo podobna, čeprav so tehnični vidiki izvedbe različni. UStore [19] je shramba, ki za zdaj deluje samo v delovnem pomnilniku.

Enigma [20] na drugi strani omogoča izvajanje obdelav nad shranjenimi podatki ob ohranjanju tajnosti teh podatkov. Podatke obdeluje samo del vozlišč (izbranih naključno), kar omogoča večjo razširljivost omrežja. Ta vozlišča obdelujejo samo del podatkov z uporabo homomorfne enkripcije, tako da nikakor ne morejo priti do celotnih podatkov. MaidSafe oz. SAFE Network [21] ponuja storitve shranjevanja koščkov datotek (in v prihodnosti tudi izvajanja postopkov), za plačevanje teh storitev ima predvideno kriptovaluto Safecoin. Sia [22] uporablja za shranjevanje blokovno verigo, na katero shranjuje pametne pogodbe, s katerimi zagotavlja plačevanje za uporabo shrambe. Pametne pogodbe omogočajo večjo frekvenco operacij shranjevanja, saj tako v blokovni verigi ni treba potrditi vsake posamezne datotečne operacije, ampak samo pametno pogodbo. Plačevanje poteka v kriptovaluti Siacoin. Deli datotek so kriptirani po algoritmu Twofish in med vozlišči razpršeni po algoritmu Reed-Solomon.

Holochain [23] gradi, podobno kot Storj, na DHT in omogoča deljenje računalniških virov, za to pa ponudnik dobi plačilo v posebni kriptovaluti. Vsak sodelujoči ima pri sebi ločeno verigo podatkov, na skupno verigo pa shrani samo nekatere podatke, ki smejo biti javni. Tako je omogočena zasebnost nekaterih podatkov. Drugače, kot pri drugih blokovnih verigah, je zasnovan konsenz, ki ni globalen, ampak se zgradi samo med udeleženci posameznega posla.

iEx.ec [24] je sistem za izmenjavo računalniških procesnih virov, ki omogoča preverljivo (z Merkle-ovim preverjanjem) izvajanje naročenih obdelav na šibko povezanih računalnikih (tudi npr. na neizkoriščenih virih na domačih računalnikih) in vključuje sistem za plačevanje teh storitev na podlagi blokovnih verig; omogoča tudi velepodatkovne pristope (prilagojen sistem MapReduce) in pri tem podpira zelo veliko udeleženih računalnikov.

IOTA [25] namesto blokovne verige uporablja usmerjeni aciklični graf (angl. Directed Acyclic Graph – DAG), potrjevanje transakcij je nekoliko drugačno. Namesto potrjevanja blokov z več transakcijami tu vsaka posamezna transakcija potrdi vsaj dve drugi transakciji. Za potrjevanje ni nagrade, zasnovana pa je na splošno ob predpostavki uporabe v IoT, kjer so vozlišča omejeno računsko zmogljiva.

BigchainDB [26] je zasnovan v nasprotni smeri, kot druge rešitve: osnova je porazdeljena velepodatkovna baza (MongoDB ali RethinkDB), ki so jo nadgradili z lastnostmi blokovnih verig. Vozlišča tu niso enakovredna, potrjevanje transakcij in dodeljevanje pravic je v rokah manjšega števila posebej izbranih zaupanja vrednih vozlišč. Poizvedovanje omogoča že uporabljena baza, nadgradnja pa zagotavlja sistem pravic, kjer avtor vsebine sme to vsebino vstaviti v bazo, drugi pa smejo dostopati do te vsebine samo, če jim je

bila dodeljena taka pravica, kar je zagotovljeno s kriptografskimi sredstvi.

4 ANALIZA PODATKOV V BLOKOVNIH VERIGAH

Analiza podatkov je glede na vrsto podatkov, ki jih analiziramo, lahko taka, da analiziramo:

- podatke, ki so pridruženi blokovni verigi in so shranjeni na drugih mestih, blokovna veriga pa jim zagotavlja določene lastnosti;
- podatke znotraj blokovne verige (naslove, zneske, časovne žige ipd.).

Postopke pod prvo alinejo bi lahko poimenovali »zunanja«, pod drugo pa »notranja« analiza.

4.1 Zunanja analiza blokovne verige

Podatke znotraj blokovne verige lahko analiziramo v povezavi z zunanjimi podatki, ki so po svojem izvoru lahko povezani z obravnavano blokovno verigo (npr. imena nekaterih lastnikov računov, ki so objavljena na spletnih portalih) ali pa nimajo take povezave (npr. podatki o dogodkih v družbi ali gospodarstvu). Primer orodja za tako analizo sta [27] in [28].

Primer novejšega predloga za izvedbo iskanja po zunanjih podatkih, namenoma shranjenih pridruženo k blokovnim verigam, je SSE-using-BC avtorjev Li et al. [29].

4.2 Notranja analiza blokovne verige

Ker so blokovne verige zelo dolg seznam transakcij, torej prenosov določenih zneskov z enega naslova na drugega, je najbolj naraven shematski zapis zanje usmerjeni graf [27] [30] [31]. V grobem oba naslova pomenita vozlišči, transakcija med njima pa pomeni usmerjeno povezavo med tema vozliščema. Podrobnejše semantično modeliranje blokovnih verig je predstavljeno v [32].

Raziskovalci so razvili veliko metod, ki z uporabo analize grafov iz blokovne verige pridobijo uporabne informacije, kot so iskanje nenavadnih transakcij, velikokrat povezanih z nezakonitimi dejavnostmi, napovedovanje rasti blokovne verige v prihodnosti ipd. Primer uporabe take metode je iskanje dejavnosti posameznega uporabnika. Čeprav je blokovna veriga anonimna, z različnimi metodami odkrivanja skupnosti (angl. Community Detection) lahko sorazmerno uspešno najdemo nabor transakcij, ki pripadajo določenemu uporabniku (reidentifikacija).

5 NOTRANJA ANALIZA BLOKOVNE VERIGE ETHEREUM ZA ISKANJE POVEZANIH TRANSAKCIJ

Motivacija za naše delo je v tem, da se večina raziskav blokovnih verig namenja verigi Bitcoin. Ker so orodja za k blokovnim verigam pridruženo shranjevanje velikih količin podatkov še v razvojni fazi, smo se usmerili v »notranjo« analizo podatkov blokovnih verig in znotraj tega posebej v blokovno verigo Ethereum. V [32] je sicer podan splošen predlog predstavitve blokovne verige Ethereum z grafom, ni pa podrobno predstavljena konkretna izvedba takega grafa. Poleg predstavitve blokovne verige z grafom smo želeli tudi po vzoru analize povezanih naslovov, ki je spet bolj na primeru verige Bitcoin predstavljena v [30], narediti podobno analizo povezanih naslovov še za verigo Ethereum.

Pomembna razlika med verigo Bitcoin in verigo Ethereum je ta, da pri prvi v isti transakciji sodeluje več izvornih in več ciljnih naslovov (angl. Multiple Input Multiple Output – MIMO). S tem, da ti naslovi sodelujejo v isti transakciji, lahko dokaj zanesljivo sklepamo na njihovo povezanost tudi zunaj blokovne verige. Pri verigi Ethereum je ta povezava šibkejša, ker se tu transakcije izvajajo vedno z enega izvornega na en ciljni naslov (angl. Single Input Single Output – SISO). Na njihovo povezanost bi lahko sklepali zgolj iz nastopanja v skupini več medsebojno povezanih transakcij. V tem delu smo se omejili na iskanje obhodov (vase zaključenih sprehodov [33]) v grafu in to v kratkem časovnem intervalu.

Blokovne verige same ne ponujajo zmogljivih orodij za njihovo analizo, ponujajo pa programske vmesnike (API), s katerimi lahko iz njih pridobimo podatke in jih potem obdelamo z drugimi orodji. Tako smo iz blokovne verige Ethereum z uporabo vmesnika Web3 [34] pridobili naslednje podatke o transakcijah:

- naslov pošiljatelja,
- naslov prejemnika,
- zgoščeno vrednost transakcije (angl. Hash),
- vrednost transakcije,
- čas nastanka,
- številko bloka in zaporedno številko transakcije v bloku.

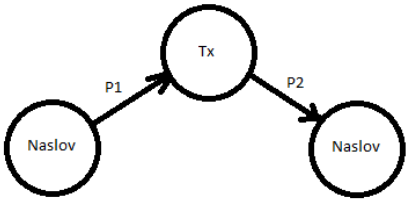
Omejili smo se samo na transakcije med t.i. »zunanjimi« naslovi, to so naslovi, ki pripadajo uporabnikom. Poleg teh veriga vsebuje še naslove, ki pripadajo pametnim pogodbam, ki pa jih nismo pridobili. Tako pridobljene podatke smo prenesli v grafovsko podatkovno zbirko Neo4j [35], ki je zmožna tudi obdelave velepodatkov, če je nameščena na gruči računalnikov. Obhode bi z drugimi orodji (npr. relacijskimi podatkovnimi zbirkami) težje odkrili. Nekatere relacijske podatkovne zbirke nimajo posebnih orodij za iskanje hierarhičnih odnosov in jih je treba simulirati z večkratnimi stiki, kar omejuje globino iskanja. Druge (npr. Oracle) imajo tako orodje (`START WITH … CONNECT BY`), ki pa postane z večjo globino iskanja manj učinkovito.

V grafovski podatkovni zbirki sta osnovna gradnika vozlišče in povezava. Vozlišče ima lahko uporabniško določeno oznako (angl. Label), povezava pa tip (angl. Type). Dodatno lahko shranjujeta tudi pridružene podatke, ki jim rečemo lastnosti (angl. Properties). Omejitev pri tem je, da je mogoče indeksiranje za hitrejše iskanje samo na lastnostih, ki so pridružene vozliščem. Zato smo opustili model, pri katerem bi vozlišča pomenila naslove, povezave pa transakcije, in smo naredili dva drugačna modela. Pri prvem smo kot

vozlišča določili naslove in transakcije, kot povezave pa dejanske povezave od izvornega naslova do transakcije in na drugi strani od transakcije do ciljnega naslova. Drugi model pa smo zastavili kot model prehajanja stanj, kjer vozlišča ponazarjajo dogodke prejema zneska na ciljni naslov, povezave pa transakcije.

5.1 Model z naslovi in transakcijami kot vozlišči

Vrednosti naslovov smo vpisali kot lastnost vozlišč naslovov, podatke o transakciji smo vpisali kot lastnosti vozlišč transakcij, povezave pa niso dobile dodatnih lastnosti. V uporabljeni podatkovni zbirki smo pri tem modelu vozlišča naslovov označili z oznako `Naslov`, vozlišča transakcij z oznako `Tx`, usmerjene povezave pa z oznakama `P1` in `P2`. Povezavi sta usmerjeni v smeri poteka transakcije. Model posamezne transakcije prikazuje slika 1.



Slika 1: Model posamezne transakcije

Posamezno transakcijo bi lahko zapisali v sintaksi deklarativnega poizvedovalnega jezika Cypher [36], ki ga uporablja izbrana podatkovna zbirka, takole:

```
(:Naslov {naslov: …})
  -[:P1]->(:Tx {zgostitev: …,
                vrednost: …,
                datum: …,
                blok: …,
                zap_stev: …})
  -[:P2]->(:Naslov {naslov: …})
```

V okroglih oklepajih so zapisana vozlišča, v oglatih pa povezave. Oznake vozlišč in tipi povezav se začnejo z dvopičjem, v zavitih oklepajih pa so navedene lastnosti objektov v sintaksi JSON. Znak minus ponazarja potek povezav, usmerjenost povezav pa določata znaka > ali <. Prelomi vrstic niso potrebni, tu so narejeni zaradi boljše preglednosti.

Izdelali smo tri indekse: na lastnostih `naslov`, `zgostitev` in `blok`. Prva dva sta enolična, tretji pa navaden.

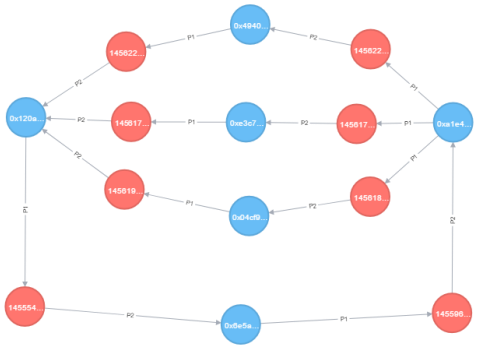
Po zgledu analize v [27] [30] [32] smo izvedli poizvedbe, ki bi lahko dale dodaten vpogled v dogajanje v blokovni verigi. Zanimivi so obhodi (sprehodi v grafu, kjer sta izhodišče in cilj isto vozlišče). Primer poizvedbe v jeziku Cypher, ki vrne take obhode:

```
MATCH     p=((n:Naslov)-[*1..5]->(t:Tx)-[*1..5]-
>(n:Naslov))
RETURN *
LIMIT 9
```

Omejitev, ki zahteva sklenjen krog povezav, je v tem, da sta na začetku in na koncu zahtevanega vzorca enaka vzdevka vozlišč (`n:Naslov`). Med vozlišči zahtevamo tudi vsaj eno in največ pet povezav (`*1..5`), in to na vsaki strani transakcije. S tem se izognemo zankam, kjer transakcija poteka v enem koraku iz istega naslova vase samega. Te preproste transakcije bi lahko hitro odkrili tudi z drugačnimi orodji, npr. relacijsko podatkovno zbirko. Omejitev LIMIT je namenjena hitrejši izvedbi poskusne poizvedbe, kjer smo želeli prikazati samo en obhod. Brez te omejitve bi preiskovanje celotne podatkovne zbirke trajalo dlje časa.

Grafično prikazuje del rezultata te poizvedbe slika 2.

Obhod se začne v vozlišču levo zgoraj in se nadaljuje s povezavo navzdol in naprej v nasprotni smeri urinega kazalca. Prikazane transakcije so se zgodile v blokih od 1007871 do 1047839, kar ustreza časovnemu obdobju od 15.2.2016 12:49:54 UTC do 23.2.2016 11:55:39 UTC. Med naslovoma desno zgoraj in levo zgoraj so bile kasneje izvedene še nekatere druge transakcije, ker pa niso ključne za tvorbo obhoda, smo jih zaradi preglednosti izpustili.

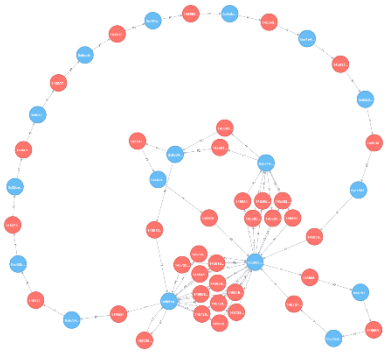


Slika 2: Primer obhoda transakcij

Vsi primeri obhodov v tem grafu pa ne kažejo povezanega toka vrednosti. Primer obhoda, ki ga kaže slika 3, vrne poizvedba:

```
MATCH     p=((n:Naslov)-[*5..]->(t:Tx)-[*5..]-
>(n:Naslov))
RETURN *
LIMIT 1
```

Številke blokov, v katerih so nastale posamezne transakcije, so za veliki lok, ki teče od leve spodaj čez zgornji del slike na desno: 1284052, 1577779, 1551452, 51913, 55500, … Ta obhod torej kaže neko drugačno, naključno povezavo med računi.

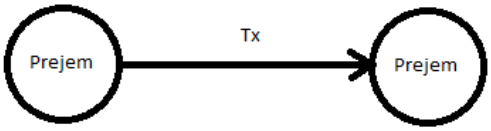


Slika 3: Primer vsebinsko nepovezanega obhoda

Poizvedba v jeziku Cypher, ki bi izločila takšne obhode, bi bila dokaj zahtevna, zato je bolje to obdelati v aplikaciji na proceduralni način. Zato smo razvili drugačen model, ki bo predstavljen v nadaljevanju.

5.2 Model s prehajanjem stanj

Drugi model smo zastavili tako, da so vozlišča označena kot `Prejem` in imajo sestavljeni enolični ključ, ki vsebuje ciljni naslov, številko bloka in zaporedno številko transakcije v bloku. Dodatna lastnost vozlišča je izvorni naslov, s katerega je prispel znesek. Povezave so tipa `Tx` in so enolično označene z zgoščeno vrednostjo transakcije, kot dodatne lastnosti pa vsebujejo vrednost transakcije in datum. Izhajajo iz zadnjega vozlišča (vzamemo največjo številko bloka), na katerem je nastopal izvorni naslov. Vozlišča, ki so na začetku takih zaporedij transakcij, smo dodali umetno in jim dali enako oznako kot preostalim, za ciljni naslov dali dejanski izvorni naslov transakcije, številko bloka in transakcije pa smo napolnili z nič.

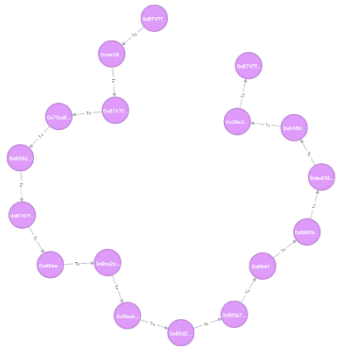


Slika 4: Model transakcije s prehajanjem stanj

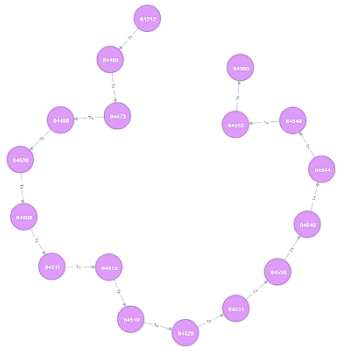
Tako modeliran graf pokaže zaporedne prehode vrednosti med naslovi, kot so se izvajali. Slaba stran tega modela pa je, da vsak naslov nastopa v grafu večkrat in je treba za iskanje naslovov vedno ustrezno omejiti želeno območje številk bloka. Obenem dejanski obhodi v smislu prvega modela v tem modelu niso grafično prikazani krožno, ampak v obliki drevesa, kjer korensko vozlišče in vsi listi pripadajo istemu naslovu.

Primer obhoda med računi, kjer pa zaključenost sprehoda vase na prvi pogled ni vidna, kaže slika 5 in jo vrne poizvedba v jeziku Cypher:

```
MATCH p=((n:Prejem)-[*8..]->(m:Prejem))
 WHERE n.cilj = m.cilj
AND m.blok < 700000
RETURN *
LIMIT 1
```



Slika 5: Primer obhodov v drugem modelu



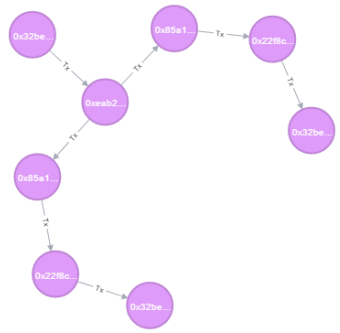
Slika 6: Obhodi v drugem modelu s številkami blokov

Obhod se začne pri vozlišču na vrhu levo in poteka v smeri, nasprotni urnemu kazalcu. Oznaka naslova iz prvega vozlišča se ponovi še na koncu prikazane verige transakcij, pa tudi vmes (na 2. in 5. mestu), kar pomeni, da je ta naslov večkrat sodeloval v tem zaporedju transakcij. To bi v smislu prvega modela pomenilo več krogov z istim izhodiščnim in ciljnim naslovom.

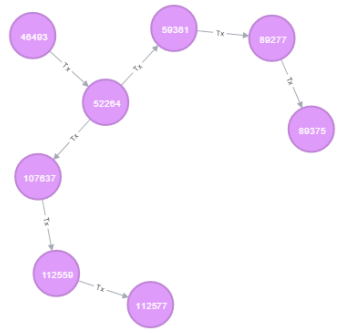
Tu smo že z modelom zagotovili, da si številke blokov sledijo v smeri povezav naraščajoče od 64242 do 64560, kot kaže slika 6.

Poizvedbe, ki iščejo obhode, v tem modelu najdejo tudi vzporedne strukture. Primer prikazujeta slika 7 in slika 8 kot rezultat podobne poizvedbe, kot zgoraj, le z drugačnimi omejitvami območja številk blokov. Tu sta se dva podobna obhoda zgodila v različnih časovnih obdobjih, imata pa prvo transakcijo skupno.

Vrednosti transakcij, ki nastopajo v krožnih povezavah, sicer niso enake. Pri obeh opisanih modelih predstavitve z grafom pa iz obhodov vidimo mogočo medsebojno povezanost teh naslovov.



Slika 7: Vzporedna obhoda v drugem modelu



Slika 8: Vzporedna obhoda v drugem modelu s številkami blokov

Z izvajanjem poizvedb v grafovski podatkovni zbirki smo našli tudi več kot 100 korakov dolge obhode, kar bi bilo z relacijsko tehnologijo težje doseči.

Tako pridobljeni podatki lahko nakazujejo povezanost udeleženih naslovov, ker je mogoče sklepati, da v kratkem času popolni neznanci ne bi mogli izpeljati tako povezanih transakcij brez dogovora. Pri tem gre lahko za popolnoma legalne dogovore, lahko pa tudi za prikrivanje prenosa denarja ali poskuse vplivanja na ceno same kriptovalute z izvajanjem fiktivnih transakcij in podobno. Iz podatkov same blokovne verige narave teh povezav ne moremo ugotoviti, zato moramo najdene povezave preveriti še z drugimi podatki, npr. z zajemanjem svetovnega spleta.

6 Sklep

Razvoj sistemov za shranjevanje velikih količin podatkov vključuje tudi uporabo blokovnih verig za zagotavljanje posebnih lastnosti: nespremenljivosti, splošne dostopnosti podatkov in boljšega upravljanja uporabe shranjenih vsebin. Ob tem se razvijajo tudi metode za analitiko nad kriptiranimi podatki, ki bi bili shranjeni povezano z blokovnimi verigami. Trenutno pa so vsi primeri teh sistemov še v razvojni fazi in široka uporaba še ni mogoča.

Podobno kot so bile blokovne verige že analizirane, smo izvedli analizo blokovne verige Ethereum z uporabo grafovske podatkovne baze. Razvili smo dva podatkovna modela za ponazoritev dogodkov v blokovni verigi, ki omogočata analizo medsebojne povezanosti naslovov: model z naslovi in transakcijami kot vozlišči in model s prehajanjem stanj. Z njuno uporabo smo v podatkih blokovne verige Ethereum našli vase zaključene verige transakcij, iz česar lahko sklepamo na njihovo povezanost tudi zunaj blokovne verige.

Ta dva modela sta potencialno uporaben pripomoček pri raziskovanju zanimivih pojavov v blokovni verigi Ethereum, tudi v povezavi z drugače pridobljenimi podatki o teh naslovih. V prihodnosti bomo z njima nadaljevali raziskave v smeri odkrivanja skupnosti, iskanja sumljivih transakcij in podobno, za kar bomo uporabili metode razpoznavanja vzorcev in širšega področja umetne inteligence.

Literatura

[1] S. Nakamoto, „bitcoin.org,“ [Elektronski]. Available: https://bitcoin.org/bitcoin.pdf. [Poskus dostopa 13 11 2017].

[2] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta in B. Ford, „OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding,“ 2017. [Elektronski]. Available: https://eprint.iacr.org/2017/406.pdf. [Poskus dostopa 16 11 2017].

[3] C. Dwork, „Differential Privacy,“ [Elektronski]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/dwork.pdf. [Poskus dostopa 12 1 2018].

[4] W.-j. Lu, S. Kawasaki in J. Sakuma, „Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data,“ 2016. [Elektronski]. Available: https://eprint.iacr.org/2016/1163.pdf. [Poskus dostopa 15 11 2017].

[5] S. Kamara, C. Papamanthou in T. Roeder, „Dynamic Searchable Symmetric Encryption,“ 2012. [Elektronski]. Available: https://eprint.iacr.org/2012/530.pdf. [Poskus dostopa 15 11 2017].

[6] R. A. Popa, N. Zeldovich in H. Balakrishnan, „CryptDB: A Practical Encrypted Relational DBMS,“ 2011. [Elektronski]. Available: https://people.csail.mit.edu/nickolai/papers/popa-cryptdb-tr.pdf. [Poskus dostopa 15 1 2018].

[7] R. A. Popa, N. Zeldovich in H. Balakrishnan, „Guidelines for Using the CryptDB System Securely,“ 2015. [Elektronski]. Available: https://eprint.iacr.org/2015/979.pdf. [Poskus dostopa 15 1 2018].

[8] Ethereum Foundation, „Ethereum Project,“ [Elektronski]. Available: https://ethereum.org/. [Poskus dostopa 14 11 2017].

[9] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi in K.-L. Tan, „BLOCKBENCH: A Framework for Analyzing Private Blockchains,“ 12 3 2017. [Elektronski]. Available: https://arxiv.org/pdf/1703.04057.pdf. [Poskus dostopa 9 11 2017].

[10] U. Sedlar, M. Volk in J. Bešter, „Pristopi k načrtovanju in razvoju rešitev digitalnega zdravja,“ *Elektrotehniški vestnik,* Izv. 82(3), pp. 130-138, 2015.

[11] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher in F. Wang, „Secure and Trustable Electronic Medical Records Sharing using Blockchain,“ 2 8 2017. [Elektronski]. Available: https://arxiv.org/pdf/1709.06528.pdf. [Poskus dostopa 9 11 2017].

[12] A. Ekblaw, A. Azaria, J. D. Halamka in A. Lippman, „A Case Study for Blockchain in Healthcare: “MedRec” prototype for electronic health records and medical research data,“ 8 2016. [Elektronski]. Available:

https://www.healthit.gov/sites/default/files/onc_blockchainchallenge_mitwhitepaper_copyrightupdated.pdf. [Poskus dostopa 3 2 2018].

[13] U.S. Department of Health & Human Services, „ONC announces Blockchain challenge winners,“ 1 9 2016. [Elektronski]. Available: http://web.archive.org/web/20170128063822/https://www.hhs.gov/about/news/2016/08/29/onc-announces-blockchain-challenge-winners.html. [Poskus dostopa 7 11 2017].

[14] Y. Liu in Q. Wang, „An E-voting Protocol Based on Blockchain,“ 2017. [Elektronski]. Available: https://eprint.iacr.org/2017/1043.pdf. [Poskus dostopa 16 11 2017].

[15] „Swarm Documentation,“ 2016. [Elektronski]. Available: https://swarm-guide.readthedocs.io/en/latest/. [Poskus dostopa 22 12 2017].

[16] S. Wilkinson, T. Boshevski, J. Brandoff, J. Prestwich, G. Hall, P. Gerbes, P. Hutchins in C. Pollard, „Storj: A Peer-to-Peer Cloud Storage Network (whitepaper),“ 15 12 2016. [Elektronski]. Available: https://storj.io/storj.pdf. [Poskus dostopa 14 11 2017].

[17] J. Benet, „ipfs/papers,“ 1 4 2015. [Elektronski]. Available: https://github.com/ipfs/papers/blob/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf. [Poskus dostopa 14 11 2017].

[18] Protocol Labs, „filecoin.io,“ [Elektronski]. Available: https://filecoin.io/filecoin.pdf. [Poskus dostopa 20 12 2017].

[19] A. Dinh, J. Wang, S. Wang, G. Chen, W.-N. Chin, Q. Lin, B. C. Ooi, P. Ruan, K.-L. Tan, Z. Xie, H. Zhang in M. Zhang, „UStore: A Distributed StorageWith Rich Semantics,“ [Elektronski]. Available: https://arxiv.org/pdf/1702.02799.pdf. [Poskus dostopa 9 11 2017].

[20] G. Zyskind, O. Nathan in A. Pentland, „Enigma: Decentralized Computation Platform with Guaranteed Privacy,“ [Elektronski]. Available: https://www.enigma.co/enigma_full.pdf. [Poskus dostopa 18 12 2017].

[21] „MaidSafe, technical,“ MaidSafe, 19 11 2013. [Elektronski]. Available: https://blog.maidsafe.net/category/technical/. [Poskus dostopa 3 1 2018].

[22] „Sia wiki: Introduction to file contracts,“ [Elektronski]. Available: https://siawiki.tech/about/introduction_to_file_contracts. [Poskus dostopa 4 1 2018].

[23] „Holochains for Distributed Data Integrity,“ [Elektronski]. Available: http://ceptr.org/projects/holochain. [Poskus dostopa 19 12 2017].

[24] iEx.ec, „iExec,“ 18 3 2017. [Elektronski]. Available: https://iex.ec/app/uploads/2017/04/iExec-WPv2.0-English.pdf. [Poskus dostopa 14 11 2017].

[25] S. Popov, „IOTA.org,“ 1 10 2017. [Elektronski]. Available: https://iota.org/IOTA_Whitepaper.pdf. [Poskus dostopa 20 12 2017].

[26] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. T. McConaghy, G. McMullen, R. Henderson, S. Bellemare in A. Granzotto, „BigchainDB Whitepaper,“ 8 6 2016. [Elektronski]. Available: https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf. [Poskus dostopa 11 1 2018].

[27] M. Fleder, M. S. Kester in S. Pillai, „Bitcoin Transaction Graph Analysis,“ 3 1 2014. [Elektronski]. Available: https://people.csail.mit.edu/spillai/data/papers/bitcoin-transaction-graph-analysis.pdf. [Poskus dostopa 18 11 2017].

[28] M. Bartoletti, A. Bracciali, S. Lande in L. Pompianu, „A general framework for blockchain analytics,“ 6 11 2017. [Elektronski]. Available: https://arxiv.org/pdf/1707.01021.pdf. [Poskus dostopa 9 11 2017].

[29] H. Li, F. Zhang, J. He in H. Tian, „A Searchable Symmetric Encryption Scheme using BlockChain,“ 2017. [Elektronski]. Available: https://arxiv.org/pdf/1711.01030.pdf. [Poskus dostopa 9 11 2017].

[30] C. G. Akcora, Y. R. Gel in M. Kantarcioglu, „Blockchain: A Graph Primer,“ 2017. [Elektronski]. Available: https://arxiv.org/pdf/1708.08749.pdf. [Poskus dostopa 9 11 2017].

[31] R. Cazabet, R. Baccour in M. Latapy, „Tracking bitcoin users activity using community detection on a network of weak signals,“ 23 10 2017. [Elektronski]. Available: https://arxiv.org/pdf/1710.08158.pdf. [Poskus dostopa 9 11 2017].

[32] C. Cachin, A. D. Caro, P. Moreno-Sanchez, B. Tackmann in M. Vukolić, „The Transaction Graph for Modeling Blockchain Semantics,“ 2017. [Elektronski]. Available: https://eprint.iacr.org/2017/1070.pdf. [Poskus dostopa 16 11 2017].

[33] G. Fijavž, „Diskretne strukture,“ 2015. [Elektronski]. Available: http://matematika.fri.uni-lj.si/ds/ds.pdf. [Poskus dostopa 20 4 2018].

[34] Ethereum Foundation, „Web3 JavaScript app API,“ 2018. [Elektronski]. Available: https://github.com/ethereum/wiki/wiki/JavaScript-API. [Poskus dostopa 22 1 2018].

[35] Neo4j, Inc., „The Neo4j Graph Platform,“ 2018. [Elektronski]. Available: https://neo4j.com/. [Poskus dostopa 21 3 2018].

[36] Neo4j, Inc., „The Neo4j Developer Manual - Cypher,“ 2018. [Elektronski]. Available: https://neo4j.com/docs/developer-manual/current/cypher/. [Poskus dostopa 26 3 2018].

Franc Drobnič je diplomiral leta 2013 in magistriral leta 2017 na Fakulteti za elektrotehniko v Ljubljani. Zaposlen je kot raziskovalec na Fakulteti za elektrotehniko Univerze v Ljubljani. Njegova raziskovalna zanimanja vključujejo tehnologije za shranjevanje in obdelavo velikih količin podatkov.

Urban Sedlar je doktoriral leta 2010 na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je trenutno tudi zaposlen. Njegovo področje raziskovanja obsega aplikacije senzorskih in analitskih sistemov v domenah digitalnega zdravja, kritičnih komunikacij in nadzora operaterskih omrežij.

Andrej Kos je doktoriral leta 2003 na Fakulteti za elektrotehniko Univerze v Ljubljani. Leta 2014 je bil izvoljen v naziv redni profesor za področje elektrotehnike. Novembra 2014 je prevzel mesto predstojnika Laboratorija za telekomunikacije (LTFE) na isti fakulteti. Trenutno se posveča področju uporabe IoT in blokovnih verig na različnih področjih.

Matevž Pustišek je doktoriral leta 2009 na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je trenutno tudi zaposlen. Raziskovalno se ukvarja z internetnimi storitvami in aplikacijami. Posebno pozornost namenja arhitekturi interneta stvari in varnostnim vidikom, v zadnjem času pa tudi uporabi tehnologij blokovnih verig v IoT.