Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Manja Cafuta

Bliskovito omrežje

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: prof. dr. Aleksandar Jurišić
Somentor: dr. Peter Nose

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Osrednji cilj diplomskega dela naj bo predstavitev novega koncepta pod imenom bliskovito omrežje (angl. Lightning Network), ki omogoča izvedbo večjega števila transakcij med lastniki bitcoinov (BTC). V delu predstavite kriptografske osnove, kot so npr. zgoščevalne funkcije in asimetrični kriptosistemi. Pojasnite tudi osnovne ideje kriptovalut, ki so potrebne za varno in učinkovito implementacijo. Podrobno preučite novo shemo bliskovitega omrežja za reševanje problema razširitve oziroma prilagoditve BTC omrežja.

Iskreno se zahvaljujem mentorju, prof. Aleksandru Jurišiću, za odlično usmerjanje, strokovno pomoč ter nasvete pri izdelavi diplomske naloge in edinstveno priložnost spoznati Univerzo v Waterlooju.

Zahvalo dolgujem tudi Nuši Zidarič za ves vloženi čas in dobre nasvete ter Petru Nosetu za predlog zanimive teme in koristne komentarje.

Največja zahvala gre dragima mami in očetu, ki sta me skozi ves študij podpirala in vedno verjela vame.

Hvala tudi vsem ostalim, ki ste mi vsa ta leta stali ob strani.

Kazalo

Slike

Seznam uporabljenih kratic

kratica	angleško	slovensko
SHA	Secure Hash Algorithm	kriptografska zgoščevalna funkcija SHA
PBKDF	Password-Based Key Derivation Function	funkcija za izpeljevanje ključev na podlagi gesla
BC	Blockchain	veriga blokov
P2P	peer-to-peer	vsak z vsakim
BTC	Bitcoin	kriptovaluta Bitcoin
LTC	Litecoin	kriptovaluta Litecoin
ETH	Ether	kriptovaluta Ether
PoW	Proof of Work	dokaz o delu
PoS	Proof of Stake	dokaz o vložku
P2PKH	Pay-to-PubKeyHash	"plačaj" na zgostitev javnega ključa
P2SH	Pay-to-ScriptHash	"plačaj" na naslov skripte
LN	Lightning Network	bliskovito omrežje
BFT	Byzantine fault tolerance	odpornost na problem bizantinskih generalov
CT, C	Commitment Transaction	zavezujoča transakcija
RSMC	Revocable Sequence Maturity Contract	prekliclјiva pogodba na podlagi zastaranja
HTLC	Hashed Timelock Contract	pogodba z zgoščeno časovno ključavnico

BGP	Border Gateway Protocol	usmerjevalni protokol BGP
IP	Internet Protocol	internetni protokol
VPN	Virtual Private Network	virtualno zasebno omrežje
LIFO	last-in-first-out	izvajanje po principu sklada
RAM	Random Access Memory	spomin z naključnim dostopom

Povzetek

Naslov: Bliskovito omrežje

Avtor: Manja Cafuta

V diplomskem delu opišemo rešitev problema skalabilnosti v omrežju kriptovalute Bitcoin. Eni ključnih elementov, ki jih kriptovalute uporabljajo, so zgoščevalne funkcije in kriptografija javnih ključev. Tehnologija, ki predstavlja temelj vseh kriptovalut, je veriga blokov. Prav mehanizmi, ki le-to naredijo varno za uporabo v omrežju brez centralne avtoritete, pa postavijo omejitev pri številu možnih transakcij. V osrednjem delu naloge predstavimo delovanje rešitve bliskovito omrežje, ki z mrežo plačilnih kanalov omogoči decentralizirano izvajanje velikega števila transakcij. Osredotočimo se na mehanizem kanalov s padajočimi časovnimi ključavnicami, ki obljublja varne transakcije izven verige blokov.

Ključne besede: bliskovito omrežje, kriptovalute, Bitcoin, skalabilnost, transakcije, kanali.

Abstract

Title: Lightning Network

Author: Manja Cafuta

The thesis describes a solution to the scalability problem of the cryptocurrency Bitcoin. Some of the key elements that cryptocurrencies use are hash functions and public key cryptography. Technology that constitutes the basis, all cryptocurrencies are built upon, is Blockchain. The mechanisms that make it safe and useful in networks without the central authority are the ones that also limit the number of transactions that are possible. The main topic of the thesis is Lightning Network, a solution that uses a network of payment channels to allow for larger quantities of transactions to be processed. We focus on the new mechanism, channels with decrementing timelocks, that promises secure offchain transactions.

Keywords: Lightning Network, cryptocurrencies, Bitcoin, scalability, transactions, channels.

Poglavje 1

Uvod

Bitcoin je najbolj široko uporabljena kriptovaluta na svetu. Kar predstavlja problem, je prav hitra rast popularnosti in s tem količine transakcij, ki jih je potrebno dodati v verigo blokov, ki jo bomo vpeljali v sekciji 2.3. Souporaba verige blokov, mehanizmov za doseganje soglasja in PoW zagotavlja varnost brez potrebe po zaupanju (oz. celo poznavanju ostalih vpletenih) ali centralni avtoriteti. Cena za to je poraba velikih količin računske moči. To pomeni omejitev števila transakcij, ki so lahko obdelane v nekem časovnem intervalu. Kaj hitro pridemo do ugotovitve, da veriga blokov ni optimalna tehnologija za uporabo v plačilnih omrežjih svetovnega merila, v kakršnega se Bitcoin razvija.

Živimo v času, ko je vprašanje zasebnosti vedno bolj pereča tema in ideja o anonimnem plačevanju vse privlačnejša. Torej kaj je lažjega, kot uporaba kriptovalut za vsakodnevno plačevanje. Hitro ugotovimo, da tako enostavno vseeno ni. Trenutno Bitcoin z omejitvijo velikosti bloka na 1 MB in konstantnim povprečnim časom 10 minut za dodajanje bloka v verigo omogoča manj kot 7 transakcij v sekundi. Če bi se želeli približati Visinemu omrežju, ki naj bi bilo, sodeč po testih, sposobno obdelati 56000 transakcij v sekundi [45], bi morali pri konstantnem času povečati dovoljeno velikost bloka na skoraj 8 GB.

Avtor Bitcoina Satoshi Nakamoto za priporočeno mejo, da se transakcijo

obravnava kot nespremenljivo, predlaga 6 potrditev [25] (5 blokov, ki sledijo bloku z dano transakcijo). Na ta način pustimo napadalcu manj kot 1 % možnosti za uspeh v primeru, da kontrolira do 10 % računske moči celotnega omrežja. Že v tem primeru bi moralo vsako vozlišče hraniti vsaj 6×8 GB $= 48$ GB podatkov.

Z letom 2010 so se v Bitcoinovem ekosistemu začela pojavljati prva *rudarska združenja* [43]. To so skupine vozlišč, ki sodelujejo s ciljem maksimizacije vsak svojega povprečnega dobička. S tem dobimo igralce, ki kontrolirajo večji del omrežja in potrebno število potrditev se skokovito poveča. Po ocenah Blockchaina ima trenutno največje združenje BTC.com skoraj 30 % tržni delež [41]. Če hočemo doseči, da imajo v primeru napada manj kot 1 % možnosti uspeha, potrebujemo vsaj 24 potrditev [25]. Že 24 blokov zahteva 24×8 GB $= 192$ GB prostora in to je le šestina povprečno dodanih blokov v enem dnevu. Samo za transakcije enega dne bi tako porabili več kot 1 TB prostora, kar bi precej zmanjšalo število vozlišč, ki so sposobna preverjanja blokov, in s tem vodilo v zlom ali v najboljšem primeru centralizacijo omrežja.

Za reševanje problema razširljivosti je bilo treba tako najti drugo rešitev. Pojavila se je množica predlogov za rešitev. Največji problem pri širitvi obsega poslovanja predstavljajo prenosi majhnih vsot (mikroplačila). Mikroplačila niso problem pri plačevanju z gotovino ali bančnimi karticami, medtem ko mora biti pri Bitcoinu vsaka transakcija preverjena in dodana v blok, za to pa se zaračuna transakcijsko provizijo. Pri prenosu velikih količin sredstev provizije ne predstavljajo težave. Za prenose manjše količine pa provizija, ki presega vrednost plačila, predstavlja precejšnjo neugodnost in postavi pod vprašaj smiselnost uporabe. Selitev dela transakcij izven verige blokov tako izgleda smiselna, z naraščanjem računske zahtevnosti in posledično dviganjem transakcijskih stroškov pa tudi vedno bolj realna rešitev.

V diplomskem delu spoznamo bliskovito omrežje, ki je rešitev Bitcoinovega problema skalabilnosti, in podrobno opišemo njegovo delovanje. V 2. poglavju predstavimo temelje, na katere se opirajo kriptovalute in so pomembni za razumevanje delovanja nove plasti v omrežju. Spoznamo strukturo in de-

lovanje verige blokov, razložimo zgoščevalne funkcije in izvemo, kako delujejo digitalni podpisi. V 3. poglavju bolj natančno spoznamo kriptovaluto Bitcoin in povemo nekaj tudi o njegovi vejitvi Litecoin, ki je poleg Bitcoina trenutno edina kriptovaluta z Lightning podporo. 4. poglavje predstavlja jedro diplomske naloge in se sprehodi skozi mehanizme delovanja bliskovitega omrežja, od preprostih kanalov, preko uporabe pametnih pogodb s časovnimi ključavnicami, do konstrukcije delujočega omrežja kanalov. Na koncu povemo še nekaj o do sedaj odkritih ranljivostih ter v 5. poglavju zaokrožimo celoto s sklepnimi mislimi.

Poglavje 2

Teoretične osnove

V drugem poglavju najprej opišemo zgoščevalne funkcije, s poudarkom na SHA-256, ki je uporabljena v Bitcoinovem mehanizmu dokaz o delu (opisan v razdelku 2.3.4). Uporabimo jih tudi v postopku kreiranja digitalnih podpisov, ki jih opišemo v naslednji sekciji. V sekciji 2.3 predstavimo še strukturo in delovanje verige blokov ter mehanizme, ki zagotavljajo varno uporabo v porazdeljenem omrežju.

2.1 Zgoščevalne funkcije

Zgoščevalna funkcija je funkcija, ki preslika podatke poljubne velikosti v niz točno določene (manjše) velikosti. Kriptografske zgoščevalne funkcije predstavljajo razred znotraj družine zgoščevalnih funkcij z določenimi lastnostmi, zaradi katerih so primerne za uporabo v kriptografiji. Rezultat funkcije je zgoščena vrednost, imenovana zgostitev (tudi izvleček, prstni odtis).
Kar naredi zgoščevalne funkcije zanimive za uporabo v kriptografiji, je enostaven izračun zgostitve na podlagi vhodnih podatkov in hkrati potreba po vložku ogromne količine računske moči za obratno operacijo – izračun začetnih podatkov iz znane zgostitve. Idealna kriptografska zgoščevalna funkcija ima naslednje lastnosti:

- je deterministična (enako vhodno sporočilo da vedno isto zgoščeno vrednost),
- izračun zgoščene vrednosti je hiter za vsako vhodno sporočilo,
- neizvedljivo je rekreirati začetno sporočilo, razen s poskušanjem vseh možnosti,
- majhna sprememba vhodnega sporočila povzroči tako veliko spremembo zgostitve glede na originalnega, da med njima ni nobene vidne povezave,
- nemogoče je v doglednem času najti dve različni sporočili z enako zgostitvijo.

Nivo varnosti, ki ga zagotavlja zgoščevalna funkcija, je določen na podlagi odpornosti na znane vrste kriptografskih napadov [19]:

- **odpornost na napad s prasliko** (angl. pre-image resistance) pomeni, da je računsko neizvedljivo v doglednem času najti katero koli možno vhodno sporočilo, za katerega funkcija vrne želeno zgoščeno vrednost; (z drugimi besedami, za dano vrednost y je zelo težko najti vrednost x, da velja $f(x) = y$),
- **odpornost na napad z drugo prasliko** (angl. second pre-image resistance) pomeni, da je računsko neizvedljivo v doglednem času najti katero koli možno sporočilo, za katerega funkcija vrne enako zgoščeno vrednost kot za znano vhodno sporočilo (z drugimi besedami, za dano vrednost x je zelo težko najti x', da velja $f(x) = f(x')$ pri $x \neq x'$),
- **odpornost na trke** (angl. collision resistance) pomeni, da je računsko neizvedljivo v doglednem času najti dve različni sporočili z istima zgoščenima vrednostma (z drugimi besedami, najti par sporočil m_1 in m_2, da velja $H(m_1) = H(m_2)$, je zelo težko. Če nam to uspe, pravimo, da gre za trčenje).

SHA-2

Večina znanih kriptografskih funkcij deluje na osnovi bločnih šifer. V to skupino spadajo MD4, MD5 in družina SHA funkcij, z izjemo zadnje pridobitve SHA-3. Kljub temu da ima pri implementacijah v strojni opremi SHA-3 prednost pri hitrosti, je še vedno najbolj pogosto uporabljena SHA-2 [11]. Slednja je naslednik algoritma SHA-1. Le-ta je 160-bitna zgoščevalna funkcija, kar pomeni, da potrebujemo 2^{159} poskusov za odkritje praslike in v povprečju $\sqrt{2^{160}} = 2^{80}$ poskusov za trčenje (glej. rojstnodnevni napad [19]). Sprememba standarda v letu 2010, ki zahteva, da je nivo zaščite vsaj 112 bitov namesto prvotnih 80 [6] (za razbitje je potrebno v povprečju $2^{st.bitov}$ poskusov), je imela za posledico zamenjavo SHA-1 z družino SHA-2. Uspešen napad s trki februarja 2017 [37] pa je dokončno potrdil SHA-1 kot neuporabno za nadaljnjo uporabo v kriptografiji.

SHA-2 je družina funkcij z zgostitvami različnih dolžin, najpogosteje z 224-, 256-, 384- in 512-bitnimi. V primeru krajših sporočil algoritmi začnejo z dopolnjevanjem sporočila z ničlami do dolžine, ki je večkratnik dvakratne dolžine zgostitve.

SHA-256 deluje na blokih dolžine 512 bitov. Algoritem [13] začne z inicializacijo zgostitev $h_0,\ldots,h_7$ na prvih 32 bitov decimalnega dela kvadratnih korenov prvih 8 praštevil in tabele konstant na prvih 32 bitov decimalnega dela kubičnih korenov prvih 64 praštevil. Vsak blok sporočila nato razbije na besede dolžine 32 bitov, ki jih shrani na prvih 16 mest v tabeli s 64 vnosi, in na podlagi teh izračuna in zapolni še preostanek tabele. Glavna zanka v algoritmu teče skozi vse besede v tabeli. V vsakem koraku uporabi zgostitve iz prejšnje iteracije (glej sliko 2.1) in izračuna nove. Vse operacije seštevanja izvede po modulu 2^{32}, za izračun vrednosti v modrih škatlah na sliki uporabi naslednje operacije, ki so definirane z logičnimi operatorji $\wedge$[1], $\oplus$[2], $\neg$[3] in $\ggg$[4] [13]:

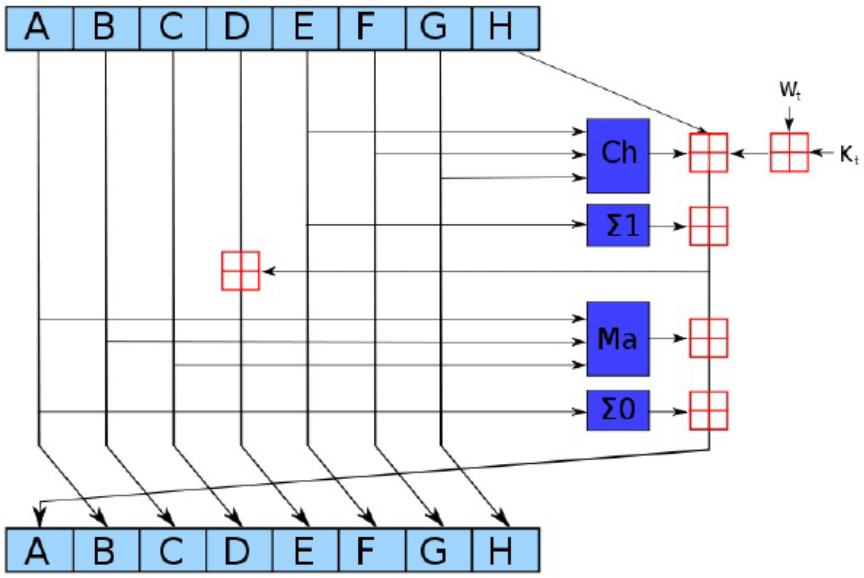
[1] $\wedge$ – logični operator in
[2] $\oplus$ – logični operator xor
[3] $\neg$ – negacija
[4] $\ggg n$ – zamik za n mest v desno

$$\mathrm{Ch}(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$$

$$\mathrm{Ma}(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$

$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$

$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$



Slika 2.1: Delovanje glavne zanke SHA-256 [24]. V prvi iteraciji A,...,H inicializiramo na $h_0,\ldots,h_7$. K_t označuje konstante, W_t pa vhodne podatke.

SHA-256 izvede zanko na sliki 2.1 64-krat. V zadnjem koraku sestavi zgostitev z združevanjem A do H v niz dolžine 256 bitov.

2.2 Digitalni podpisi

Digitalni podpis (angl. Digital Signature) je rezultat kriptografske transformacije podatkov, ki zagotavlja mehanizem za preverjanje avtentičnosti sporočila, celovitosti podatkov in preprečevanje možnosti zanikanja [3]. Sestavljen je iz dveh delov: algoritma za generiranje podpisa in algoritma za

preverjanje podpisa. Podpisnik sporočila je lastnik para zasebnega in javnega ključa. *Zasebni ključ* (angl. Private Key) je znan samo njemu, kar preprečuje ponarejanje. *Javni ključ* (angl. Public Key) je splošno dostopen in ga prejemnik sporočila uporabi za preverjanje verodostojnosti. Pred uporabo javnega ključa se mora prejemnik prepričati o lastništvu, za kar poskrbi zaupanja vredna tretja oseba ali institucija – najpogosteje certifikatna agencija (CA).
Opis algoritmov v nadaljevanju je povzet po standardu za digitalne podpise [3] in López&Dahab [21]. Za bolj poglobljeno razlago in nadaljnje branje glej Jurišić&Menezes [18].

2.2.1 DSA

Algoritem *DSA* uporablja za izračun digitalnega podpisa množico domenskih parametrov, zasebni ključ x, skrivnost k, podatke, ki jih podpisujemo, in zgoščevalno funkcijo. Za preverjanje je uporabljena ista množica domenskih parametrov, javni ključ, podatki, ki jih preverjamo, in ista zgoščevalna funkcija, kot je bila uporabljena za generiranje podpisa. Generiranje ključev poteka v dveh fazah. V prvi izberemo vhodne parametre algoritma, ki si jih lahko deli več uporabnikov:

- p – praštevilski modul dolžine L bitov,
- q – praštevilski delitelj števila $p-1$ dolžine N bitov,
- g – generator podgrupe reda q v $\mathbb{Z}_p$, kjer velja $1 < g < p$.

V drugi fazi sledi izračun zasebnega in javnega ključa posameznega uporabnika:

- x – zasebni ključ, ki je naključno generirano celo število med 1 in q,
- y – javni ključ, ki je generiran kot $y = g^x \mod p$.

Podpis določenega sporočila je generiran kot par (r, s), s sledečimi značilnostmi:

- k – skrivnost, ki je naključno generirano celo število med 0 in q,
- $H(m)$ – zgoščena vrednost sporočila,
- $r = (g^k \mod p) \mod q$,
- $s = k^{-1} \times (H(m) + x \times r) \mod q$.

Preverjanje poteka v več korakih. Sporočilo je zavrnjeno, če ne velja $0 < r < q$ ali $0 < s < q$. Sledi izračun:

- $w = s^{-1} \mod q$,
- $u_1 = (H(m) \times w) \mod q$,
- $u_2 = (r \times w) \mod q$,
- $v = ((g^{u_1} \times y^{u_2}) \mod p) \mod q$.

Podpis je veljaven, če velja $v = r$.

2.2.2 ECDSA

ECDSA je algoritem za digitalne podpise, ki uporablja kriptografijo z eliptičnimi krivuljami. Opira se na algebraične značilnosti eliptičnih krivulj v končnih obsegih (angl. Finite, Galois Fields, GF) in zato za zagotavljanje enake stopnje varnosti potrebuje precej krajše ključe kot DSA. Za stopnjo varnosti n-bitov (napad s surovo silo (angl. Brute Force) potrebuje za razbitje 2^n operacij) je potrebna dolžina javnega ključa $2 \times n$. Algoritem sprejme naslednje parametre, ki so javni:

- q – velikost obsega (ponavadi vzamemo praštevilski obseg $\mathbb{Z}_p$ in je $q = p$),
- FR – indikacija o uporabljeni bazi,
- a in b – elementa obsega, ki definirata enačbo krivulje,

- dps – seme domenskih parametrov (angl. Domain Parameter Seed), je poljuben niz, ki omogoča generiranje preverljivih naključnih parametrov,
- P – bazna točka praštevilskega reda na krivulji ($P = (X_P, Y_P)$),
- n – red točke P,
- h – kofaktor, ki je enak kvocientu reda krivulje in n.

Zasebni ključ je kriptografsko varno naključno izbrano število d iz intervala $[1, n-1]$, javni pa točka na krivulji $Q = d \times P$. Digitalni podpis predstavlja par (r, s), ki je generiran na podlagi skrivnosti k, ki se določi na novo za vsako sporočilo. Postopek generiranja podpisa je sledeč:

- izračun zgoščene vrednosti sporočila $e = H(m)$ in z – prvih L bitov števila e, kjer je L dolžina reda grupe n v bitih,
- generiranje naključnega naravnega števila k iz intervala $[1, n-1]$,
- izračun točke na krivulji $(x_1, x_2) = k \times P$,
- izračun $r = x_1 \mod n$,
- izračun $s = k^{-1} \times (z + d \times r) \mod n$,
- ponovitev v primeru, da je katera od vrednosti r, s enaka 0.

Za preverjanje podpisa potrebuje prejemnik veljaven pripadajoč javni ključ in prej specificirane domenske parametre. Preverjanje poteka v dveh korakih. Prejemnik se najprej prepriča, da sta r in s naravni števili iz intervala $[1, n-1]$, nato pa sledi izračun:

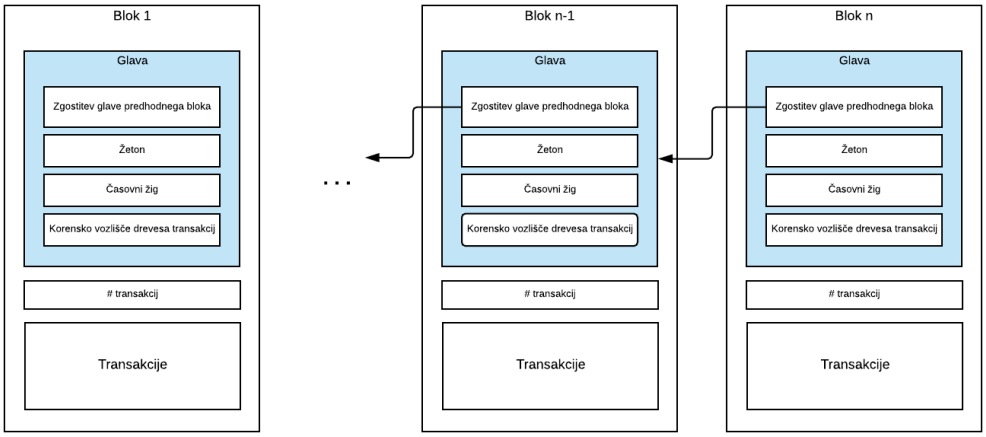
- $e = H(m)$,
- z – prvih L bitov zgostitve števila e,
- w – obratna vrednost s po modulu n ($w = s^{-1} \mod n$),

- $u_1 = (z \times w) \mod n$,
- $u_2 = (r \times w) \mod n$,
- $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$,

Podpis je veljaven, če $(x_1, y_1) \neq 0$ in $r = x_1 \mod n$.

2.3 Veriga blokov

Veriga blokov (angl. Blockchain) je globalno gledano porazdeljen seznam, katerega člene predstavljajo strukture, imenovane bloki (glej sliko 2.3). Preprost blok v verigi vsebuje transakcije, zgoščeno vrednost prejšnjega bloka v verigi, žeton in časovni žig.



Slika 2.2: Preprosta veriga blokov

Za začetke verige blokov šteje leto 2008, ko je anonimna oseba oz. skupina ljudi pod psevdonimom Satoshi Nakamoto predlagala koncept verige blokov in ga v naslednjem letu tudi implementirala kot temelj kriptovalute Bitcoin. Kriptovalute uporabljajo verigo blokov za porazdeljeno bazo podatkov v omrežju tipa *vsak za vsakim* (angl. peer-to-peer, P2P). V resnici se je prvi

opis ideje o kriptografsko zaščiteni verigi blokov pojavil že veliko prej. Že leta 1991 sta ga vpeljala S. Haber in W. S. Stornetta [12]. Verigo blokov srečamo tudi pri časovnem žigu (glej Stinsonov učbenik Cryptography: Theory and Practice [33, sek. 7.8]).

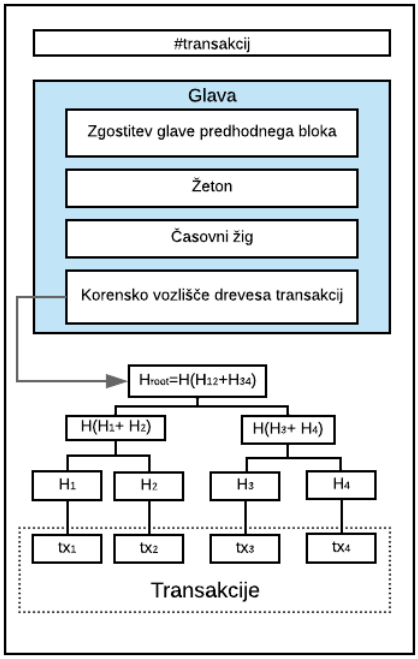
2.3.1 Struktura in delovanje

Veriga blokov je podatkovna struktura, ki je s programerskega vidika povezan seznam blokov. V nadaljevanju bomo opisali osnovne elemente, ki so potrebni za razumevanje njenega delovanja.

Vozlišča v porazdeljenem omrežju predstavljajo posamezne naprave, ki posedujejo kopijo verige blokov. Njihove naloge se razlikujejo glede na tip verige in njihovo vlogo v omrežju. Večinoma imamo dve vrsti vozlišč: končne uporabnike in rudarje. Končni uporabnik je v omrežju s ciljem uporabe storitve. Vozlišče ne hrani celotne verige blokov, ampak samo glave, in preverja le, ali je bila določena transakcija vključena v nek blok. Rudarji hranijo celotno verigo. Njihova naloga je preverjanje in potrjevanje transakcij ter vključevanje blokov v verigo. Algoritem je podrobneje opisan v razdelku 2.3.1. Na tem mestu omenimo le, da uporablja mehanizem, ki je običajno zasnovan na principu dokaza o delu. Le-ta za procesiranje transakcij porabi velike količine računske moči. To vodi v uporabo namenskih naprav za rudarjenje, ki imajo veliko prednost pred navadnimi računalniki.

Naslovi so enolični identifikatorji, ki jih uporabljamo za prepoznavanje pošiljatelja in prejemnika. Po navadi naslov nekega uporabnika predstavlja njegov javni ključ (ali izpeljanka le-tega). V splošnem to sicer preprečuje možnost neposrednega razkritja identitete, a le v primeru, da za vsako transakcijo generiramo nov par ključev. Uporaba istega ključa pri več transakcijah lahko poveča možnost povezave le-teh z uporabnikom in na ta način omogoči identifikacijo uporabnika.

Blok je, kot omenjeno že v uvodu, osnovni gradnik verige blokov. Zaporedni bloki se med seboj povezujejo s kazalci. Edini blok v verigi, ki ne vsebuje kazalca na prejšnji blok, je prvi, *začetni blok* v verigi (angl. genesis block). Struktura samega bloka je odvisna od tipa verige, v splošnem pa vsi vsebujejo nekaj elementov, ki so ključni za delovanje bloka: glavo bloka, števec transakcij in drevo transakcij (ki ga bomo podrobneje opisali v kratkem). Glava vsebuje kazalec na prejšnji blok, časovni žig, žeton in pa korensko vozlišče drevesa transakcij.



Slika 2.3: Blok z osnovnimi gradniki (Bitcoinov blok).

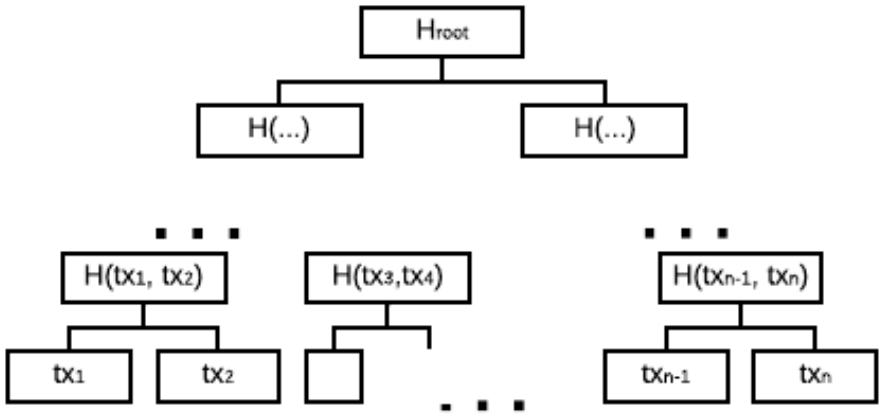
Transakcija je osnovni gradnik bloka in predstavlja prenos vrednosti z enega naslova na drugega (glej sliko 3.2 za podrobnejšo razlago na primeru Bitcoin transakcije).

Žeton (angl. nonce, token) je naključna številka, ki jo rudar vključi v blok, da doseže konkretno obliko zgostitve, ki ustreza trenutnemu nivoju zahtevnosti.

Časovno žigosanje (angl. time stamping) je mehanizem, ki ga uporabljamo za dokazovanje obstoja nekih podatkov v določenem trenutku in zagotovilo, da se od takrat naprej le-ti niso več spremenili. Recimo, da želimo prej omenjeni lastnosti za nek podatek x. Po Stinsonu [33] najprej izračunamo njegovo zgostitev $h = H(x)$ in h podpišemo s svojim zasebnim ključem $y = \mathrm{sig}(h)$. V primeru, da uporabimo zaupanja vrednega *izdajatelja časovnih žigov* (angl. Time Stamp Authority, TSA), mu pošljemo par (h, y). TSA temu doda datum D in nam vrne podpisano trojico (h, y, D). V kolikor zaupamo TSA, je to dokaz, da je podatek x obstajal pred datumom D, saj bi imela vsaka sprememba podatka x za rezultat popolnoma drugačno zgostitev. Če bi hoteli dokazati še, da smo x podpisali po določenem datumu, bi morali h pred podpisom dodati še neko aktualno informacijo javnega značaja i (npr. statistične podatke o delnicah od prejšnjega dne ...), in izračunati $y = \mathrm{sig}(h, i)$.
V primeru, da namesto TSA uporabljamo decentralizirano časovno žigosanje, bo zgornjo časovno mejo postavila objava para (h, y) v verigi blokov.
V verigi blokov uporabimo časovni žig za indikacijo o času nastanka bloka.

Drevo zgoščenih vrednosti transakcij (Merkle tree) [26] bomo opisali z binarnim drevesom (glej sliko 2.4). Vozlišče imenujemo *list*, če ima stopnjo 1 (na naši sliki so na dnu). V listih se nahajajo zgoščene vrednosti posameznih blokov podatkov. Vsako vozlišče, ki ni list, vsebuje zgoščeno vrednost para sosednjih vozlišč pod njim (naslednikov). Drevesa zgoščenih vrednosti tako omogočajo učinkovito preverjanje, ali je določena vrednost list v drevesu, saj je za to potrebno preveriti le vse starše po določeni veji – izračun $\log_2(n)$ vrednosti pri n listih v drevesu. Drevo zgoščenih vrednosti je podatkovna struktura s širokim spektrom uporabe v kriptografiji. Veriga blokov uporablja Merkle drevo transakcij za hitro preverjanje, ali je neka

transakcija del določenega bloka ali ne.



Slika 2.4: Merkle drevo transakcij

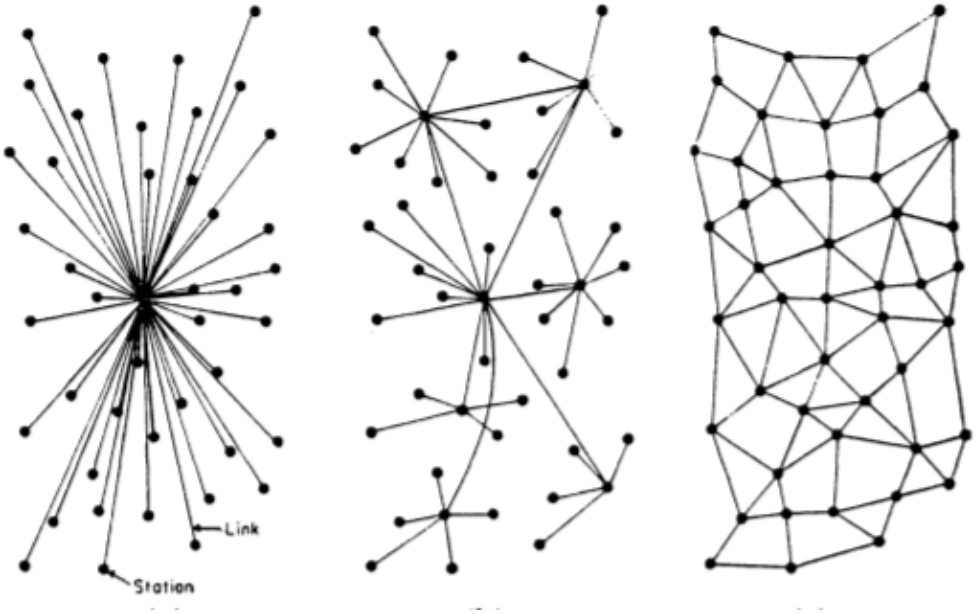
Delovanje verige blokov

Ko želi uporabnik opraviti plačilo, kreira novo transakcijo, jo podpiše s svojim zasebnim ključem in jo objavi. Ostala vozlišča prejeto transakcijo preverijo na podlagi vnaprej pripravljenih pravil. Transakcija sedaj čaka, da jo bo kateri od rudarjev vključil v blok in bo s tem dobila svojo prvo potrditev. Pot bloka do tega, da postane del verige, lahko v grobem razdelimo na dve fazi. Ko se v bloku nabere dovolj transakcij, nad katerimi so vozlišča dosegla soglasje, začnejo vozlišča s tekmo za vključevanje bloka v verigo. Ko eno od njih uspešno najde rešitev problema (v primeru Bitcoina hashcash), lahko blok objavi in za to pobere denarno nagrado z v bloku vključeno transakcijo za generiranje sredstev (angl. coinbase transaction). V tej fazi postane blok del verige. S tem dobi blok prvo potrditev, transakcija pa svojo drugo. Naslednji blok, ki se bo vključil v verigo, bo pripet nanj (ali na katerega od drugih blokov, če veja verige, v kateri smo, ni najdaljša) s kazalcem na nje-

govo zgoščeno vrednost. Z vsakim novim blokom, ki postane del njegove veje verige, dobijo tako blok kot transakcije v njem dodatno potrditev. Transakcijo lahko smatramo kot uspešno opravljeno že ob vključitvi v blok. Kljub temu je pri večini verig za to, da je transakcija dokončno zaključena, kot smo omenili že v uvodu, potrebnih več potrditev.

2.3.2 Decentralizacija

Decentralizacija predstavlja temelj verige blokov in tudi eno njenih glavnih prednosti. Kot pove že ime, pomeni odpravo *centralne avtoritete* – posrednika, ki bi preverjal veljavnost transakcij. S tem močno zmanjša tveganje za zlom sistema, saj so podatki razpršeni med več sodelujočimi vozlišči in tako ni več centralne točke odpovedi. Vsako vozlišče poseduje kopijo verige, kar sicer otežuje kontrolo nad podatki, a hkrati tudi prepreči (ali pa vsaj močno oteži) manipulacijo le-teh. Za preverjanje in potrjevanje transakcij so namesto centralne avtoritete uporabljeni t. i. mehanizmi za doseganje soglasja med vozlišči v porazdeljenem omrežju.
Pogosto zmoto predstavlja zamenjevanje decentraliziranega sistema s porazdeljenim (glej sliko 2.5). V idealnem primeru decentralizacije bi lahko besedi uporabljali kot sopomenki, v splošnem pa je razlika bistvena. Pri decentralizaciji gre za odpravo enega centra, ki bi predstavljal točko odpovedi za celotno omrežje. Še vedno imamo znotraj omrežja določeno mero centraliziranosti v obliki podomrežij s po eno točko odpovedi. V našem omrežju so to vozlišča z veliko povezavami. Porazdeljeno omrežje (kot idealen primer decentralizacije) je struktura, v kateri odpoved enega člena ne predstavlja nobenega problema, saj sta med vsakim parom članov omrežja vsaj dve različni poti brez skupnih točk odpovedi.



Slika 2.5: Od leve proti desni: centralizirano omrežje, decentralizirano omrežje, porazdeljeno omrežje [5].

2.3.3 Porazdeljeno soglasje in problem bizantinskih generalov

V porazdeljenih sistemih predstavlja osnovni problem *doseganje soglasja* med posameznimi vozlišči v omrežju, ki si med sabo ne zaupajo. Drug za drugega ne vedo, ali so poštena ali zlonamerna. V bistvu soglasje predstavlja dogodek, ko dovolj veliko število vozlišč sprejme transakcijo kot veljavno.
Po Bashirju morajo biti za dosego (porazdeljenega) soglasja izpolnjene naslednje zahteve [7]:

- strinjanje (vsa poštena vozlišča se strinjajo z neko vrednostjo),
- končni čas izvajanja (vsa poštena vozlišča bodo sprejela odločitev v nekem doglednem času),
- veljavnost (transakcija, na podlagi katere je bilo doseženo soglasje, je bila predlagana s strani vsaj enega poštenega vozlišča),

- odpornost na napake (algoritem mora biti sposoben učinkovitega delovanja tudi v prisotnosti pokvarjenih ali zlonamernih vozlišč),
- doslednost (vsako vozlišče sprejme odločitev le enkrat v ciklu za doseganje soglasja in je ne spreminja).

Problem bizantinskih generalov in odprava napak

Recimo, da deljena mnenja povzročijo nastanek dveh skupin, ki pa morata za uspešno izvedeno akcijo doseči soglasje. Problem ima korenine v vojaških strategijah. Gre za skupino generalov, katerih vsak vodi del bizantinske vojske, s ciljem, da napadejo in osvojijo mesto. Njihova dilema je preprosta, napad ali umik. To jih razdeli v dve skupini. Nasprotujoči si skupini generalov se morata zediniti in izbrati isto: koordinirani napad ali koordinirani umik. Pri tem se pojavi več problemov, ki izvirajo iz dejstva, da se generali med sabo ne vidijo (torej ne morejo komunicirati neposredno, ampak le prek kurirjev).
Največjo nevarnost predstavljajo izdajalski generali, katerih cilj ni iskanje optimalne strategije, pač pa so pripravljeni celo povzročati zmedo z morebitnimi lažnimi sporočili. Dodaten problem predstavlja komunikacija preko kurirja, ki je nezanesljiv.
Na videz preprost problem se hitro spremeni v zelo zahtevnega. Če ga prenesemo v okvir porazdeljenih omrežij, so generali vozlišča, ki se trudijo doseči soglasje glede veljavnosti transakcij, in posredniki komunikacijske poti (povezave) med njimi. Bizantinska vozlišča (kot imenujemo zlonamerna vozlišča), so izdajalci in cilj je najti mehanizem, ki bo deloval navkljub prisotnosti leteh.
Bizantinska napaka ne pomeni nujno, da je neko vozlišče zlonamerno, le da (občasno) posreduje različna (zavajajoča) sporočila različnim sosednim vozliščem. Odpornost nanjo je možno zagotoviti, če poštena vozlišča dosežejo večinsko soglasje. Ob neodzivnosti nekega vozlišča je lahko vnaprej dogovorjen odgovor `null`, prav tako je v primeru večinskega odgovora `null` pripravljena privzeta strategija (ponavadi umik).

V grobem lahko mehanizme za dosego soglasja razdelimo v dve kategoriji:

- mehanizmi za dosego soglasja na podlagi dokaza (vozlišča izberejo vodjo in ta predlaga končno vrednost),
- mehanizmi, odporni na problem bizantinskih generalov (tradicionalni pristop, ki doseže soglasje skozi kroge glasovanja).

2.3.4 Mehanizmi za zagotavljanje varnosti

Pri rokovanju z denarjem se ne moremo izogniti vprašanju varnosti. Pri klasičnih, *FIAT valutah*[5] s tem nimamo težav, saj nam varnost zagotavlja centralna avtoriteta – banka. V trenutku, ko to centralno avtoriteto odpravimo, pa se pojavi vprašanje, kako se prepričati, da bodo naš denar ali naši podatki res varni. V verigi blokov se da ta problem rešiti na več načinov. Najpogosteje uporabljena mehanizma, s katerima je zagotovljena varnost, sta koncepta dokaz o delu in dokaz o vložku, ki ju predstavimo v nadaljevanju.

Dokaz o delu

Dokaz o vloženem delu (angl. Proof of Work, PoW) je najbolj uveljavljen mehanizem za zagotavljanje varnosti. Prvotno je bil razvit z namenom preprečevanja napadov za zavrnitev storitve (angl. Denial of Service, DOS) in omejevanja nezaželjenih sporočil (angl. spam) pri elektronski pošti. Za dokaz o vloženem delu nam služi rešitev določene kriptografske uganke (npr. hashcash pri Bitcoinu). Ideja mehanizma je, da je rešitev uganke računsko precej zahteven problem, medtem ko je preverjanje le-te hitro in računsko nezahtevno. Pri verigi blokov je za vključitev posameznega bloka v verigo zahtevan izračun zgostitve prejšnjega bloka in neke vrednosti, imenovane žeton (angl. nonce, token). Problem, s katerim se srečamo pri tem, je najti tak žeton, da bo zgoščena vrednost ustrezala predpisanim zahtevam. Najpogosteje se uporabljata algoritma SHA-256, ki je računsko zelo zahteven, in

[5]To ime se uporablja za denar, ki za kritje nima zlata (ali drugega fizičnega blaga), ampak ima plačilno vrednost le zato, ker je tako določeno, glej [22].

scrypt, ki je preprostejši, a porabi večje količine spomina z naključnim dostopom (angl. Random Access Memory, RAM). V primeru Bitcoina uporabimo dve iteraciji SHA-256, zgoščena vrednost pa mora imeti določeno število vodilnih ničel.
Dejstvo, da rešitev uganke zahteva vnaprej določeno, precej veliko količino dela, naredi spreminjanje nekega bloka, ki je že del verige, precej neučinkovito. Za uspešen napad bi morali ponovno izračunati tudi zgoščene vrednosti vseh sledečih blokov, kar bi zahtevalo ogromno količino računske moči. Prav to učinkovito preprečuje (ali vsaj zelo otežuje) tudi napade, pri katerih bi se neko vozlišče lahko pretvarjalo, da predstavlja več vozlišč in si s tem zagotovilo večji vpliv v omrežju (angl. Sybil attacks [4]).

Dokaz o vložku

Za razliko od PoW se mehanizem *dokaz o vložku* (angl. Proof of Stake, PoS) ne poslužuje rudarjenja in se s tem izogne porabi ogromnih količin elektrike. Izbira vozlišča zgolj na podlagi vložka v sistemu bi vodila v centralizacijo, saj bi imelo vozlišče z največ sredstvi stalno prevlado nad ostalimi. Vozlišče, ki bo predlagalo naslednji blok, se lahko določi na dva načina: na podlagi različnih kombinacij naključne izbire in količine sredstev, ki jih le-to poseduje, ali pa na podlagi starosti kovancev. Pri prvem načinu, ki ga uporabljata kriptovaluti Nxt [16] in Blackcoin [38], je naslednje vozlišče tisto, ki ima kombinacijo najnižje zgoščene vrednosti in največjega vložka. Najbolj poznan predstavnik drugega načina je Peercoin [42]. Pri njem se vozlišče vključi v tekmo za naslednji blok, ko so bili njegovi kovanci nazadnje porabljeni za podpis bloka pred 30 dnevi, predlagatelja bloka pa se določi na podlagi kombinacije starosti in velikosti vložka.

2.3.5 Pametne pogodbe

Za začetke pametnih pogodb štejemo leto 1994, ko je Nick Szabo objavil članek [34], v katerem definira pametno pogodbo kot računalniški transakcijski protokol, ki izvede pogodbene pogoje. Njihov glavni cilj je zadostiti

pogojem, definiranim v pogodbi, in minimizirati potrebo po posrednikih.
Prva in tudi najbolj razširjena platforma, namenjena implementaciji pametnih pogodb, je Ethereum. Za razliko od Bitcoina in še nekaterih, ki so sicer že prej omogočale implementacijo pametnih pogodb v obliki skript, Ethereum uvede nov Turingovo popoln programski jezik Solidity in koncept virtualnih strojev EVM (angl. Ethereum Virtual Machine).
Od začetka do danes se je definicija pametne pogodbe spreminjala oz. razširila in še danes obstaja več različnih [20]. S tehničnega vidika bomo mi privzeli, da je *pametna pogodba* avtonomen program, ki se nahaja na določenem naslovu v verigi blokov. Izvajanje pogodbe sproži kombinacija dogodka, specificiranega v pogodbi, in transakcije s podatki na naslovu v verigi. Program se lahko izvede neomejenokrat, za izvajanje pa poskrbi porazdeljen virtualni stroj v verigi blokov. Pametne pogodbe niso odvisne od verige blokov, a je ta zaradi varnosti, ki jo nudi, postala že standardna platforma za decentralizirano izvajanje pametnih pogodb pri sodobnih implementacijah.

Poglavje 3

Kriptovalute

Kriptovalute so digitalni sistem za izmenjavo sredstev tipa vsak z vsakim, pri katerem je za generiranje sredstev in njihov prenos ključna kriptografija [24]. Kriptovalute predstavljajo razred znotraj družine digitalnih sredstev, še zdaleč pa ne prvega predstavnika. Digitalna sredstva so sredstva, ki obstajajo le v digitalni obliki, ne pa tudi v fizični, in so se prvič pojavila že veliko prej. Za pionirja lahko štejemo Davida Chauma, ki je predstavil idejo anonimnega elektronskega monetarnega sistema z uporabo slepih podpisov [9] že leta 1983 in jo kasneje tudi realiziral s podjetjem DigiCash v obliki programske opreme eCash. Ta je lokalno na uporabnikovem računalniku shranil denar v digitalni obliki, kriptografsko podpisan s strani banke, uporabnik pa ga je lahko nato porabil za plačilo fizičnih dobrin in storitev. Temu je sledilo še kar nekaj monetarnih sistemov, kar pa jim je skupno, je, da vsi bazirajo na FIAT denarju [22, stran 81]. Po drugi strani pa imajo kriptovalute svojo valuto. Značilnost, ki je vodila v popularizacijo kriptovalut, je decentralizacija – odprava potrebe po centralni avtoriteti za preverjanje transakcij. V sistemih kriptovalut varnost, integriteto in uravnoteženost seznama vzdržuje skupina rudarjev, ki se med seboj ne poznajo in si ne zaupajo. *Rudar* (angl. miner) je izraz, ki opisuje posameznika, ki uporablja svoj računalnik za potrjevanje in časovno žigosanje transakcij. Cilj je uspešno dodati blok s transakcijo v seznam (verigo blokov) in za to prejeti finančno nagrado.

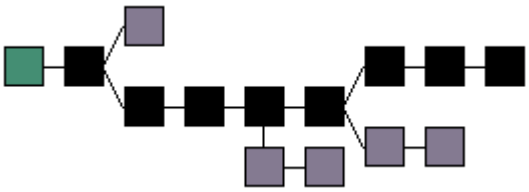
3.1 Bitcoin

Bitcoin (BTC) je prva popolnoma decentralizirana kriptovaluta, hkrati pa ga štejemo tudi za prvo implementacijo verige blokov. Idejo je prvič predstavil posameznik ali skupina ljudi pod psevdonimom Satoshi Nakamoto, leta 2008, v članku Bitcoin: A Peer-to-Peer Electronic Cash System [25]. Bitcoin združuje kriptografske prvine in tehnologije, ki so bile plod desetletij raziskav na tem področju. Uporablja programski jezik script, ki izvaja ukaze po principu sklada (angl. last-in-first-out, LIFO) in v izogib neskončnemu izvajanju ne podpira zank. Za generiranje parov zasebnih in javnih ključev je v Bitcoin omrežju uporabljena kriptografija z eliptičnimi krivuljami, podrobneje opisana v razdelku 2.2.2.
Bitcoin omrežje je namenjeno plačevanju tipa vsak z vsakim. Za začetek Bitcoina štejemo januar 2009, ko je Satoshi Nakamoto uspešno zminiral *izvorni blok* (angl. genesis block) v verigi, le nekaj dni po tem, ko je bil izdan prvi odprtokodni Bitcoin odjemalec. Protokol je dobil do danes precej novih verzij, ki so odpravljale ranljivosti in predstavile izboljšave. Kljub hitri rasti popularnosti in s tem tudi številu poskusov izkoriščanja, pa je trda vejitev iz leta 2010 [44] posledica edine uspešno izkoriščene varnostne luknje v protokolu.

3.1.1 Veriga blokov

Struktura bloka v Bitcoinovi verigi je precej preprosta. Sestavljajo ga glava, števec transakcij in transakcije. V glavi se nahajajo podatki o verziji, zgoščeni vrednosti glave prejšnjega bloka in korena Merkle drevesa, stopnje zahtevnosti uganke (številu vodilnih ničel), časovni žig, in žeton.
Za dodajanje blokov Bitcoin uporablja že opisan mehanizem dokaz o delu z reševanjem uganke hashcash. Za zavarovanje pred napadom z dvojno porabo

vozlišča za veljavno verzijo verige vedno upoštevajo najdaljšo verigo. V primeru, da dve vozlišči objavita blok naenkrat, se veriga razveji. Dokler se ena veja ne podaljša, vozlišča iščejo naslednika na poljubno izbrani verigi, nato pa krajšo vejo (ponavadi) opustijo, kot je razvidno iz slike 3.1.

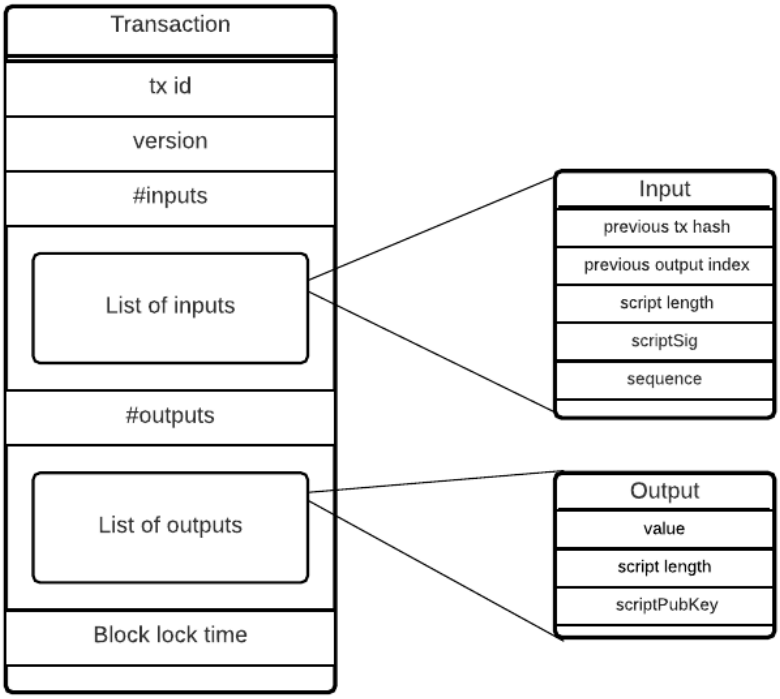


Slika 3.1: Zelen blok predstavlja izvorni blok. Črna veriga blokov je najdaljša, zato bo večina vozlišč iskala naslednika zadnjemu črnemu bloku [39].

Čas, potreben za dodajanje bloka v verigo, je v povprečju 10 minut, na podlagi tega pa se vsaka dva tedna tudi prilagodi zahtevnost uganke – število vodilnih ničel, ki jih zahtevamo v zgoščeni vrednosti.

3.1.2 Transakcije

Osnovo Bitcoinovega ekosistema predstavljajo transakcije. Vsako transakcijo sestavljajo metapodatki ter seznama *vhodov* (angl. Input) in *izhodov* (angl. Output). Med metapodatki najdemo ID transakcije, njeno zgostitev, števca (angl. counter) vhodov in izhodov, verzijo (angl. version) in druge.



Slika 3.2: Bitcoinova transakcija (na kratko tx)

Transakcijski vhod (angl. Transaction Input, TxIn) je kazalec na izhod prejšnje transakcije. Pove nam, kje in kako dobimo vsoto, ki jo želimo vključiti v transakcijo. Vsebuje zgostitev prejšnje transakcije, indeks izhoda, ki ga želimo porabiti, sekvenčno številko (mehanizem, ki omogoča dodajanje transakcije v blok, kljub temu da časovna ključavnica še ni potekla) ter prvi del skripte `ScriptSig`. Ta je namenjen preverjanju, ali izhod, ki ga želimo porabiti, res pripada nam. Da je transakcija lahko veljavna, mora biti vsak vhod podpisan z zasebnim ključem, ki ustreza javnemu ključu izhoda, ki ga porablja.

Transakcijski izhod (angl. Transaction Output, TxOut) predstavlja podatek o vrednosti, ki jo želimo prenesti. `ScriptPubKey` je drugi del skripte in poda navodila o tem, kaj je potrebno, da bomo zgoraj omenjeno vrednost

lahko porabili. Vsota izhodov neke transakcije mora biti manjša ali enaka vsoti vrednosti, ki jih skupaj v transakcijo prinesejo vhodi. V primeru, da je manjša, dobi razliko rudar v obliki transakcijske provizije.

Bitcoin podpira več vrst transakcij, ki se ločijo na podlagi tipa naslova, na katerega pošiljamo sredstva [1]:

"Plačaj-na-zgostitev-javnega-naslova" (angl. Pay-to-PubkeyHash, P2PKH) transakcija nakaže sredstva na navaden Bitcoin naslov, ki je zgostitev javnega ključa prejemnika. Preprosta skripta, ki preverja veljavnost, zahteva le javni ključ in podpis tistega, ki želi izhod porabiti (vključiti v novo transakcijo).

"Plačaj-na-zgostitev-naslova-skripte" (angl. Pay-to-ScriptHash, P2SH) transakcija nakaže sredstva na naslov skripte. Ta specificira dodatna navodila, katerih izpolnitev je zahtevana za porabo sredstev. Ta navodila nato zgostimo in da lahko sredstva porabimo v neki transakciji, mora biti zgostitev skripte, ki jo vključimo v vhod transakcije, enaka vključenemu podatku o zgostitvi prejšnje transakcije (glej sliko 3.2).

Večpodpisna (angl. Multisig) transakcija predstavlja enega od načinov uporabe P2SH. Ponavadi večpodpisni naslov označimo z m-od-n (angl. m-of-n). To označuje, da smo v transakcijo vključili n javnih ključev in je za porabo sredstev, ki se nahajajo na naslovu, potrebnih m ustreznih podpisov. Kot bomo spoznali v naslednjem poglavju, bliskovito omrežje uporablja 2-od-2 večpodpisne naslove. To pomeni, da skripta za izvajanje potrebuje podpisa lastnikov obeh javnih ključev.

Posebno vrsto predstavlja *transakcija za generiranje kovancev* (angl. Coinbase Transaction) [7], ki jo v blok vključi rudar in si z njo izplača nagrado za uspešno dodan blok.

3.2 Litecoin

Litecoin (LTC) je vejitev Bitcoinove verige iz leta 2011. Predstavlja eno od mnogih alternativnih valut. Na kratko ga bomo opisali, saj je poleg Bitcoina trenutno edina kriptovaluta z Lightning podporo [23]. Kreator Litecoina, Charlie Lee je njegov velik podpornik in pričakuje, da bo imel Litecoin zaradi nižjih transakcijskih provizij in hitrejših transakcij pomembno vlogo kot vstopna točka v bliskovito omrežje.

V Litecoin omrežju je povprečni čas za generiranje bloka 2.5 minute, kar je rezultat uporabe drugačnega dokaza o delu. Kot alternativo Bitcoinovemu algoritmu PoW (opisanemu v razdelku 2.3.4), ki ima za osnovo SHA-256 zgoščevalno funkcijo, uporablja za dokaz o delu pomnilniško zahtevnejši algoritem Scrypt [24]. Za osnovo vzame računsko precej enostaven problem in tako omogoči rudarjenje vsakomur, ki ima računalnik in povezavo z internetom. Namenske naprave za rudarjenje Litecoina bi potrebovale večje količine hitro dostopnega spomina, kar pa jih naredi precej dražje od Bitcoinovih. To med drugim preprečuje vedno večjo centralizacijo, ki smo ji priča pri Bitcoinu.

Poglavje 4

Bliskovito omrežje

Bliskovito omrežje (angl. Lightning Network) je plast nad Bitcoin protokolom, ki omogoča veliko količino plačil manjših vsot (mikroplačil) z nizkimi zakasnitvami, brez potrebe po zaupanja vrednem posredniku [2]. Bitcoinov problem mikroplačil rešuje z uporabo pametnih pogodb. Sredstev, ki jih prenašamo, nima nikoli v ekskluzivni lasti nobeden od posrednikov, kar močno zmanjša tveganje tretje osebe in transakcijske provizije. Bistvena tehnologija, ki jo uporablja bliskovito omrežje, je dvostransko soglasje v obliki plačilnega kanala. V grobem je postopek delovanja bliskovitega omrežja sledeč:

1. sodelujoča pošljeta v kanal začetno vsoto bitcoinov in objavita v verigi blokov večpodpisno transakcijo z dogovorjeno razporeditvijo sredstev,

2. vsaka sprememba razdelitve sredstev se zgodi v obliki nove transakcije, ki porabi sredstva v kanalu za nakazilo v novem razmerju, za transakcijo pa je potrebno soglasje obeh strani.

Objavljena začetna transakcija odpre plačilni kanal, hkrati z njo pa se ustvari tudi vračilna transakcija, ki omogoča vrnitev originalnih pologov sodelujočima. Mikrotransakcije tako v osnovi predstavljajo zaporedje dvojnih porab sredstev, nakazanih v začetni transakciji, ki samo posodobijo razmerje sredstev, in niso objavljene v verigi, dokler se z njimi strinjata obe strani. Če se eden

od sodelujočih z zadnjo transakcijo ne strinja, ali pa želi sredstva porabiti, lahko v vsakem trenutku brez strinjanja druge strani objavi v verigo zadnje stanje v kanalu, nad katerim je bilo doseženo soglasje, in s tem zapre kanal. V izogib goljufanju ima nepravilno objavljeno stanje enega od sodelujočih za posledico avtomatsko dodelitev vseh sredstev iz kanala nasprotni strani. Politika delovanja je precej podobna načelu *osebnega jamstva* (angl. fidelity bond), ki ga uporabljajo podjetja, da se zavarujejo pred poskusi oškodovanja. Če ena stran krši pogoje, specificirane za kanal, in lahko druga to dokaže v ustreznem časovnem roku, je upravičena do odškodnine. V našem primeru to predstavlja vsa sredstva v kanalu. Če sta obe strani pošteni, lahko sodelujeta neomejeno dolgo in kanal ostaja odprt.
V naslednjih sekcijah se bomo najprej na kratko sprehodili skozi zgodovino projekta Lightning in za lažje sledenje dodali slovarček novih pojmov, ki jih vpeljemo. Nato postopoma zgradimo mehanizem, ki bo omogočal plačevanje v omrežju kanalov, kjer večina transakcij nikoli ne pride v verigo blokov. V sekciji 4.3 predstavimo osnovni pojem plačilnega kanala. V sekcijah 4.4 in 4.5 spoznamo mehanizma pripisovanja krivde in prekliclјivih transakcij, ki poskrbita za sankcije v primeru poskusa goljufanja. V sekciji 4.6 vpeljemo še pogodbe z zgoščeno časovno ključavnico, ki končno omogočijo tudi uporabo v kanalih s posredniki, opisano v sekciji 4.7. Na koncu poglavja naredimo še kratek pregled do sedaj odkritih ranljivosti.

4.1 Zgodovina

O neki pravi zgodovini bliskovitega omrežja ne moremo govoriti, saj so že kriptovalute same novost zadnjih desetih let. Vseeno pa velja omeniti pomembnejše mejnike v njegovem razvoju do danes. Za začetke štejemo leto 2015, ko sta Joseph Poon in Thaddeus Dryja objavila predlog rešitve [29], ki opisuje razširljiv sistem takojšnjih plačil izven verige blokov. Na podlagi njunega predloga so bile objavljene specifikacije in več podjetij se je lotilo implementacije. V decembru 2017 so trije veliki razvijalci in ponudniki teh-

nologije verige blokov: Blockstream, Lightning Labs in ACINQ opravili serijo uspešnih testnih transakcij [14]. Za nov mejnik štejemo 8. januar 2018, ko je TorGuard kot prvi VPN ponudnik začel sprejemati plačila preko glavne mreže bliskovitega omrežja (angl. mainnet).
Z 18. januarjem je Blockstream odprl obdobje hitrih plačil s plačilnim sistemom za spletne trgovce Lightning Charge [30]. V marcu sta mu sledila Lightning Labs z lnd 0.4-beta verzijo [36] in ACINQ z namizno verzijo Eclair [40]. Vsi trije začetni produkti so bili namenjeni predvsem testiranju s strani razvijalcev, že v aprilu pa je Btcduke.com [27] kot prva Bitcoin – FIAT menjalnica začel sprejemati pologe in dvige preko glavne mreže bliskovitega omrežja. V maju je zgledu sledila še Bitcoin – zlato menjalnica Vaultoro.com [10].

4.2 Slovarček

V 4. poglavju vpeljemo kar nekaj novih pojmov. Spodaj je za lažje sledenje naveden seznam novih pojmov s kraticami, angleškimi prevodi in sekcijo, v kateri jih vpeljemo. Podrobnejša razlaga sledi v nadaljevanju.

kratica	**angleško**	**slovensko**	**sek.**
F	Funding Transaction	začetna transakcija, ki priskrbi sredstva za kanal	4.3.2
CT, C	Commitment Transaction	zavezujoča transakcija	4.3.1
RSMC	Revocable Sequence Maturity Contract	prekličljiva pogodba na podlagi zastaranja	4.5
D	Delivery Transaction	dostavna transakcija	4.5.1
RD	Revocable Delivery Transaction	prekličljiva dostavna transakcija	4.5.1
BR	Breach Remedy Transaction	transakcija za preprečevanje kršitev	4.5.2
ES	Excercise Settlement Transaction	transakcija za poravnavo	4.5.3

HTLC	Hashed Timelock Contract	pogodba z zgoščeno časovno ključavnico	4.6
HE	HTLC Execution	izvedbena transakcija	4.6.2
HED	HTLC Execution Delivery	izvedbena dostavna transakcija	4.6.2
HT	HTLC Timeout	transakcija za potek časovne omejitve	4.6.2
HTD	HTLC Timeout Delivery	dostavna transakcija za potek časovne omejitve	4.6.2
HERD	HTLC Execution Revocable Delivery	izvedbena preklicljiva dostavna transakcija	4.6.2
HTRD	HTLC Timeout Revocable Delivery	dostavna transakcija za potek časovne omejitve	4.6.2
$\Delta cltv$	`CheckLockTimeVerify` expiry delta	min. razlika med potekoma časovnih ključavnic v kanalu	4.7.1

4.3 Plačilni kanali

Plačilni kanali niso novost, ki bi jo vpeljal projekt Lightning. Prvič se pojavijo že v letu 2013, ko sta Matt Corallo in Mike Hearn implementirala podporo enosmernih plačilnih kanalov v BitcoinJ [46]. Uporabnost le-teh je bila precej omejena, podpirali so le transakcije med dvema vozliščema, ki sta bili neposredno povezani s kanalom. To predstavlja prednost v primerih ponavljajočih plačil med istima uporabnikoma, za potrebe vsakdanjega plačevanja pa je precej neuporabno. Bliskovito omrežje uporablja transakcije s časovnimi ključavnicami (ki jih bomo vpeljali v nadaljevanju) in žetone, ki omogočajo povezavo dvostranskih kanalov v omrežje, kjer lahko plačila potujejo preko več kanalov brez potrebe po zaupanju vozliščem na poti. Kot pri večini poslov se potreba po globalnem soglasju pojavi le v primeru, ko eden od sodelujočih ne upošteva pravil in rešitev spora zahteva posredovanje

višje avtoritete (tj. verige blokov).

4.3.1 Enosmerni kanali

Kot pove že ime, je *enosmerni kanal* namenjen transakcijam med dvema osebama, ki potekajo izključno v eno smer – od kreatorja kanala do prejemnika. Če želimo transakcije v nasprotni smeri, potrebujemo za to nov kanal. Kot omenjeno v začetku poglavja, je rešitev uporabna za serije ponavljajočih se transakcij med dvema uporabnikoma. Če želimo uporabo posplošiti, število kanalov, potrebnih za poslovanje, kaj hitro naraste, kar v praksi preprosto ni izvedljivo. Postopek odpiranja kanala je precej preprost in za pošiljateljico Ano in prejemnika Bojana izgleda takole:

1. Ana kreira začetno transakcijo z želenim izhodom Bojanu[1].
2. Ana pošlje Bojanu ID začetne transakcije, izhod in količino sredstev, ki jih želi prenesti.
3. Bojan ustvari vračilno transakcijo, ki porabi začetni izhod in ga nakaže nazaj na Anin naslov, ter jo podpiše. Ta transakcija je uporabna šele po poteku določenega časovnega okvirja.
4. Bojan pošlje Ani vračilno transakcijo.
5. Ana podpiše začetno transakcijo.

Sedaj lahko Ana opravi neomejeno število nakazil – podpiše neomejeno število zavezujočih transakcij za posodobitev stanja. *Zavezujoča transakcija* (angl. Commitment Transaction, C) ima 2 izhoda. Enega, ki del sredstev nameni Bojanu, in drugega, ki preostala sredstva pošlje nazaj na njen naslov. Bojan ima vedno možnost porabiti prejšnji izhod, če mu v novi transakciji Ana dodeli manj sredstev. Bojan lahko kadarkoli zapre kanal, tako da objavi

[1] V tem in naslednjih primerih privzamemo, da so vhodi, s katerimi Ana priskrbi sredstva za transakcijo, veljavni in predstavljajo celotno količino sredstev v kanalu. Katere neporabljene izhode prejšnjih transakcij je uporabila, tu ni bistveno.

zadnjo potrditveno transakcijo in s tem prenese denar na svoj račun. Če tega ne naredi v določenem časovnem okvirju, lahko Ana sredstva vzame nazaj z objavo vračilne transakcije.

4.3.2 Obojesmerni kanali

Bliskovito omrežje vpelje nove, posplošene plačilne kanale, ki se po predlagateljih rešitve imenujejo tudi Poon-Dryja kanali. Ti uporabljajo skupno *začetno transakcijo* (angl. Funding Transaction, F), ki zagotovi sredstva za kanal, nato pa simetričen par transakcij za posodabljanje stanja v kanalu. Obojesmerni kanal v osnovi deluje na podoben način kot enosmerni. V primeru uporabnikov Ane in Bojana iz prejšnjega razdelka je kreirana začetna transakcija, v kateri eden ali oba priskrbita sredstva za kanal. Obe strani ustvarita vhode in, na podlagi skupne količine sredstev, izhode za začetno transakcijo, vendar je še ne podpišeta. Izhod te transakcije je 2-od-2 večpodpisna skripta, ki razporeja skupno vsoto sredstev v kanalu.
Za vsako posodobitev razmerja sredstev je potrebno soglasje. Kot prej ima vsaka transakcija 2 izhoda, enega za Ano in enega za Bojana, le da izhod samemu sebi tukaj zahteva podpis obeh. Ana in Bojan zato vedno najprej ustvarita vračilni transakciji, ki porabita v tem trenutku še neobstoječo, zavezujočo transakcijo, in šele nato podpišeta slednjo.
Zaradi potrebe po porabi še neobstoječe transakcije z mehko vejitvijo uvedemo zastavico `SIGHASH_NOINPUT`. Za porabo neke transakcije potrebujemo ID le-te. Podpisi *predhodne* (angl. parent) transakcije predstavljajo del ID Bitcoinove transakcije; brez zastavice transakcije dobijo ID šele, ko so podpisani vsi njihovi vhodi. Če hočeta porabiti transakcijo, bi si morala Ana in Bojan izmenjati podpisa začetne transakcije, kar pa bi omogočilo enemu od njiju, da transakcijo objavi, še preden so kreirani otroci (zavezujoče transakcije). Zastavica `SIGHASH_NOINPUT` obide ta problem, tako da dovoli porabo otrok brez podpisa vhodov. Zaporedje operacij je tako izvedeno z naslednjimi koraki [29]:

1. kreiranje starša (začetne transakcije),

2. kreiranje otrok (zavezujočih transakcij in njihovih porab),
3. podpis otrok,
4. izmenjava podpisov za otroke,
5. podpis starševske transakcije,
6. izmenjava podpisov za starševsko transakcijo,
7. objava začetne transakcije v verigo blokov.

Vse do 6. koraka začetna transakcija ne more biti objavljena v verigi, kar prepreči možnost zadrževanja zaklenjenih sredstev v primeru, da Ana in Bojan nočeta sodelovati.
Sedaj lahko Ana in Bojan ustvarita neomejeno število zavezujočih transakcij, ki jih ne objavljata v verigi blokov, dokler ne želita zapreti kanala pri neki porazdelitvi. To storita z objavo trenutne zavezujoče transakcije.

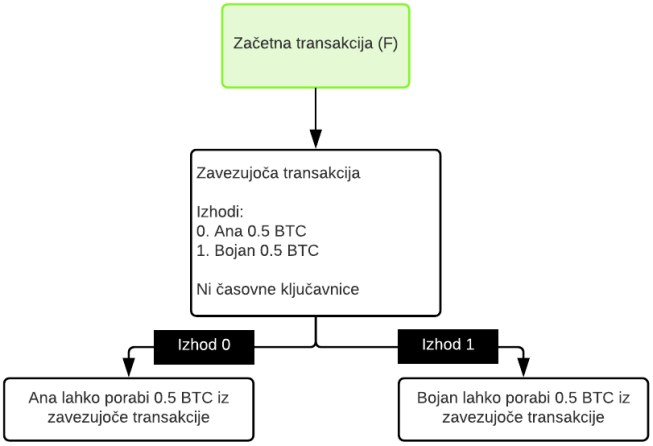
4.4 Pripisovanje krivde

V naivni implementaciji ima začetna transakcija predvidenega enega naslednika: zavezujočo transakcijo s po enim izhodom za Ano in Bojana, ki porabi sredstva s tem, da jih nakaže v dogovorjenem razmerju[2] (glej sliko 4.1[3]). Problem nastane v naslednjem koraku, ko ena od strani želi posodobiti razmerje sredstev v kanalu. Če se oba strinjata, nova zavezujoča transakcija odraža posodobljeno stanje. Ker pa sta podpisala tako prejšnjo kot tudi novo transakcijo in izmenjala podpisa za obe, imamo naenkrat dve transakciji, ki sta lahko objavljeni. Vsakemu udeležencu je v interesu, da objavi tisto od transakcij, ki je bolj ugodna zanj, kar pa vodi v nemudno zaprtje

[2]V tem in naslednjih primerih zaradi preglednosti privzamemo za skupno vsoto v kanalu 1 BTC, ki je bil za časa pisanja diplomske naloge vreden 7290$. V praksi se (vsaj za sedaj) za transakcije v bliskovitem omrežju uporabljajo veliko manjši zneski.

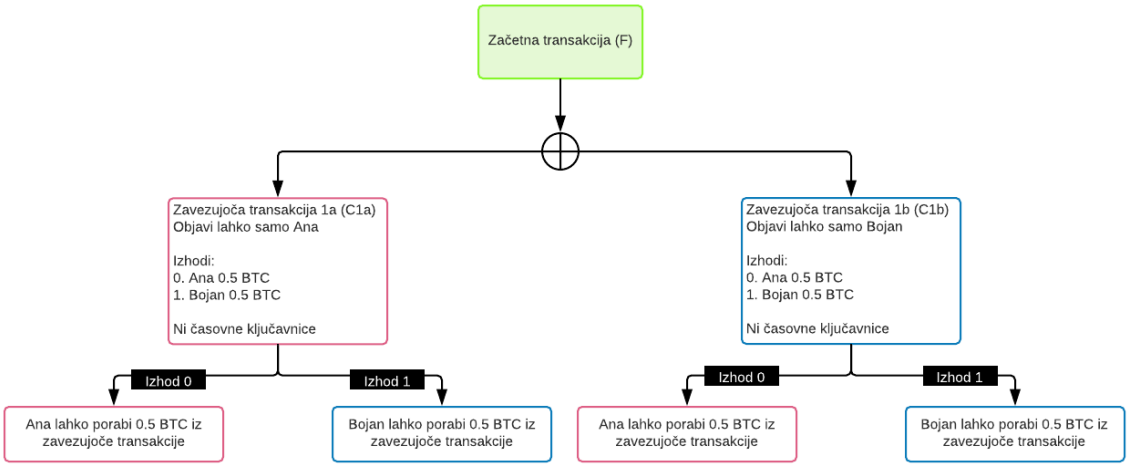
[3]Sheme v tem poglavju so prirejene po Poon in Dryja [29].

kanala in dejansko krajo sredstev.



Slika 4.1: Zavezujoča transakcija porabi začetno transackcijo. Vsak lahko svoj izhod porabi takoj, ko je zavezujoča transakcija objavljena v verigi.

Pravkar opisano krajo skušamo preprečiti z uvedbo pravila, da lahko vsaka stran objavi le zadnjo transakcijo, prejšnje pa so preklicane. Po sprejetju soglasja nad posodobljeno transakcijo morata sodelujoča objaviti svoja zasebna ključa prejšnje. V primeru, da Ana podpiše preklicano potrditveno transakcijo, je prekršila pravila. Želimo doseči, da lahko Bojan takoj porabi svoj izhod, poleg tega pa lahko generira še transakcijo, ki bo porabila izhod, namenjen Ani.
Za ugotavljanje, katera stran je objavila staro transakcijo, tako uvedemo sistem s pari zavezujočih transakcij. Vsaka stran ima svojo, unikatno verzijo transakcije. Hkrati je lahko objavljena le ena, saj obe porabita isto začetno transakcijo. Ana ima transakcijo, katere vhodi so predhodno podpisani z Bojanovim zasebnim ključem, in Bojan transakcijo, katere vhode je predhodno podpisala Ana. Vsak lahko objavi le svojo verzijo, in sicer tako, da transakcijo podpiše še s svojim zasebnim ključem.



Slika 4.2: Začetna transakcija dobi dva naslednika, različni verziji zavezujoče transakcije. Transakcije v barvi maline lahko objavi in porabi Ana, modre pa Bojan. Hkrati je lahko objavljena le ena od zavezujočih transakcij 1a in 1b (oznaka $\oplus$).

4.5 Prekličljive zavezujoče transakcije

Če je prejšnji korak zagotovil, da je v vsakem trenutku jasno, katera stran je transakcijo objavila v verigo, sedaj potrebujemo še mehanizem, ki poskrbi za preverjanje, ali je objavljena transakcija res zadnja (tj. veljavna), in izvedbo sankcij v primeru, da to ni.

Nova, *prekličljiva zavezujoča transakcija* (angl. Revocable Commitment Transaction) se od tistih, ki smo jih spoznali doslej, razlikuje po tem, da je na mestu izhoda, namenjenega lastniku, t. i. *prekličljiva pogodba na podlagi zastaranja* (angl. Revocable Sequence Maturity Contract, RSMC). Zagotoviti želimo, da s kreiranjem nove zavezujoče transakcije stare transakcije postanejo neveljavne. Prekličljiva pogodba odredi naslednje:

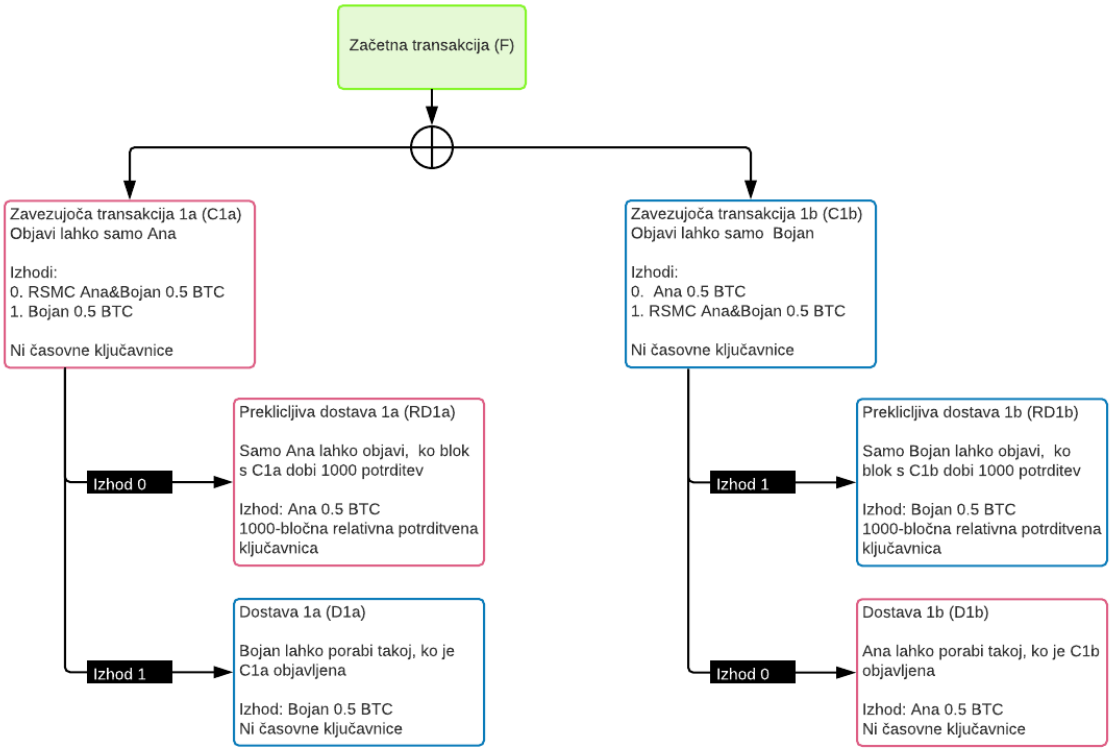
1. obe strani nakažeta sredstva na naslov skripte z določeno potrditveno ključavnico (številom potrditev, pred katerim izhodov ni možno porabiti); ta sredstva predstavljajo višino prekličljivega izhoda,

2. sodelujoča lahko ne objavita pogodbe; po zakasnitvenem obdobju lahko v tem primeru oba porabita sredstva takoj,

3. če nobeden ni objavil transakcije, lahko prekličeta nakazila samo, če se oba strinjata s kreiranjem nove transakcije, ki sredstva razdeli v drugem razmerju,

4. če si sredstev po objavi nista razdelila po novem razmerju, lahko katera od strani porabi katerega od prej preklicanih razmerij.

Ko se eden odloči, da bo objavil zavezujočo transakcijo v verigo blokov, bo izhod, namenjen drugemu, postal veljaven takoj. Tisti, ki je transakcijo objavil, pa mora počakati, da se izteče *potrditvena ključavnica*, tj. število blokov, ki morajo biti dodani, preden bo lahko dostopal do svojega dela sredstev. Potrditveno ključavnico določimo z dodelitvijo večje sekvenčne številke izhodu z RSMC in tako poskrbimo, da ta postane veljaven šele, ko dobi blok dovolj potrditev. Čakanje, da blok dobi določeno število potrditev, služi kot časovno okno za preverjanje, da zavezujoča transakcija ni ena izmed že preklicanih.



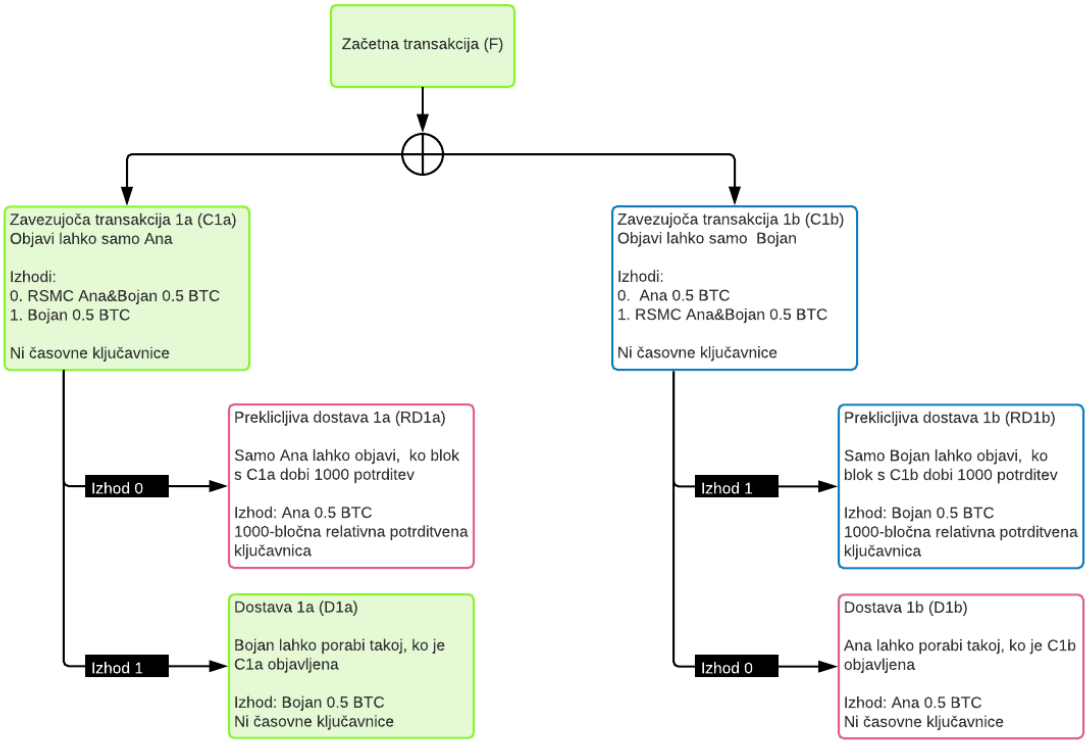
Slika 4.3: Zavezujoči transakciji (C1a in C1b) imata na mestih izhoda samemu sebi RSMC. V našem primeru zahtevamo, da dobi naš blok 1000 potrditev, preden postane veljavna katera od prekličljivih dostav (RD1a, RD1b).

Vnaprej podpisana transakcija se lahko porabi šele, ko je starševska transakcija dobila za sekvenčno številko potrditev. Za preklic podpisane transakcije se morata obe strani strinjati, da ustvarita novega otroka, tj. transakcijo, ki bo imela na mestu sekvenčne številke `MAX_INT`[4]. Število dovoljuje porabo le-te kadarkoli. Nova transakcija ima tako prednost pred prekličljivo, ki bo lahko objavljena šele po koncu zakasnitvenega obdobja.

[4]Največje število `0xffffffff` iz obsega nepredznačenih celih števil (angl. unsigned integer) označuje konstanta `MAX_INT`.

4.5.1 Objava in poraba sredstev iz kanala

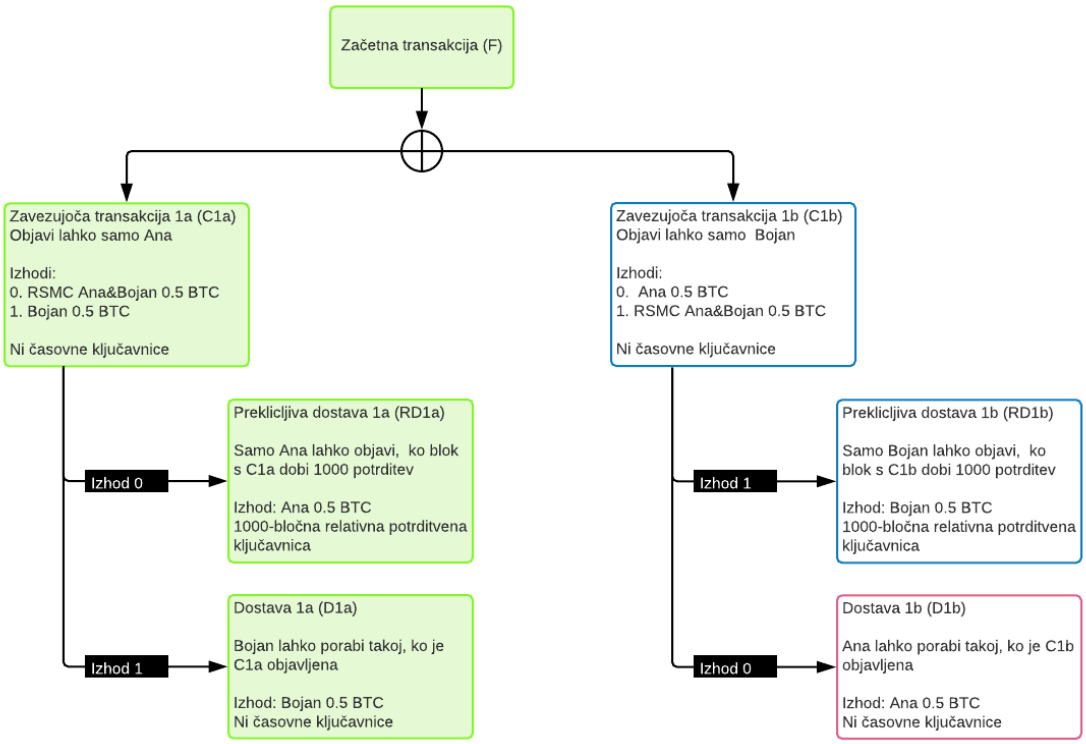
Vzemimo pod drobnogled primer, ko Ana pravilno objavi svojo zadnjo zavezujočo transakcijo. *Dostavna transakcija* (angl. Delivery Transaction) Bojanu je preprost izhod, ki ni vezan na RSMC in ga Bojan lahko porabi takoj po objavi. Na mestu izhoda samemu sebi je *prekličljiva dostavna transakcija* (angl. Revocable Delivery Transaction, RD) Ani. RSMC med Ano in Bojanom zagotavlja Ani možnost porabe njenega dela sredstev, ko bo blok dobil določeno število potrditev.



Slika 4.4: Ana objavi svojo zavezujočo transakcijo (C1a). Ko C1a pride v verigo blokov, lahko Bojan takoj porabi svoj izhod – dostavno transakcijo D1a, medtem ko Ana čaka, da dobi blok s transakcijo C1a 1000 potrditev.

V trenutku, ko zavezujoča transakcija vstopi v verigo blokov, vstopi v njo tudi izhod, namenjen Bojanu, ki je veljaven takoj. Izhod, namenjen Ani, je

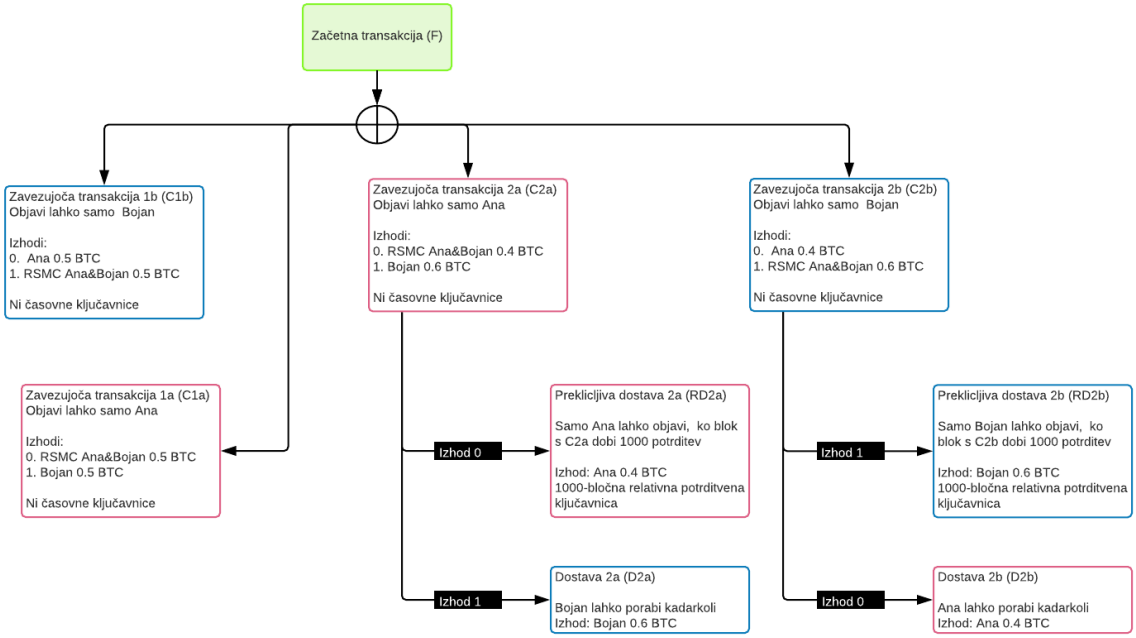
še zaklenjen pod pogodbo. Ana ga lahko poskusi porabiti takoj, a je neveljaven do izteka potrditvene ključavnice (glej sliko 4.4). Ker je bila pravilno objavljena zadnja transakcija in je Bojan ni mogel razveljaviti, lahko sedaj tudi Ana objavi svojo preklicljivo dostavo ter s tem dokončno zapre kanal.



Slika 4.5: Blok s transakcijo C1a je dobil 1000 potrditev. Transakcija RD1a je postala veljavna in sedaj lahko tudi Ana porabi svojega 0.5 BTC.

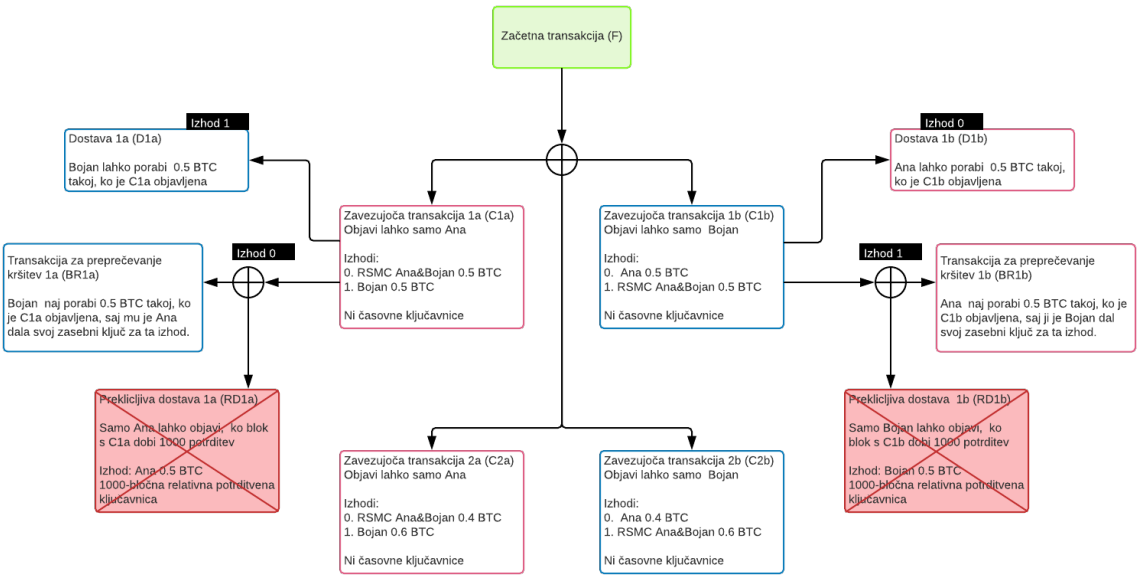
4.5.2 Posodabljanje stanja v kanalu

V primeru, da se namesto za objavo zadnje transakcije Ana in Bojan odločita za posodobitev stanja v kanalu, ustvarita nov par zavezujočih transakcij. Oba podpišeta transakciji in izmenjata podpisa. V tem trenutku sta veljavna tako stari kot tudi novi par transakcij, vsak od udeleženih pa lahko objavi po eno transakcijo iz vsakega para.



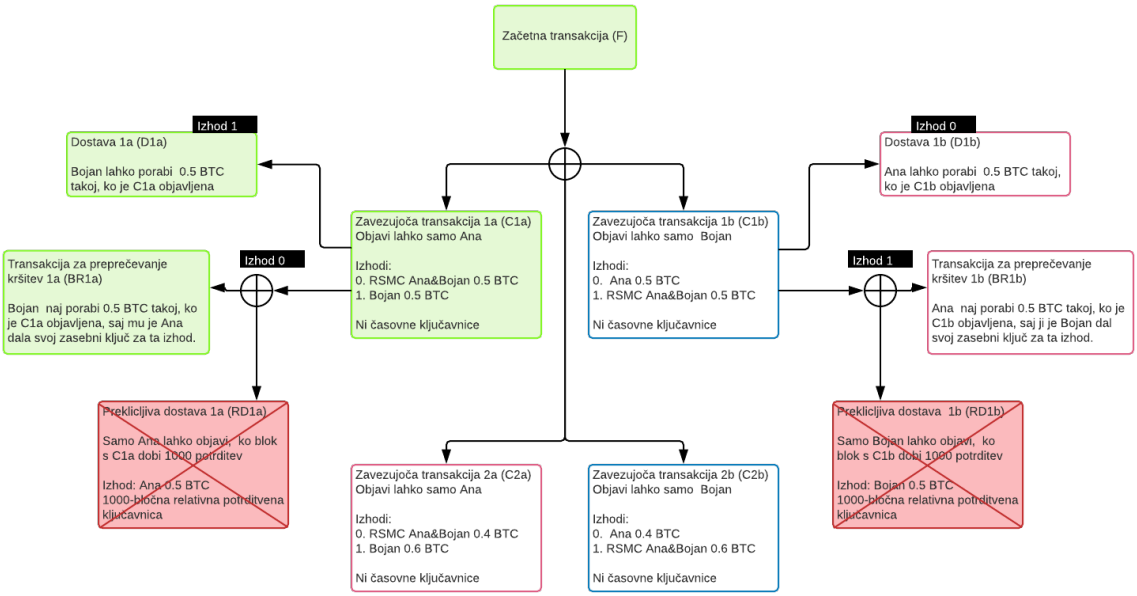
Slika 4.6: Stanje pred razveljavitvijo starega para transakcij. Začetna transakcija ima 4 otroke: prvi par transakcij (C1a, C1b) in drugi par (C2a, C2b). Ana lahko objavi transakciji C1a in C2a, Bojan pa C1b in C2b.

Hkrati je lahko objavljena le ena od zavezujočih transakcij na sliki 4.6, saj vse porabijo isto začetno transakcijo (veljavnost transakcije se preverja pred vključitvijo v blok). Za razveljavitev starega para morata oba podpisati *transakcijo za preprečevanje kršitev* (angl. Breach Remedy Transaction, BR), ki nadomesti prekličljivo dostavno transakcijo (glej rdečo prečrtano transakcijo na sliki 4.7). Če je transakcija RD namenila neko količino sredstev tistemu, ki je objavil zavezujočo transakcijo, jih transakcija BR nakaže drugemu. Ana in Bojan izmenjata podpisana preklica (transakciji BR), ki porabita staro zavezujočo transakcijo in vsa sredstva nakažeta nasprotni strani. S tem se vsak obveže, da ne bo objavljal starih transakcij, saj v nasprotnem primeru izgubi vsa sredstva.



Slika 4.7: Izhoda s prekličljivo dostavno transakcijo RD1 (na sliki prečtrana) nadomestita transakciji za preprečevanje kršitev pogodbe.

Če Ana v nasprotju s pravili objavi staro, preklicano transakcijo, bo Bojan lahko pobral vsa sredstva iz kanala. Bojan ima čas za objavo transakcije za preprečevanje kršitev BR1a do poteka potrditvene ključavnice. Če to okno zamudi, postane veljavna Anina prekličljiva dostava RD1a. Sedaj lahko Bojan še vedno kadarkoli objavi BR1a, lahko pa ga prehiti Ana in objavi svojo prekličljivo dostavno transakcijo. Prav slednje je razlog, da mora vsak spremljati, če je drugi v verigo objavil preklicano transakcijo, in na to čim hitreje odreagirati. Alternativno lahko izbere tretjo osebo, da verigo spremlja namesto njega. Ta mora za to posedovati transakcijo za preprečevanje kršitev. Za spodbudo njene poštenosti pa dobi delež izhoda v primeru, da prepreči poskus goljufanja in kraje sredstev nasprotne strani.

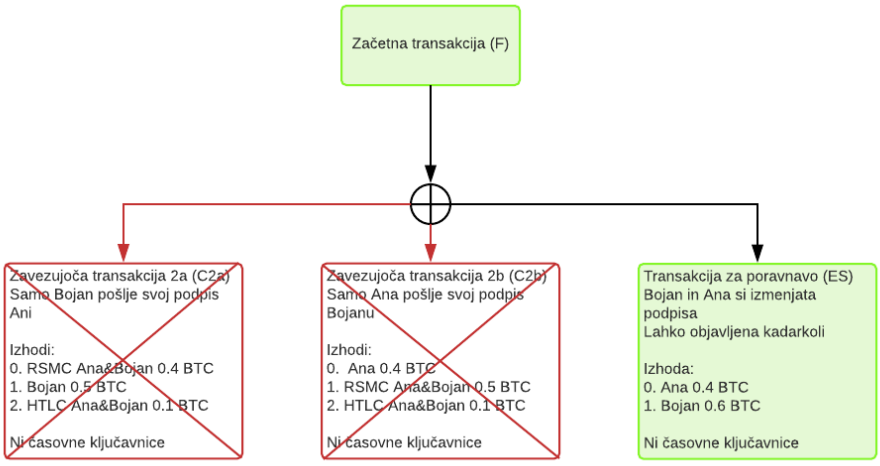


Slika 4.8: Ana je objavila staro transakcijo (C1a). Bojan lahko takoj porabi svoj izhod (D1a), poleg tega pa ima Anino podpisano transakcijo za preprečevanje kršitev in tako lahko porabi tudi 0.5 BTC, ki bi pripadal Ani, če bi bila C1 še veljavna.

4.5.3 Sporazumno zapiranje kanala

Pri objavi zavezujoče transakcije mora tisti, ki jo je objavil, čakati do izteka potrditvene ključavnice. Če potegnemo vzporednice z navadnimi pogodbami, objava v verigo predstavlja reševanje spora na sodišču v primeru, da bi ena ali obe strani želeli goljufati. V večini primerov pričakujemo pošteno poslovanje obeh strani in se zato želimo izogniti nepotrebnemu čakanju in stroškom. Kot alternativo objavi zavezujoče transakcije zato uvedemo *transakcijo za poravnavo* (angl. Exercise Settlement Transaction, ES). Slednja ne predstavlja nobene novosti. Je preprosta transakcija brez RSMC in ima, kot vidimo na sliki 4.9, le navadna izhoda, ki sredstva v kanalu delita v razmerju zadnje zavezujoče transakcije. Dosežemo enako stanje, kot bi ga z objavo zadnje zavezujoče transakcije. Hkrati se izognemo obveznemu čakanju, da poteče

potrditvena ključavnica, saj sta oba izhoda veljavna takoj.



Slika 4.9: Ana in Bojan imata podpisa transakcij drug od drugega. Kadarkoli lahko objavita transakcijo za poravnavo, ki sredstva razdeli v nazadnje določenem razmerju. S tem sporazumno zapreta kanal in porabita vsak svoja sredstva.

4.6 Pogodbe z zgoščeno časovno ključavnico

Če so rešitve, opisane v prejšnjih sekcijah, poskrbele za uspešno poslovanje med dvema vozliščema, ki sta neposredno povezani s kanalom, pa hitro ugotovimo, da v omrežju Bitcoinovih razsežnosti direktni kanali med vsakim parom vozlišč niso praktično izvedljiva rešitev. Pojavi se potreba po varnem prenosu sredstev skozi kanale z več *posredniki* (angl. hops) do končnega cilja. *Pogodba z zgoščeno časovno ključavnico* (angl. Hashed Timelock Contract, HTLC) v osnovi predstavlja način za prenos sredstev preko več posrednikov. Sistem deluje brez potrebe po zaupanju med posredniki, vsak člen v kanalu mora naslednjemu dokazati, da pozna *prasliko* R neke znane vrednosti H, kar služi kot zagotovilo, da je opravičen do sredstev. Pogoji pogodbe z zgoščeno ključavnico za Ano in Bojana, ki želita imeti odprt kanal, so po Poon&Dryja[29] sledeči:

1. če Bojan v nekem časovnem oknu priskrbi prasliko R znane zgoščene vrednosti H, bo Ana plačala Bojanu dogovorjeno vsoto,

2. če je preteklo časovno okno, v katerem bi Bojan moral dokazati poznavanje vrednosti R, je prva točka neveljavna,

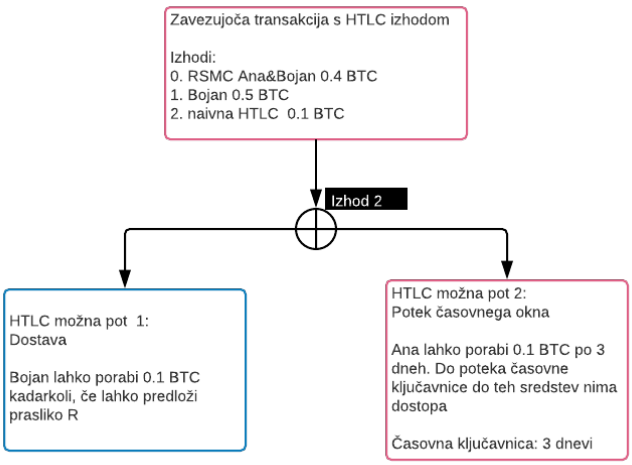
3. vsaka stran sme plačati drugi v obliki kakršne koli metode, ki je v skladu z dogovorom, in končati sodelovanje predčasno, če se obe strani strinjata,

4. v primeru poskusa goljufanja dobi poštena stran vsa sredstva v kanalu.

V nadaljevanju najprej opišemo preprostejše pogodbe z zgoščeno časovno ključavnico, ki niso ponujale možnosti preklica. Nato nadgradimo konstrukcijo, da omogoča preklic tudi brez objave v verigo blokov. V razdelku 4.6.3 povežemo rešitve, ki smo jih opisali do sedaj, in predstavimo celoten mehanizem delovanja transakcij v kanalu.

4.6.1 Neprekličljiva HTLC

Prvi poskus implementacije pogodb z zgoščeno časovno ključavnico je bila transakcija, ki se od navadne zavezujoče razlikuje po dodatnem HTLC izhodu. HTLC ponudi dve možnosti. V primeru, da je vrednost R predložena v dogovorjenem času, lahko prejemnik porabi vsoto, ki je bila zaklenjena pod pogodbo, tako da objavi dostavno transakcijo z vključeno vrednostjo R. Po poteku časovnega okna lahko plačnik objavi t. i. transakcijo za pretečeno časovno omejitev. Prav tako prejemnik še vedno lahko kadarkoli objavi svojo dostavno transakcijo, če pozna vrednost R.



Slika 4.10: Nova transakcija s HTLC izhodom.

Recimo, da Ana in Bojan želita posodobiti razmerje v kanalu iz 0.5 BTC za vsakega v 0.6 BTC za Bojana in 0.4 BTC za Ano (glej sliko 4.6). Del sredstev, ki ne menja lastnika, predstavljata izhoda 0 in 1, ki ostaneta enaka, kot v prejšnjih primerih. Vsota 0.1 BTC, ki jo želi Ana prenesti Bojanu, je v izhodu 2 s HTLC. Če Bojan lahko dokaže, da je upravičen do te vsote (predloži prasliko R), lahko porabi dostavno transakcijo. Drugače lahko Ana z objavo transakcije za potek časovne omejitve vzame svoja sredstva nazaj po poteku časovne ključavnice.

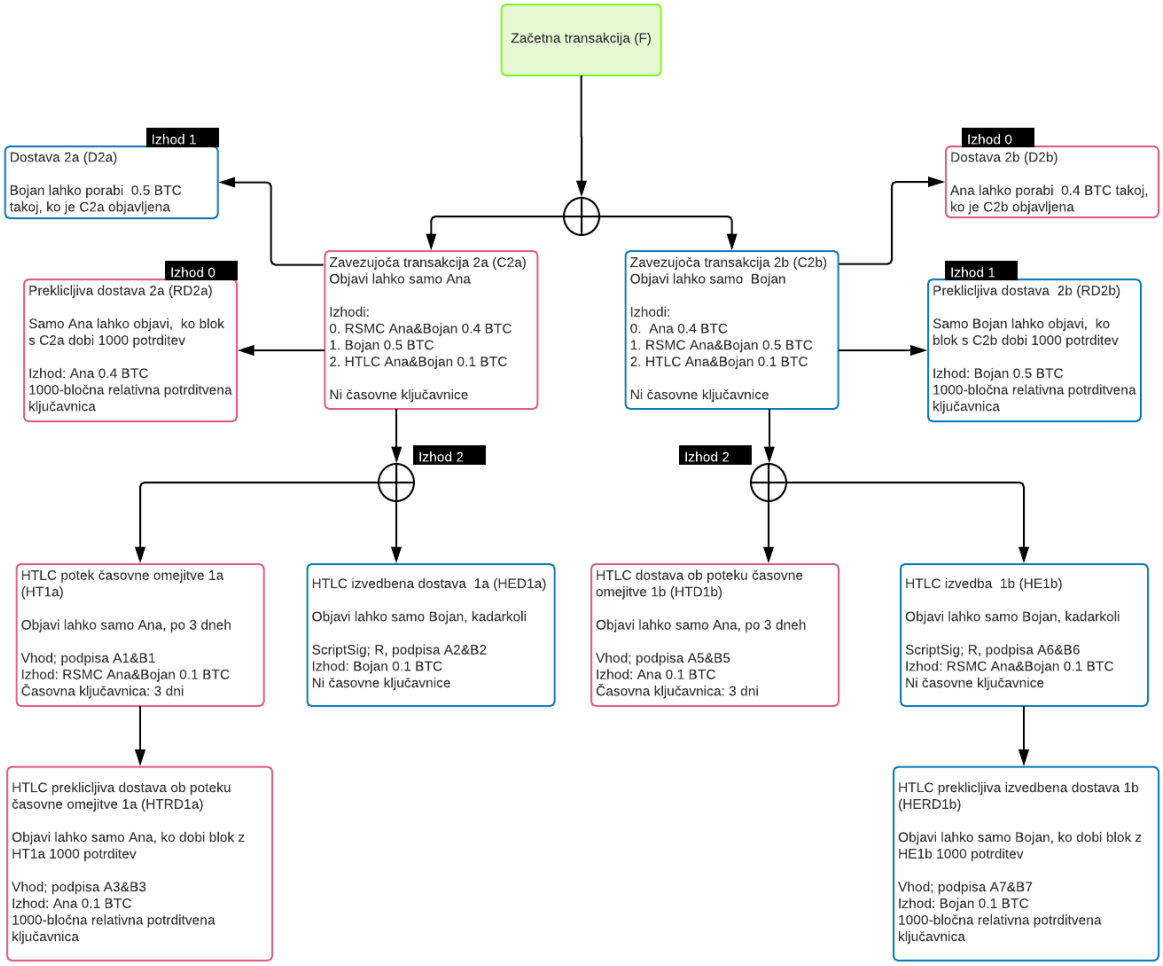
Preprost sistem se kaj hitro izkaže za pomanjkljivega, saj pride v primeru, ko Ana in Bojan ne sodelujeta, do situacije, podobne opisani v začetku sekcije 4.4. V primeru, da je objavljena stara zavezujoča transakcija, lahko ena od strani poskuša ukrasti sredstva, saj sta hkrati veljavni obe poti.

4.6.2 Zunanja prekliclјiva HTLC

Če so bile do sedaj vse transakcije paroma simetrične, se v tem trenutku stvari zapletejo. Struktura same zavezujoče transakcije se ne spremeni glede na neprekliclјivo iz prejšnjega razdelka. Kot vidimo na sliki 4.10, ostane del sredstev, ki jih vsak obdrži, znotraj prvih dveh izhodov. Del, ki menja

lastnika, je rezerviran pod HTLC. Spremeni se le sam HTLC izhod. Osredotočimo se na situacijo, ki nastane, ko se Ana ali Bojan odločita, da želita posodobiti stanje na sliki 4.10. Eden od njiju kreira novo zavezujočo transakcijo, in hitro pridemo do problema, saj je naenkrat veljaven tako nov, kot tudi star, tj. nepreklican par transakcij.

Da bo možno stare transakcije preklicati brez objave v verigi, moramo tako še v HTLC izhod vgraditi RSMC. S tem dobimo 4 možne poti, ki niso več paroma simetrične za obe strani. Že neprekliljivi HTLC je vpeljal dve novi vrsti transakcij, glede na vlogo tistega, ki jih objavi. *Izvedbene transakcije* lahko objavi prejemnik, da porabi sredstva, ki so mu bila nakazana, takoj ko razkrije prasliko R. *Transakcije ob poteku časovne omejitve* so namenjene plačniku, da lahko v primeru neodzivnosti prejemnika sredstva dobi nazaj. Z vključitvijo RSMC v izhod pa jih razdelimo še naprej, v navadno in preklicljivo dostavo.



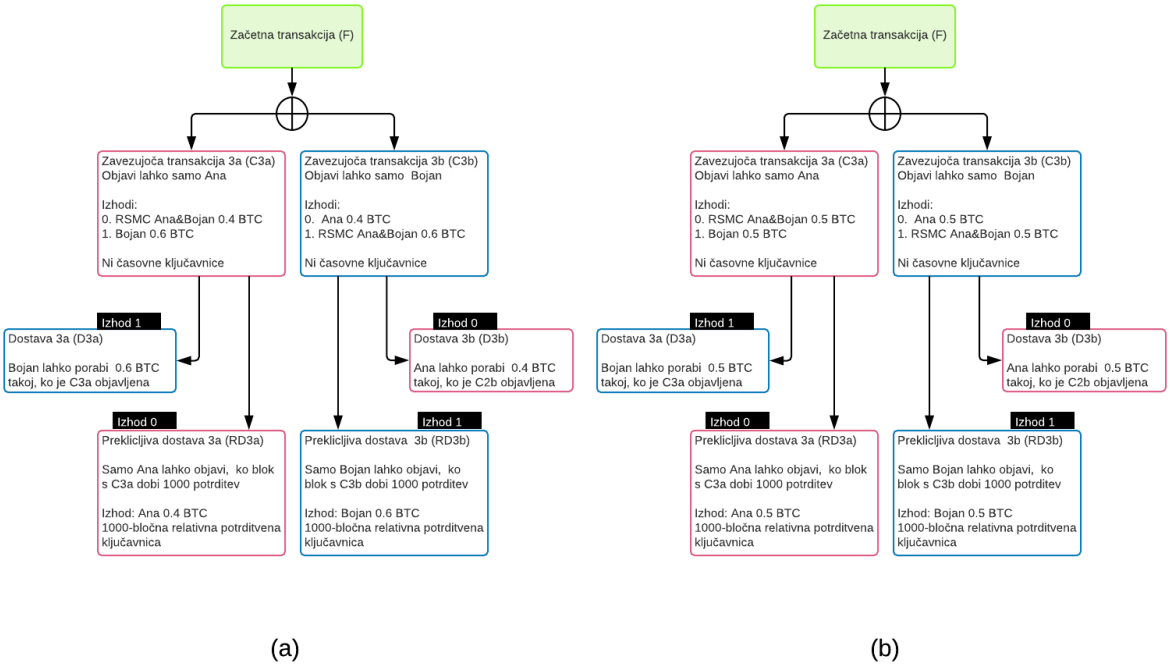
Slika 4.11: Zavezujoča transakcija C2 s prekličljivim HTLC izhodom

Kot vidimo na sliki 4.11, ima vsaka zavezujoča transakcija po dve poti za izhod 2 in para nista več simetrična. Vsaka stran podpiše transakcije od drugega in pri tem za vsako transakcijo uporabi drug zasebni ključ. Vsak tako potrebuje po štiri pare javnih in zasebnih ključev na zavezujočo transakcijo, torej skupno 8. Transakcije v barvi maline je podpisal Bojan in jih lahko objavi le Ana, modre pa je podpisala Ana in jih lahko objavi le Bojan. Para ključev $A4$, $B4$ in $A8$, $B8$, ki ju na sliki 4.11 ne vidimo, sta namenjena za potencialne nadomestne transakcije.

V primeru, da zavezujočo transakcijo objavi plačnik, lahko prejemnik t. i. *izvedbeno dostavno transakcijo* (angl. HTLC Execution Delivery, HED) porabi takoj, ko dokaže, da pozna vrednost R. Če tega ne stori do poteka časovne ključavnice, ga lahko pošiljatelj prehiti z objavo t. i. *transakcije za potek časovne omejitve* (angl. HTLC Timeout, HT). Izhod HT je RSMC med sodelujočima, ki zagotavlja plačniku vračilo nakazanih sredstev v primeru, da je bila njegova objavljena zavezujoča transakcija veljavna. Plačnik mora tako počakati, da dobi blok s transakcijo HT vnaprej določeno število potrditev. Nato lahko porabi vsoto, ki mu je bila vrnjena s t. i. *preklicljivo dostavno transakcijo* (angl. HTLC Timeout Revocable Delivery, HTRD).
Če zavezujočo transakcijo objavi prejemnik, plačnik ponovno čaka na potek časovne ključavnice. Prejemnik lahko, če pozna vrednost R, tudi v tem primeru že takoj objavi *izvedbeno transakcijo* (angl. HTLC Execution, HE), katere izhod je RSMC z večpodpisno skripto. Če je objavil veljavno zavezujočo transakcijo, mora sedaj počakati le še, da dobi blok s HE dovolj potrditev, da postane veljavna *preklicljiva izvedbena dostavna transakcija* (angl. HTLC Execution Revocable Delivery, HERD). V primeru, da prejemnik iz nekega razloga ni pravočasno predložil vrednosti R, postane veljavna t. i. *dostavna transakcija ob poteku časovne omejitve* (angl. HTLC Timeout Delivery, HTD). Sedaj ga lahko plačnik prehiti in z objavo le-te vzame prenesena sredstva nazaj.

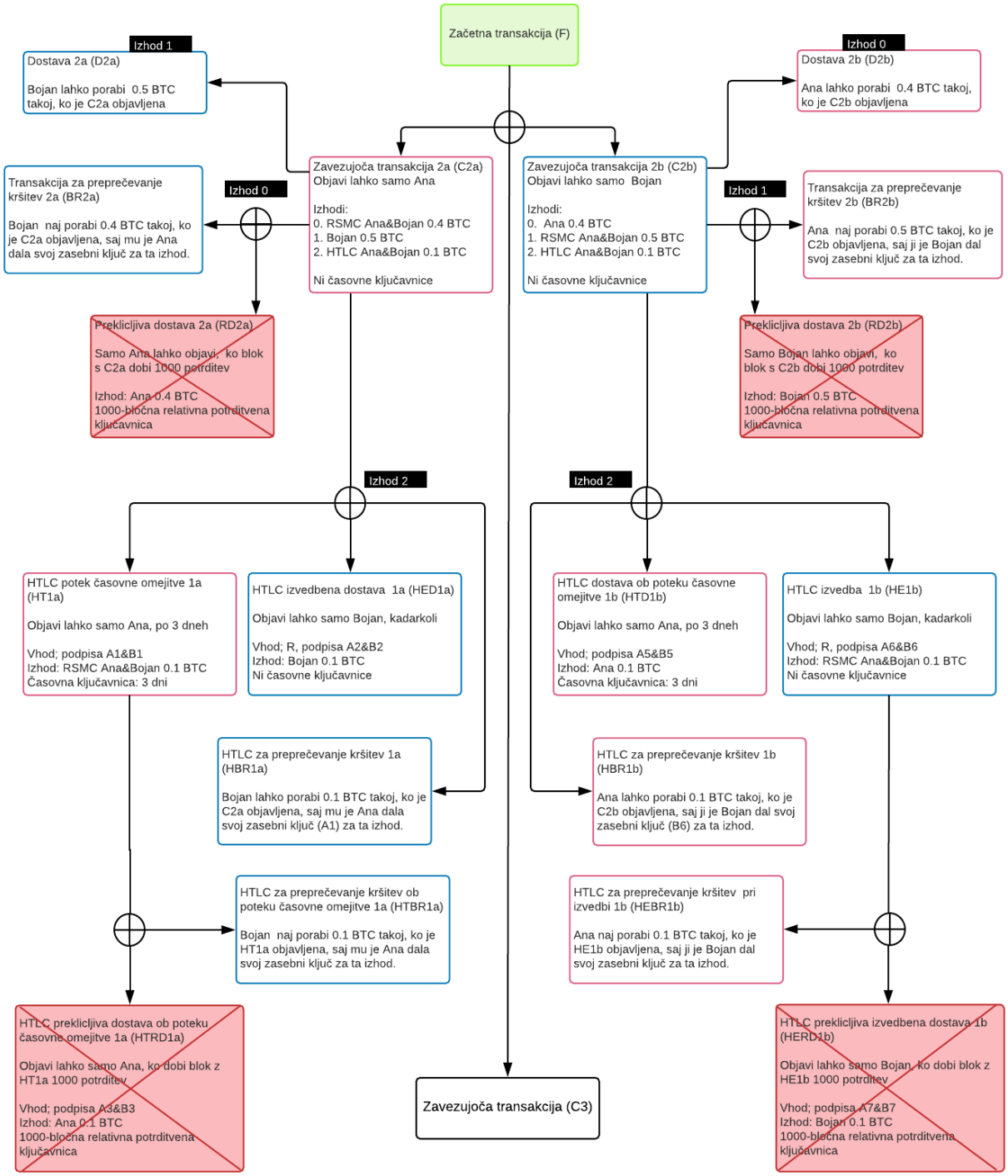
4.6.3 Zaključevanje HTLC izven verige

Kot pri sporazumnem zapiranju kanalov tudi zaključevanje HTLC izven verige zahteva strinjanje obeh strani. Če želita Ana in Bojan v primeru na sliki 4.11 ohraniti odprt plačilni kanal in zaključiti le HTLC, se morata strinjati s kreiranjem nove zavezujoče transakcije, ki odraža enega od stanj na sliki 4.12 (bodisi razmerje pred transakcijo s HTLC bodisi tistega po njej).



Slika 4.12: (a) Bojan je dokazal, da pozna vrednost R, Ana se strinja s kreiranjem nove zavezujoče transakcije C3, ki posodobi stanje v razmerju iz C2. (b) Bojan ni priskrbel praslike R, strinja se s kreiranjem nove zavezujoče transakcije C3, ki vrne stanje nazaj na razmerje pred C2.

Opazimo, da smo se ponovno znašli v situaciji, ko sta hkrati veljavna oba para zavezujočih transakcij, tako C2 kot tudi ena od različic C3 na sliki 4.12. Kot pri prekličljivih transakcijah moramo tudi tu razveljaviti stari par. Za to uporabimo transakciji za preprečevanje kršitev (BR). Poleg tega je za končanje HTLC potrebna še izmenjava zasebnih ključev, uporabljenih pri podpisu le-teh.



Slika 4.13: Preklicana transakcija C2 z zaključenimi HTLC izhodi

Ana in Bojan sta se dogovorila za kreiranje nove zavezujoče transakcije C3 iz točke (a) na sliki 4.12 in preklic transakcije C2. Bojan je Ani razkril svoj zasebni ključ za izhod pod RSMC (RD2b) ter ključa od HTLC transakcij $B6$ in $B8$, Ana pa njemu ključ od svoje transakcije RD2a ter ključa od HTLC transakcij $A1$ in $A4$. HTLC je s tem zaključena in transakcija C2 popolnoma preklicana. Kot je vidno na sliki 4.13, v primeru, da kateri od njiju kljub temu objavi transakcijo C2, za kazen izgubi vsa sredstva iz kanala.

4.7 Omrežje kanalov

Struktura *bliskovitega omrežja* je podobna topologiji pri protokolu IP. Paketi (oz. v našem primeru transakcije) potujejo po kanalih. Če prispejo na svoj cilj, končnih uporabnikov ne zanima, kaj se je dogajalo z njimi med potjo [2]. V prejšnji sekciji smo dokončno zgradili mehanizem za poslovanje dveh uporabnikov, ki sta neposredno povezana s kanalom, brez tveganja za krajo sredstev. Sedaj v prvem razdelku predstavimo komunikacijo med vozlišči v omrežju ter iskanje najbolj ugodnih kanalov med plačnikom in prejemnikom. Nato opišemo še uporabo mehanizma, ki smo ga zgradili v prejšnji sekciji, v kanalu z več posredniki.

4.7.1 Usmerjanje

Algoritmi za usmerjanje po omrežju delujejo na podoben način, kot je usmerjanje realizirano v protokolih omrežne plasti; Poon in Dryja sta v osnutku rešitve [29] predlagala uporabo protokola, podobnega BGP (angl. Border Gateway Protocol).

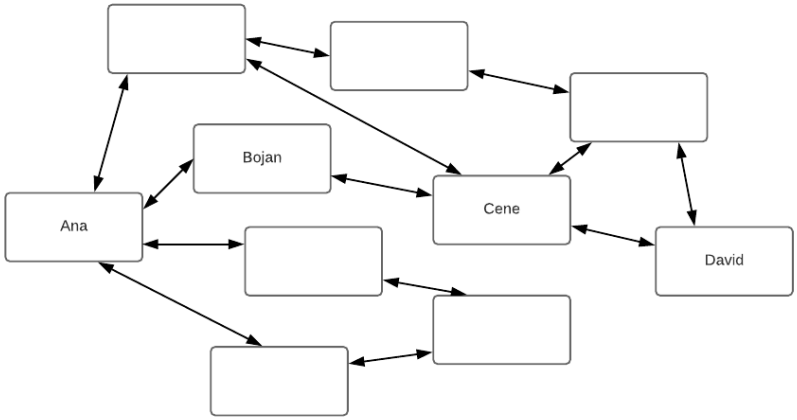
Implementacije omogočajo vsakomur, ki ima primerno opremo, da se pridruži omrežju in postane ponudnik plačilnih storitev. Uporabljen je *protokol govoric* (angl. Gossip protocol) [17], ki skrbi za razširjanje informacij o stanju kanalov v omrežju.

Ko uporabnika odpreta kanal, dobita njuna Bitcoin ključa ustrezajoča ključa za uporabo v bliskovitem omrežju. Vsak zase določi pogoje: provizijo in najmanjšo dovoljeno *razliko med potekoma časovnih ključavnic* (angl. `CheckLockTimeVerify` expiry delta, $\Delta cltv$), ki jih zahteva za posredovanje plačila skozi kanal. Medtem ko lahko kanal za svoje potrebe uporabita takoj, ga lahko začneta objavljati preko protokola govoric šele, ko je dovolj globoko v verigi [32].
Kot je navedeno v dokumentaciji implementacije c-lightning [31], se pri iskanju poti skupna cena izračuna kot vsota posameznih provizij in $\Delta cltv$ vsakega kanala na poti. Plačnik naključno izbere eno od poti z dobro kombinacijo transakcijskih provizij, zakasnitev in stabilnosti. Privzeti mehanizem izbire (iz ožjega nabora kandidatov s provizijami v mejah zahtevanega največjega odstopanja) je namenjen prikritju pošiljatelja in prejemnika. Dodatno varovalo ima na voljo plačnik. Ko izbere pot, lahko doda izračunani vrednosti $\Delta cltv$ naključen ali drugače izračunan odmik. S tem prepreči, da bi lahko kateri od posrednikov ugotovil, kje v verigi se nahaja in posledično komu je plačilo namenjeno.

4.7.2 Padajoče časovne ključavnice in plačevanje skozi kanale z več posredniki

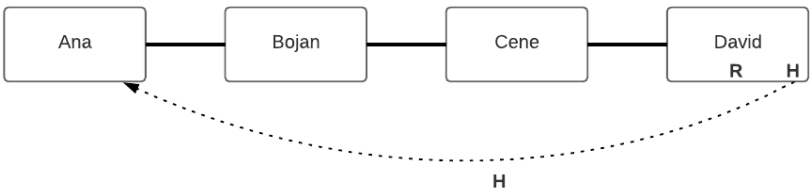
Bliskovito omrežje predstavlja rešitev Bitcoinovega problema skalabilnosti. Pričakujemo, da se bo njegov obseg vedno bolj približeval obsegu Bitcoin omrežja. V omrežju takšnih razsežnosti je nerealno pričakovati, da bomo lahko vzpostavili neposreden kanal z vsakim, s katerim želimo poslovati.
Če smo v prejšnjih sekcijah zagotovili pošteno poslovanje v kanalu, ki neposredno povezuje uporabnika, sedaj potrebujemo še način, da bo plačilo doseglo prejemnika preko enega ali več posrednikov, ne da bi ti imeli možnost ukrasti sredstva. S tem namenom moramo kot pošiljatelj pri odpiranju kanala določiti *časovno ključavnico* na podlagi izbrane poti do prejemnika.
Recimo, da se Ani in Bojanu iz prejšnjih primerov sedaj pridružita še Cene in David. Ana želi Davidu nakazati 0.01 BTC, a kot vidimo na sliki 4.14,

z njim nima odprtega kanala. Izbere naslednjo pot: Ana – Bojan – Cene – David.



Slika 4.14: Nekaj vozlišč bliskovitega omrežja. Ana z Davidom nima odprtega direktnega kanala.

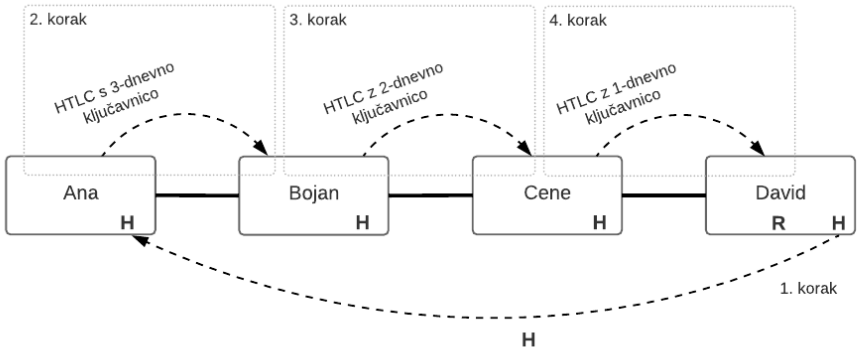
Prvi korak je odvisen od Davida. Izbrati mora neko naključno število R, ki služi kot praslika za plačilo. Izračunati mora še njegovo zgostitev $H(R)$ in jo poslati Ani.



Slika 4.15: David pošlje Ani zgostitev H, ki služi za preverjanje, ali je prejemnik v verigi res opravičen do sredstev.

Ana mora sedaj izbrati ustrezno časovno ključavnico. Z Bojanom nato ustvarita nov HTLC izhod v svojem kanalu, ki prenese Aninega 0.01 BTC Bojanu. Bojan ponovi postopek kreiranja HTLC s Cenetom, pri čemer zmanjša

časovno ključavnico za čas, ki ga ima specificiranega za svoj kanal. Cene naredi isto v kanalu z Davidom. Hkrati s kreiranjem HTLC izhodov si po kanalu posredujejo še zgostitev H.



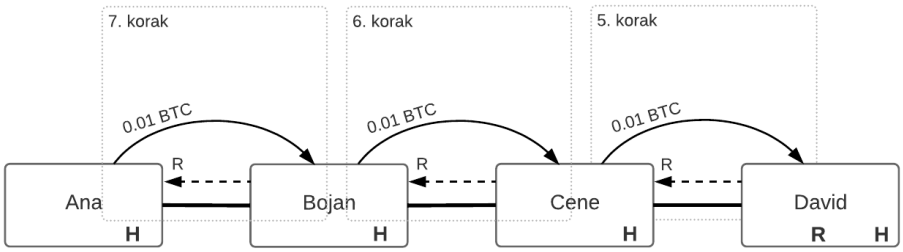
Slika 4.16: Proces kreiranja HTLC po korakih

Kot vidimo na sliki 4.16, Ana v našem primeru ključavnico nastavi na število vozlišč na poti, za vsakega naslednjega posrednika pa se le-ta zmanjša za en dan. Vsak člen na poti dobi zgostitev $H(R)$ v istem koraku, kot je kreiran HTLC, ki mu nameni sredstva, saj ga potrebuje za kreiranje HTLC z naslednjim v verigi. Vsem HTLC na poti je skupno, da za porabo izhodov, namenjenih prejemniku, poleg podpisov obeh v kanalu za izvedbo zahtevajo še vrednost R.

HTLC transakcije so sedaj na mestu. V 5. koraku na sliki 4.17 David predloži prasliko R Cenetu v roku enega dne in lahko sedaj porabi Cenetovega 0.01 BTC. Če se želita izogniti objavi v verigo, posodobita stanje z odprtjem zavezujoče transakcije, ki sredstva razdeli v novem razmerju. Sedaj Cene pozna vrednost R in ima en dan časa, da izvede 6. korak. Z njim prenese k sebi Bojanovega 0.01 BTC, Bojan pa uporabi isti postopek, da dobi sredstva od Ane. Bojan in Cene imata na koncu enako vsoto sredstev, kot sta jo imela pred transakcijo, Anino plačilo pa je uspešno doseglo Davida.

V primeru, da nek posrednik na poti postane neodziven, je tisti, ki ima z njim odprt kanal, odgovoren za objavo zavezujoče transakcije in s tem zaprtje ka-

nala med njima. Verigo potrebujemo kot posrednika samo za poravnavo z neodzivnim vozliščem, delovanje ostalih kanalov se ne spremeni.



Slika 4.17: Proces razkrivanja R in posodabljanja stanja v kanalu

Z naraščajočimi časovnimi ključavnicami v smeri primarnega plačnika poskrbimo, da ima vsak (pred iztekom svoje ključavnice) čas, da sredstva vzame nazaj, če mu tisti pred njim ni pravočasno razkril praslike R. Če Cene Bojanu ne razkrije vrednosti R v dveh dneh, lahko Bojan porabi transakcijo za potek časovne omejitve in vzame svojega 0.01 BTC nazaj. Če bo s tem čakal predolgo in bo medtem potekla njegova časovna ključavnica v kanalu z Ano, lahko Ana stori isto. Če časovna ključavnica izhoda Cenetu medtem še ni potekla in ta razkrije vrednost R, dobi Bojanovega 0.01 BTC, Bojan pa ostane brez sredstev, saj je Ana svojega 0.01 BTC že vzela nazaj. Zaradi tega je nujno, da plačnik v kanalu poskrbi za posodobitev stanja v kanalu, preden poteče njegova časovna ključavnica transakcije, s katero sredstva dobi v naslednjem kanalu.

4.8 Ranljivosti

V krovnem dokumentu projekta Lightning Network januarja 2016 [29] so za največja tveganja označene težave s potekom časovne ključavnice. Pri izbiri časovnega okvirja iščemo ravnovesje med čimbolj ugodno rešitvijo in potrebo

po učinkovitem delovanju v interakciji z neodzivnimi ali zlonamernimi sodelujočimi (bizantinska vozlišča, glej razdelek 2.3.3). Ti predstavljajo največje sistemsko tveganje v primeru, ko hkrati prisilno zaprejo veliko število kanalov in s tem naenkrat pošljejo maso transakcij proti verigi blokov. Če je količina transakcij velika, lahko povzroči dovolj dolgo zakasnitev za pretek časovne ključavnice.

Drugo tveganje predstavlja shranjevanje zasebnih ključev. Medtem ko je za varnost svoje denarnice odgovoren končni uporabnik sam, se pojavi vprašanje, kako zagotoviti ustrezno varnost posrednikov. Predlagana je razpršitev sredstev, da nobeden od posrednikov naenkrat ne poseduje velike količine sredstev v svoji spletni denarnici. Podobno kot razkritje zasebnega ključa lahko krajo sredstev omogoči tudi izguba nešifriranih podatkov. Predvideno je spodbujanje sodelujočih k zagotavljanju čim višje stopnje varnosti z vplivom le-te na izbiro posrednikov in posledično na transakcijske provizije. To je eden klasičnih pristopov pri finančnih storitvah.

Bliskovito omrežje se, kot vsa omrežja, ki uporabljajo rudarjenje, ne more izogniti tveganju za napad z združevanjem sodelujočih. Napadalec lahko podkupi sodelujoče pri napadu, da ignorirajo določene transakcije in jih ne vključijo v verigo, oni pa za dokaz označijo svoje transakcije. V izogib temu je predlagano spodbujanje neoznačenih transakcij. V pomoč je tudi dejstvo, da napadalec poleg sredstev za podpisane transakcije plača tudi provizije v kanalu. Napad zahteva visoko stopnjo sodelovanja, kar naredi izvedbo zelo zahtevno in s tem manj verjetno.

Kot so januarja 2018 opozorili razvijalci, bliskovito omrežje kljub delujoči *glavni mreži* (angl. mainnet) ni bilo priporočljivo za uporabo pri dejanskih plačilih do prihodov beta verzij. Namenjeno je bilo predvsem testiranju s strani razvijalcev in pridobivanju povratnih informacij za odpravo pomanjkljivosti.

Kot enega glavnih kritikov je potrebno omeniti Bitcoin Core razvijalca Petra Todda, ki je preko twitterja delil nepretirano spodbudna opažanja [35] pri testiranju. Todd je v svojih objavah označil izbiro programskega jezika C v implementaciji Blockstreamovega c-lightninga kot nevarno in slabo idejo. Očitne posledice so *napake pri segmentaciji* (angl. Segmentation Fault, `segfault`) in neuspela plačila. Opozoril je tudi na pomanjkljivosti pri implementaciji Eclair denarnice za Android, ki so vodile v izgubo sredstev. Zanimivo pa je, da je napovedal tudi, da se bo sam protokol Lightning izkazal za ranljivega za *napade z zavračanjem storitve* (angl. Denial of Service Attack, DoS). To se je dejansko zgodilo manj kot mesec dni kasneje. V marcu, le dober teden po zagonu beta verzij, je omrežje že postalo tarča *porazdeljenega napada z zavračanjem storitve* (angl. Distributed Denial of Service Attack, DDoS). Napad je z zapolnitvijo kapacitet povezav med vozlišči preprečil vzpostavljanje novih povezav in onemogočil skoraj 200 vozlišč od takrat delujočih 1050.

Drugega pomembnega igralca predstavljajo hekerji. Razkritje ranljivosti v začetni fazi zmanjša verjetnost velikih izgub, ki se zgodijo v primeru kasnejših uspešnih napadov, ko je delovanje omrežja v polnem razmahu. Eden takih je posameznik ali skupina pod psevdonimom bitPico [15], ki je prevzel odgovornost za prej omenjene DDoS napade z uporabo t. i. *avtomatiziranega seta orodij* (angl. toolkit), namenjenega napadom s poplavljanjem vozlišč. Sodeč po objavi na twitterju napadalčevo omrežje obsega cca 384 naprav iz osmih različnih držav, ki je hkrati uporabljalo 22 *vektorjev za napad* (angl. attack vector). Prav v času pisanja diplomske naloge bitPico prilagojeno orodje uporablja za napad na Bitcoin Cash, katerega cilj je razcepiti verigo v več vej. In čeprav je napad na Bitcoin Cash izgledal kot resna grožnja, v primeru napada na bliskovito omrežje ta ni prizadel uporabnikov. Napad je sicer povzročil motnje v delovanju omrežja in neprijetnost, ni pa imel finančnih posledic. Najdene poti služijo za spodbudo razvijalcem, da odpravijo

dan-0 ranljivosti (angl. day-0, zero-day vulnerabilities), kar je bil po besedah bitPica tudi namen napada.

S povratnimi informacijami kritikov in hekerjev so razvijalci že odpravili veliko ranljivosti, a protokol je še vedno v beta fazi testiranja, kar opozarja na potencialne nevarnosti pri uporabi v netestne namene. Prav s tem razlogom razvijalci uporabnike spodbujajo k uporabi manjših transakcij [28].

Poglavje 5

Zaključek

V diplomski nalogi smo se v začetnih poglavjih sprehodili skozi osnove, ki so potrebne za razumevanje delovanja verige blokov, ki služi kot temelj kriptovalut. Nato smo v poglavju 4 predstavili in opisali delovanje nove plasti v omrežju, ki poskrbi za rešitev problemov skalabilnosti prej omenjene tehnologije.

Svet kriptovalut je precej nemirno in nepredvidljivo področje, ki se s hitro rastjo popularnosti širi v vse smeri. Spremembe so vsakodnevne, tveganja za naložbe ogromna, novih idej veliko. Kljub temu da je večina novejših kriptovalut bolj dodelanih kot Bitcoin, pa ta ostaja vodilni na lestvici. To ga postavi v vlogo kriptovalute, ki bi jo bilo najbolj smiselno uporabljati kot sredstvo za vsakodnevno plačevanje in bi imela potencial nadomestiti sedaj uporabljane metode elektronskega plačevanja. Prav veriga blokov, tehnologija, ki je poskrbela za velik del zanimanja za kriptovalute, pa postavi omejitev za količino izvedenih plačil, ki ne more biti konkurenčna.

Bliskovito omrežje je bilo le ena mnogih idej, kako rešiti Bitcoinov problem skalabilnosti in mikrotransakcij, a hkrati edina, ki je zaživela v praksi. Predstavlja rešitev, ki poskrbi, da večina transakcij nikoli ne pride do verige. Ta služi zgolj kot višja avtoriteta, ki skrbi za spoštovanje dogovorov med posameznimi uporabniki.

Kot vsaka novost je imelo bliskovito omrežje veliko pomanjkljivosti. Čeprav,

kot je navedel Butnix [8], rešitev še ni popolna, se stvari premikajo v pravo smer.

Občutek, ki nas dela skeptične za zaupanje novim tehnologijam, je pogosto nepoznavanje in nerazumevanje, kar je tudi razlog, da veliko drugače dobrih predlogov konča svojo pot na "pokopališču idej". Z dolgim obdobjem za testiranje, podporo entuziastov in vključevanjem vedno več pomembnih igralcev v skupnosti kriptovalut, se je projekt Lightning uspešno izognil temu prepadu. Če lahko sklepamo po hitrem naraščanju števila vozlišč, že pridobiva tudi na zaupanju širše skupnosti in lahko pričakujemo, da bo ostal pomemben del Bitcoinovega omrežja.

Literatura

[1] Bitcoin developer guide. Dosegljivo: `https://bitcoin.org/en/developer-guide`. [Dostopano: 4. 9. 2018].

[2] Lightning Network Technical Design Overview. Technical report. [Dostopano: 22. 8. 2018].

[3] *FIPS PUB 180-4: Secure Hash Standard*. National Institute of Standards and Technology, Mar 2012.

[4] N. Balachandran, S. Sanyal. A Review of Techniques to Mitigate Sybil Attacks. *International Journal of Advanced Networking & Applications*, 4, Jul 2012.

[5] P. Baran. On Distributed Communications Networks. Technical Report P-2626, RAND Corporation, Santa Monica, CA, USA, Sep 1962.

[6] E. B. Barker, A. L. Roginsky. SP 800-131A. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. Technical report, Gaithersburg, MD, US, 2011.

[7] I. Bashir. *Mastering Blockchain*. Packt Publishing, 2017.

[8] J. Buntinx. Lightning "Critic" Peter Todd Confirms Bitcoin's Scaling Solution is Maturing. Dosegljivo: `https://www.newsbtc.com/2018/04/22/lightning-critic-peter-todd-confirms-bitcoins-scaling-solution-maturing/?utm_source=dlvr.it&utm_medium=twitter`. [Dostopano: 10. 7. 2018].

[9] D. Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 199–203. Plenum, 1982.

[10] S. Gordon. Vaultoro's Bitcoin-to-Gold Exchange Implements Lightning Network Payments. Dosegljivo: `https://bitcoinmagazine.com/articles/vaultoros-bitcoin-gold-exchange-implements-lightning-network-payments/`. [Dostopano: 24. 8. 2018].

[11] R. A. Grimes. All you need to know about the move from SHA-1 to SHA-2 encryption. Dosegljivo: `https://www.csoonline.com/article/2879073/encryption/all-you-need-to-know-about-the-move-from-sha1-to-sha2-encryption.html`. [Dostopano: 1. 6. 2018].

[12] S. Haber, W. S. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, Jan 1991.

[13] T. Hansen, D. E. Eastlake 3rd. US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF). RFC 6234, May 2011.

[14] A. Hertig. Lightning At Last? Bitcoin Scaling Layer Almost Readyn. Dosegljivo: `https://www.coindesk.com/lightning-last-test-shows-bitcoin-scaling-solution-almost-ready/`. [Dostopano: 24. 8. 2018].

[15] A. Hertig. Lightning is being attacked for its own good. Dosegljivo: `https://www.coindesk.com/bitcoins-lightning-network-attacked-good/`. [Dostopano: 10. 7. 2018].

[16] A. Hofman. NXT – Proof of Stake and the New Alternative Altcoin. Dosegljivo: `https://bitcoinmagazine.com/articles/nxt-proof-stake-new-alternative-altcoin-1391226906/`. [Dostopano: 20. 8. 2018].

[17] M. Jelasity. Gossip-based Protocols for Large-scale Distributed Systems.

[18] A. Jurišić, A. J. Menezes. Elliptic curves and cryptography. *Dr. Dobb's Journal*, pages 26–37, 1997.

[19] J. Katz, Y. Lindell. *Introduction to Modern Cryptography, Second Edition.* Chapman & Hall/CRC, 2014.

[20] K. Lauslahti, J. Mattila, T. Seppälä. Smart Contracts – How will Blockchain Technology Affect Contractual Practices?. *ETLA Reports*, 68, Jan 2017.

[21] J. López, R. Dahab. An overview of elliptic curve cryptography. Technical report, Institute of Computing, State University of Campinas, May 2000.

[22] N. G. Mankiw. *Macroeconomics, 7th Edition.* Worth Publishers, 2010.

[23] C. Masters. Litecoin (LTC) Can Be Key Element in Bitcoin (BTC) Lightning Network, Says Charlie Lee. Dosegljivo: `https://cryptovest.com/news/litecoin-ltc-can-be-key-element-in-bitcoin-btc-lightning-network-says-charlie-lee/`. [Dostopano: 24. 8. 2018].

[24] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, R. Brooks. A brief survey of Cryptocurrency systems. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 745–752, Dec 2016.

[25] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Dosegljivo: `http://www.bitcoin.org/bitcoin.pdf`, 2008. [Dostopano: 16. 5. 2018].

[26] M. S. Niaz, G. Saake. Merkle hash tree based techniques for data integrity of outsourced data. *CEUR Workshop Proceedings*, 1366:66–71, Jan 2015.

[27] B. Patryk. Finally we have first exchange accepting LN on MAINNET! Dosegljivo: https://twitter.com/BigajPatryk/status/981155354636685312. [Dostopano: 24. 8. 2018].

[28] D. Pollock. Lighting Labs Co-Founder and Bitcoin Developers Slam Study on Lightning Network. Dosegljivo: https://cointelegraph.com/news/lighting-labs-co-founder-and-bitcoin-developers-slam-study-on-lightning-network. [Dostopano: 22. 8. 2018].

[29] J. Poon, T. Dryja. The bitcoin lightning network: Scalable off-chain instant payments. Technical report, 2016.

[30] G. Prisco. Blockstream Releases Lightning Charge, Launches Test E-Commerce Store. Dosegljivo: https://bitcoinmagazine.com/articles/blockstream-releases-lightning-charge-launches-test-e-commerce-store/. [Dostopano: 24. 8. 2018].

[31] R. Russell et al. c-lightning — a lightning network implementation in c. Dosegljivo: https://github.com/ElementsProject/lightning/. [Dostopano: 28. 8. 2018].

[32] R. Russell et al. Lightning network specifications. Dosegljivo: https://github.com/lightningnetwork/lightning-rfc. [Dostopano: 28. 8. 2018].

[33] D. R. Stinson. *Cryptography: Theory and Practice*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1995.

[34] N. Szabo. Smart Contracts. Dosegljivo: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html. [Dostopano: 10. 7. 2018].

[35] P. Todd. Initial impressions of lightning on testnet. Dosegljivo: https://twitter.com/peterktodd/status/968190530294337538. [Dostopano: 22. 8. 2018].

[36] Announcing our first Lightning mainnet release, lnd 0.4-beta! Dosegljivo: https://blog.lightning.engineering/announcement/2018/03/15/lnd-beta.html. [Dostopano: 24. 8. 2018].

[37] Announcing the first SHA1 collision. Dosegljivo: https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html. [Dostopano: 30. 8. 2018].

[38] Blackcoin: Faq. Dosegljivo: https://blackcoin.org/faq/. [Dostopano: 20. 8. 2018].

[39] Blockchain. Dosegljivo: https://en.bitcoin.it/wiki/File:Blockchain.png. [Dostopano: 20. 8. 2018].

[40] Eclair v0.2-beta1. Dosegljivo: https://github.com/ACINQ/eclair/releases. [Dostopano: 24. 8. 2018].

[41] Ocena porazdelitve moči zgoščevanja med največjimi rudarskimi združenji. Dosegljivo: https://blockchain.info/pools. [Dostopano: 12. 6. 2018].

[42] Peercoin: minting. Dosegljivo: https://peercoin.net/minting. [Dostopano: 20. 8. 2018].

[43] Slush pool. Dosegljivo: https://slushpool.com/home/. [Dostopano: 31. 8. 2018].

[44] Value overflow incident. Dosegljivo: https://en.bitcoin.it/wiki/Value_overflow_incident. [Dostopano: 22. 8. 2018].

[45] Visa Inc. at a Glance . Dosegljivo: https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf. [Dostopano: 29. 8. 2018].

[46] What is bitcoinj? Dosegljivo: `https://bitcoinj.github.io/`. [Dostopano: 22. 8. 2018].