Raspodijeljene glavne knjige i kriptovalute

Digitalni potpis i jednostavne kriptovalute

Ante Đerek, Zvonko Konstanjčar

13. listopada 2023.

Ponavljanje: Kriptografska hash funkcija

Osnovna svojstva

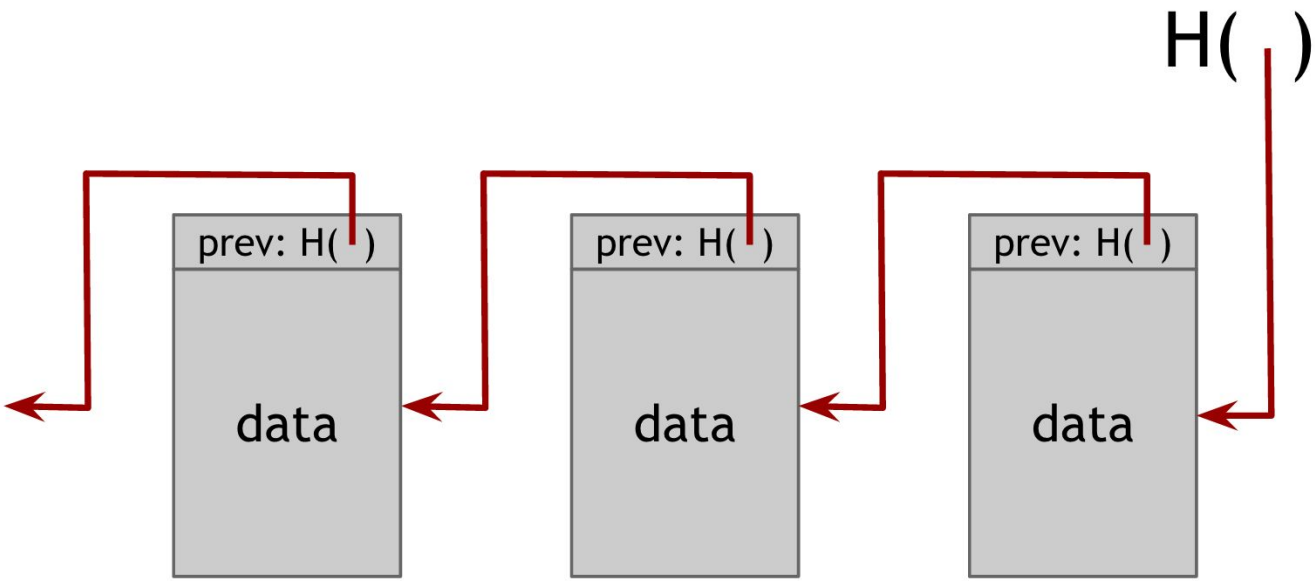
- Ulaz je niz bitova proizvoljne duljine.
- Izlaz je niz bitova fiksne duljine (npr. točno 256 bita).
- Funkcija je deterministička i može se brzo i efikasno izračunati.

Otporna na kolizije

Ako je "praktički nemoguće" pronaći dvije različite poruke x i y takve da vrijedi $H(x) = H(y)$.

Definicija

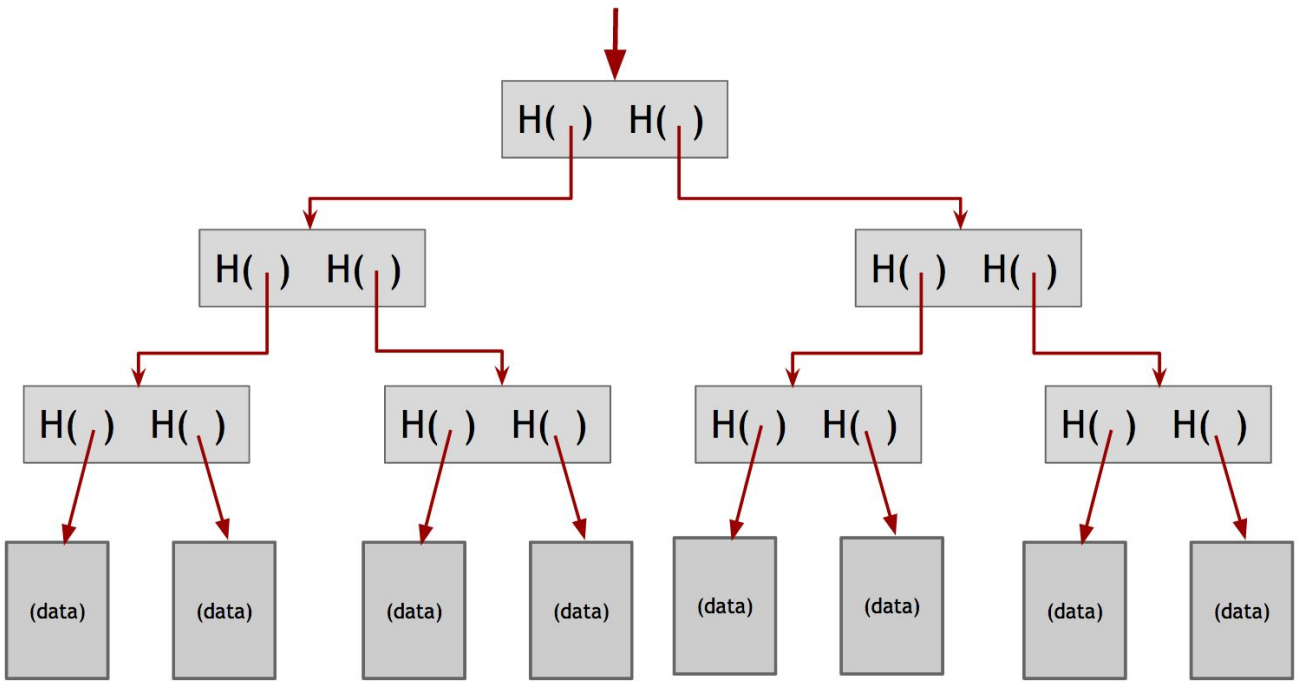
Kriptografski lanac blokova je jednostruko povezana lista u kojoj svaki element (uz neke podatke) sadrži hash pokazivač na prethodni element.



Izvor: `bitcoinbook.cs.princeton.edu`

Definicija

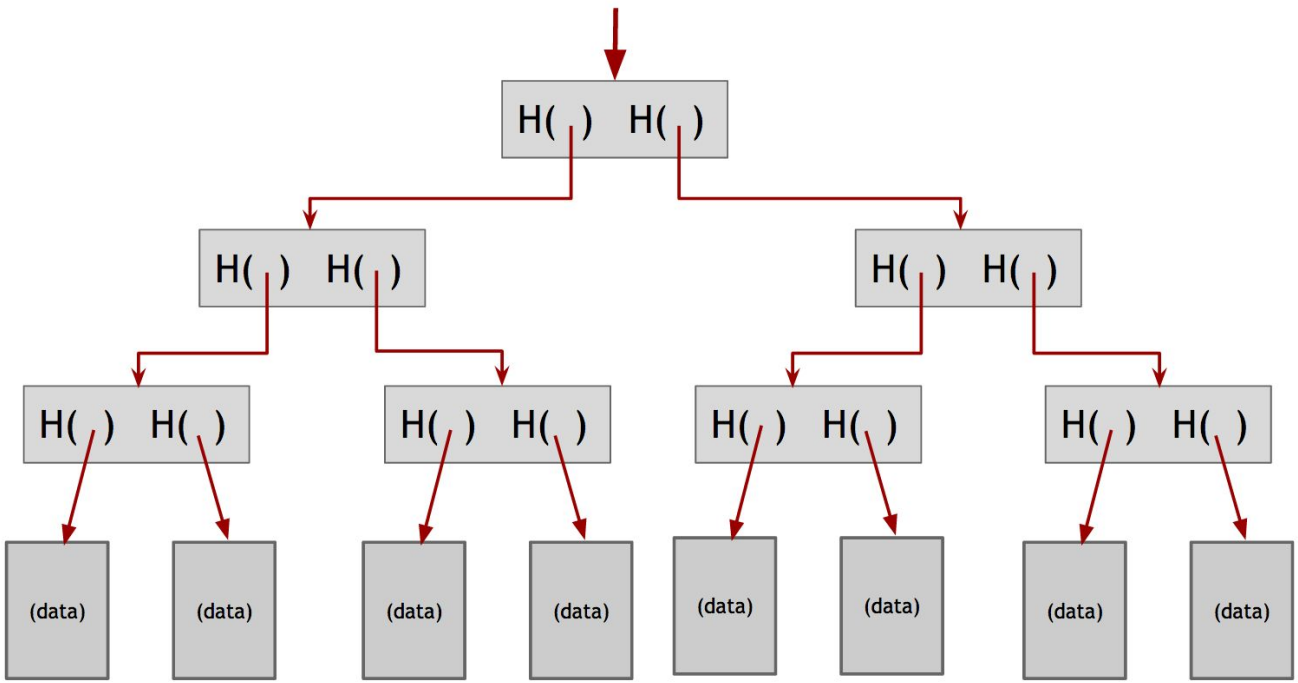
Merkleovo stablo je potpuno binarno stablo u kojem svaki unutarnji čvor sadrži hash pokazivače na svoja dva djeteta.



Izvor: `bitcoinbook.cs.princeton.edu`

Zadatak

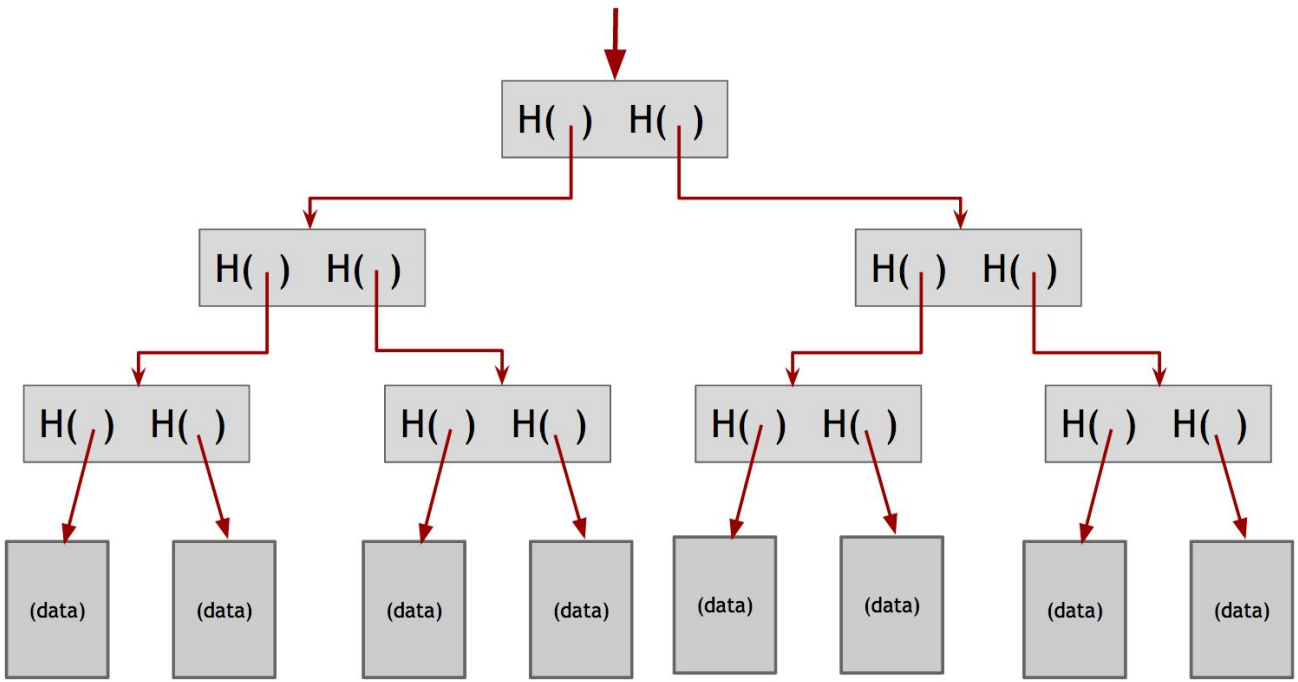
Kako možemo nekoga uvjeriti da se određeni list stvarno nalazi u stablu?



Izvor: `bitcoinbook.cs.princeton.edu`

Izazov

Kako možemo nekoga uvjeriti da element nije dio stabla?



Izvor: `bitcoinbook.cs.princeton.edu`

Sustav digitalnog potpisa

Trojka algoritama:

- $G()$ – algoritam koji generira par ključeva (sk, pk).
- $S(sk, m)$ – algoritam koji na temelju privatnog ključa sk i poruke m generira *potpis* $\sigma \leftarrow S(sk, m)$.
- $V(pk, m, \sigma)$ – algoritam koji prima javni ključ, poruku i njezin tobožnji potpis i vraća `true` ako je σ ispravan potpis poruke m odgovarajućim privatnim ključem, a `false` ako nije.

Svojstvo korektnosti: Ispravni potpisi prolaze provjeru

Ako je $(sk, pk) \leftarrow G()$, onda za svaku poruku m vrijedi $V(pk, m, S(sk, m)) = \texttt{true}$.

Praktički je nemoguće krivotvoriti potpis

Napadač koji nema privatni ključ ne može konstruirati niti jednu *novu* poruku m i njezin potpis σ koji prolazi postupak provjere, tj. za koje vrijedi $V(pk, m, \sigma) = \texttt{true}$.

- Čak i ako napadač zna odgovarajući javni ključ pk.
- Čak i ako napadač ima mogućnost da dobije potpis $\sigma' \leftarrow S(sk, m')$ proizvoljne poruke m'.

Sve konstrukcije koje znamo su bazirane na matematici:

- RSA
- DSA
- ECDSA
- ...

Bitcoin koristi ECDSA sustav s "secp256k1" krivuljom:

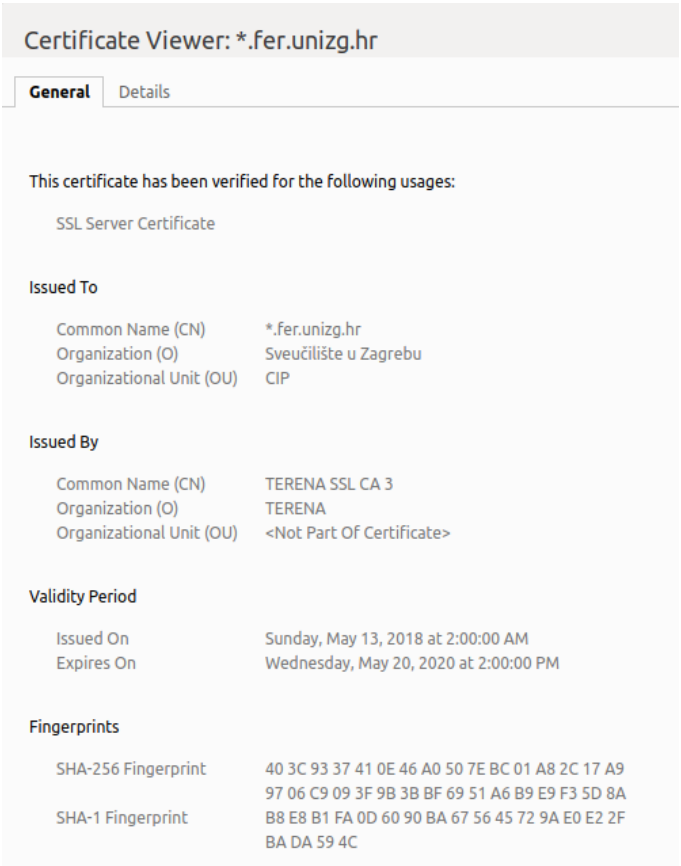
- Privatni ključ: 256 bita
- Javni ključ: 520 bita ("kompresirani" javni ključ 264 bita)
- Potpis: 512 bita
- Efektivna veličina ključa: 128 bita
- Hash funkcija SHA256 je dio algoritma potpisivanja.

Digitalni potpis – primjene

- Potpisivanje elektronskih dokumenata.
- Sigurnosni protokoli (TLS, . . .).
- Autentifikacija email-a.
- Provjera integriteta software-a (apk, exe, firmware, . . .).
- Kriptovalute.
- . . .

Većina primjena digitalnog potpisa

Bitna je veza između identiteta i javnog ključa!



Certificate Viewer: *.fer.unizg.hr

General | Details

This certificate has been verified for the following usages:

SSL Server Certificate

Issued To

Common Name (CN)	*.fer.unizg.hr
Organization (O)	Sveučilište u Zagrebu
Organizational Unit (OU)	CIP

Issued By

Common Name (CN)	TERENA SSL CA 3
Organization (O)	TERENA
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Sunday, May 13, 2018 at 2:00:00 AM
Expires On	Wednesday, May 20, 2020 at 2:00:00 PM

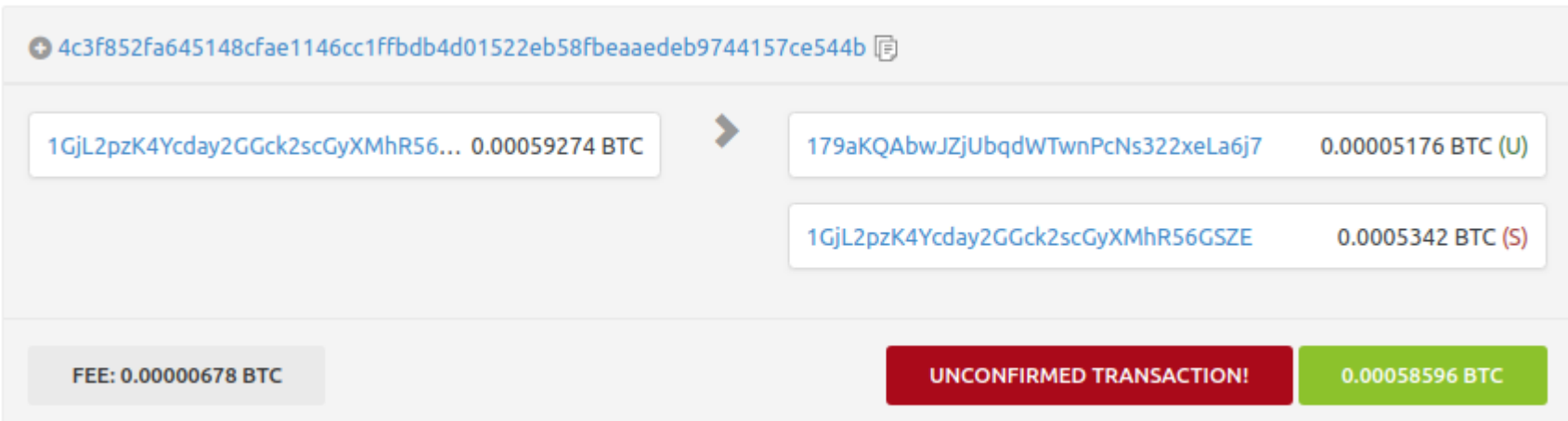
Fingerprints

SHA-256 Fingerprint	40 3C 93 37 41 0E 46 A0 50 7E BC 01 A8 2C 17 A9 97 06 C9 09 3F 9B 3B BF 69 51 A6 B9 E9 F3 5D 8A
SHA-1 Fingerprint	B8 E8 B1 FA 0D 60 90 BA 67 56 45 72 9A E0 E2 2F BA DA 59 4C

Kriptovalute

Nema veze između identiteta i javnog ključa!

Identitet = Javni ključ!



Izvor: `blockexplorer.com`

Željko stvara novi ŽeljkoCoin novčić tako da:

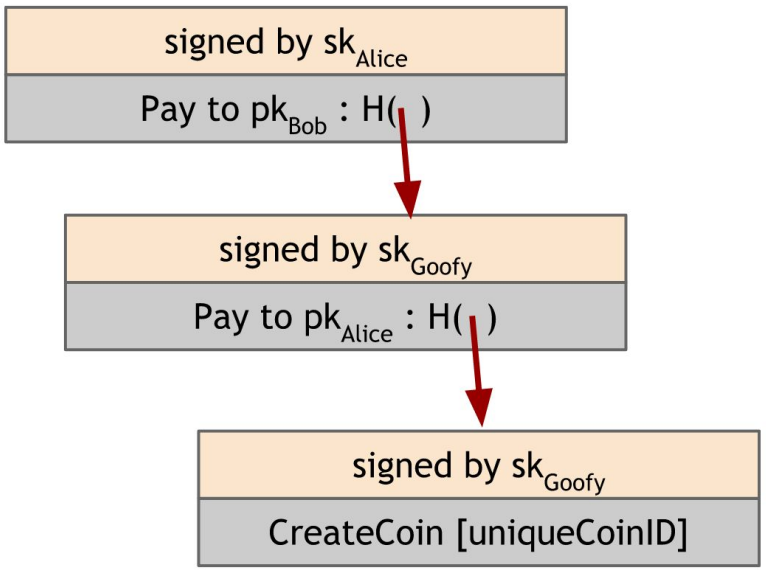
1. Izabere novi jedinstveni *uniqueCoinId*.
2. Izgradi poruku $m =$ "`CreateCoin[`*uniqueCoinId*`]`".
3. Potpiše poruku m: $\sigma = S(sk_{Zeljko}, m)$.
4. Par (m, σ) je novi novčić.

Željko prenese ŽeljkoCoin novčić Ani tako da:

1. Izgradi hash pokazivač c na novčić koji želi prenijeti.
2. Izgradi poruku $m =$ "`Pay to` $pk_{Ana} : c$",
3. Potpiše poruku m: $\sigma = S(sk_{Zeljko}, m)$.
4. Par (m, σ) je novi novčić.

Stvaranje i prenošenje novčića

- Samo Željko može stvarati novčiće.
- *Vlasnik* novčića može ga prenijeti nekome drugome.



Izvor: `bitcoinbook.cs.princeton.edu`

Zanimljiva svojstva

- Što je to identitet?
- Tko je vlasnik novčića?
- Tko može provjeriti ispravnost novčića?
- Moraju li novčići biti tajni?
- Kako netko može *ukrasti* novčiće?
- Je li potreban centralni autoritet za transakcije?

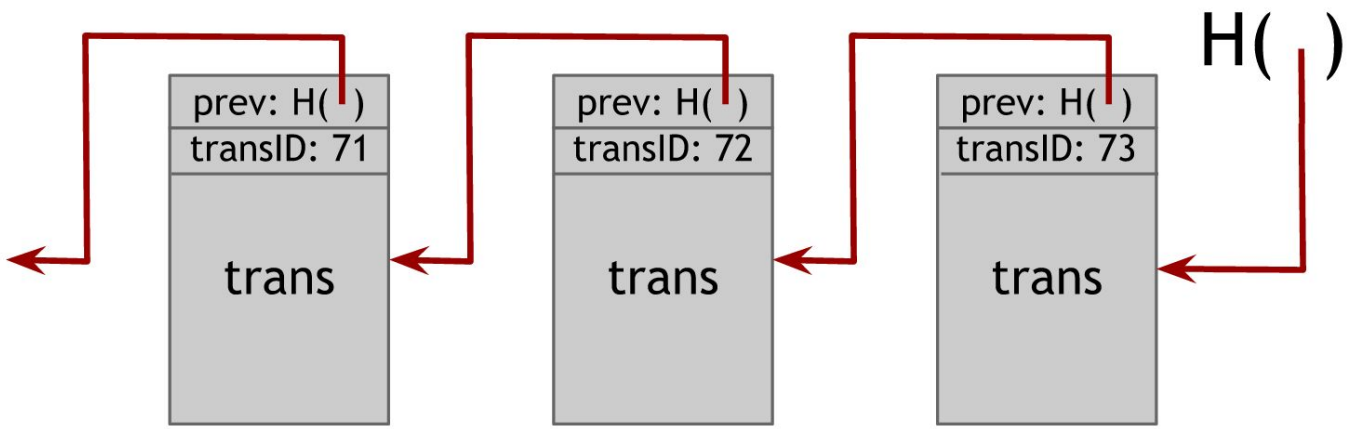
Ključni problem – dvostruko trošenje!

Ana može isti novčić c prenijeti i Mirku i Slavku.

- $m_1 =$ "`Pay to` $pk_{Mirko} : c$", $\sigma_1 = S(sk_{Ana}, m_1)$.
- $m_2 =$ "`Pay to` $pk_{Slavko} : c$", $\sigma_2 = S(sk_{Ana}, m_2)$.

Lanac blokova kao zaštita od dvostrukog trošenja!

- Stvaranje i prenošenje: slično kao kod ŽeljkoCoin-a.
- Branko održava i javno objavljuje (digitalno potpisani) kriptografski lanac blokova koji sadrži sve transakcije ikad izvršene u sustavu.
- Transakcija se smatra *izvršenom* samo ako se nalazi u lancu blokova.



Izvor: bitcoinbook.cs.princeton.edu

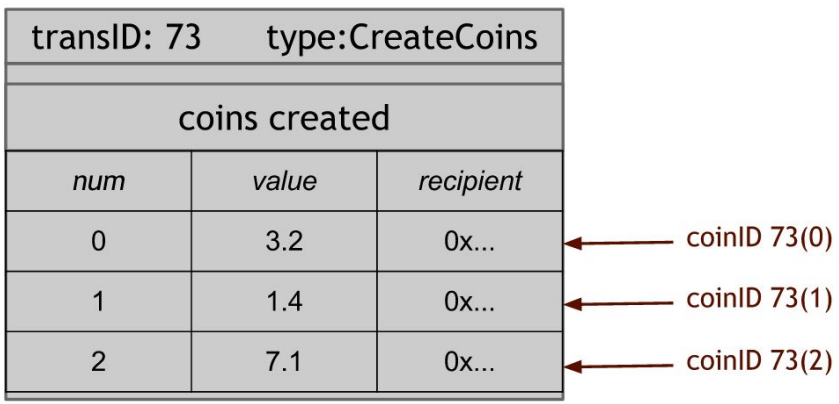
Branko stvara novi BrankoCoin novčić tako da:

1. Izgradi novu CreateCoins transakciju u kojoj zabilježi iznos i javni ključ vlasnika novčića.
2. Potpiše transakciju svojim privatnim ključem sk_{Branko}.
3. Doda potpisanu CreateCoins transakciju na kraj lanca blokova.

```
transaction:
  type: CreateCoins
  coinsCreated:
    - value: 3.2
      recipient: 0xf9c817928ebb56e4b7b49c75c08b9d2e...
signature: 0xd7fddbbc75e769dfa1e47886f4770db7...
```

Implementacijski detalji

- Moguće stvoriti više novčića odjednom.
- Svakoj transakciji Branko dodijeli jedinstveni serijski broj.
- Svaki novčić ima jedinstveni redni broj unutar transakcije.



Izvor: `bitcoinbook.cs.princeton.edu`

Ana plaća BrankoCoin novčićem Mirku tako da:

1. Odabere vlastiti nepotrošeni novčić c u lancu blokova.
2. Izgradi PayCoins transakciju u kojoj kaže da troši novčić c, a da nastaje novi novčić istog iznosa kojemu je javni ključ vlasnika pk_{Mirko}.
3. Potpiše transakciju svojim privatnim ključem sk_{Ana}.
4. Pošalje potpisanu transakciju Branku za objavljivanje u lancu.

```
transaction:
  type: PayCoins
  consumedCoinId: 73(0)
  consumedCoinHash: 0x530be0576140831c7900b271ce90c0f5...
  coinsCreated:
    - value: 3.2
      recipient: 0xe7a0f06858dc8a2323e387cbe797cf48...
signature: 0xc9491ba77e2e8a19040826f0e070162d...
```

Branko bilježi transakciju tako da:

1. Provjeri da novčić c nije već potrošen.
2. Provjeri da iznos novog novčića odgovara iznosu novčića c.
3. Provjeri da je c stvarno novčić koji pripada Ani.
4. Provjeri ispravnost potpisa na transakciji pomoću Aninog javnog ključa.
5. Dodaje transakciju u lanac blokova.

```
transaction:
  type: PayCoins
  consumedCoinId: 73(0)
  consumedCoinHash: 0x530be0576140831c7900b271ce90c0f5...
  coinsCreated:
    - value: 3.2
      recipient: 0xe7a0f06858dc8a2323e387cbe797cf48...
signature: 0xc9491ba77e2e8a19040826f0e070162d...
```

Zadatak

Kako Branko provjeri da je novčić c stvarno Anin?
Odakle mu Anin javni ključ pk_{Ana}?

```
transaction:
  type: CreateCoins
  coinsCreated:
    - value: 3.2
      recipient: 0xf9c817928ebb56e4b7b49c75c08b9d2e...
signature: 0xd7fddbbc75e769dfa1e47886f4770db7...
```

```
transaction:
  type: PayCoins
  consumedCoinId: 73(0)
  consumedCoinHash: 0x530be0576140831c7900b271ce90c0f5...
  coinsCreated:
    - value: 3.2
      recipient: 0xe7a0f06858dc8a2323e387cbe797cf48...
signature: 0xc9491ba77e2e8a19040826f0e070162d...
```

Zadatak

Što to znači da novčić nije potrošen i kako to provjeriti?

```
transaction:
  type: CreateCoins
  coinsCreated:
    - value: 3.2
      recipient: 0xf9c817928ebb56e4b7b49c75c08b9d2e...
signature: 0xd7fddbbc75e769dfa1e47886f4770db7...
```

```
transaction:
  type: PayCoins
  consumedCoinId: 73(0)
  consumedCoinHash: 0x530be0576140831c7900b271ce90c0f5...
  coinsCreated:
    - value: 3.2
      recipient: 0xe7a0f06858dc8a2323e387cbe797cf48...
signature: 0xc9491ba77e2e8a19040826f0e070162d...
```

Mirko primi BrankoCoin plaćanje od Ane tako da:

1. Pošalje svoj javni ključ Ani ako je potrebno.

Kada je transakcija u lancu blokova može raspolagati s novčićem!

```
transaction:
  type: PayCoins
  consumedCoinId: 73(0)
  consumedCoinHash: 0x530be0576140831c7900b271ce90c0f5...
  coinsCreated:
    - value: 3.2
      recipient: 0xe7a0f06858dc8a2323e387cbe797cf48...
signature: 0xc9491ba77e2e8a19040826f0e070162d...
```

Detalj – Moguće potrošiti i stvoriti više novčića odjednom.

Branko provjerava:

- Svi novčići koji se troše su nepotrošeni.
- Svi vlasnici svih novčića koji se troše su potpisali transakciju.
- Ukupna vrijednost potrošenih novčića je jednaka kao ukupna vrijednost stvorenih novčića.

transID: 73	type:PayCoins	
consumed coinIDs: 68(1), 42(0), 72(3)		
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...
signatures		

Izvor: `bitcoinbook.cs.princeton.edu`

Zadatak

Zašto Ana ne može isti novčić potrošiti dvaput?

Zadatak

Može li netko drugi potrošiti Anin novčić?

Zadatak

Što ako Ana ne želi potrošiti cijeli novčić odjednom?

Zadatak

Može li Branko ukrasti ili potrošiti tuđi novčič?

Zadatak

Može li Branko obrisati ili modificirati staru transakciju?

Zadatak

Može li Branko uskratiti uslugu?

Cilj

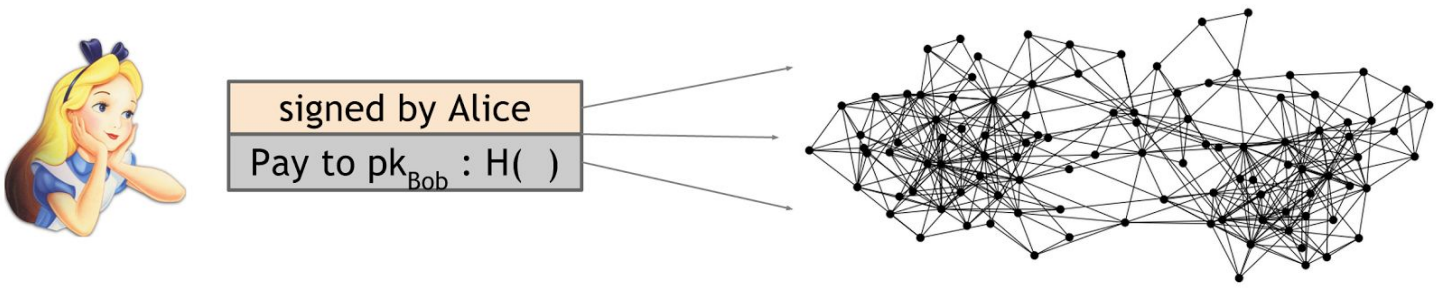
BrankoCoin sustav ali bez Branka!

Tko će preuzeti Brankov posao?

- Kako održavati raspodijeljeni lanac blokova?
- Tko provjerava ispravnost transakcija?
- Tko i kada smije stvarati nove novčiće?

Arhitektura sustava

- Puno čvorova u "peer-to-peer" mreži.
- Svi ćvorovi imaju identične kopije lanca blokova.
- Svaki čvor održava skup transakcija koje treba dodati u lanac.
- Periodički se dodaje novi blok u lanac:
  - Svaki čvor predloži potencijalni sljedeći blok.
  - Čvorovi se nekim usaglase čiji će prijedlog dodati u lanac.
  - Svaki čvor doda odabrani blok u svoj lanac.

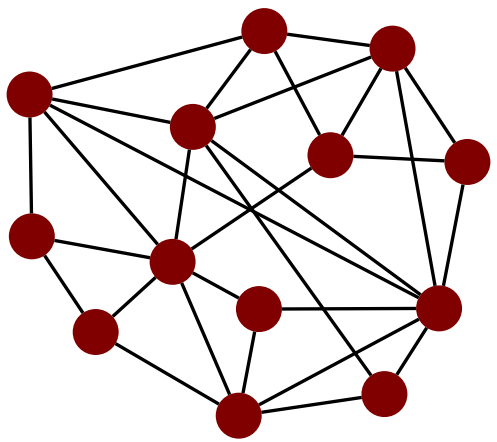


Izvor: `bitcoinbook.cs.princeton.edu`

Definicija

U mreži se nalazi n čvorova, neki čvorovi su ispravni i oni vjerno prate pravila protokola, dok su drugi neispravni ili zlonamjerni. Svaki čvor k ima neku ulaznu vrijednost x_k. Protokol za raspodijeljeni konsenzus je mehanizam za kojeg vrijedi:

- *Svaki ispravni čvor k izračuna izlaznu vrijednost y_k.*
- *Izlazna vrijednost svih ispravnih čvorova je jednaka.*
- *Ta izlazna vrijednost je jednaka ulaznoj vrijednosti x_k nekog ispravnog čvora.*



Važan praktičan problem!

- Replikacija baze podataka.
- Sinkronizacija satova.
- Sustavi za upravljanje letom.
- "Real-time strategy" igrice.
- ...

Težak problem

- Teorijski rezultati: nemoguće ako je komunikacija asinkrona, nemoguće ako je više od jedne trećine čvorova zlonamjerno.
- "Standardno" rješenje: Paxos protokol.
- Bitcoin: "Proof-of-work"

Nerealna pretpostavka

Postoji mehanizam (nazovimo ga "KBV") koji omogućuje odabir slučajnog čvora u mreži. Štoviše, mehanizam je takav da je vjerojatnost da je slučajno odabrani čvor *ispravan* veća od pola.

```
kbvAnnouncement:
  time: 13.10.2022. 16:50
  winnerPk: 0xf9c817928ebb56e4b7b49c75c08b9d2e...
signature: 0xd7fddbbc75e769dfa1e47886f4770db7...
```

Postupak određivanja sljedećeg bloka

- KBV odabere slučajni čvor A i objavi ga svim čvorovima.
- A predloži sljedeći blok x i objavi ga svim čvorovima.
- Ostali čvorovi provjeravaju autentičnost ispravnost bloka x.
- Čvorovi *prihvaćaju* blok x ako je ispravan te dolazi od čvora A, ignoriraju ako nije.

Pažnja!

- Blok sadrži hash pokazivač na prethodni blok. Dakle, prihvaćanje bloka je *prihvaćanje lanca*, provjera ispravnosti bloka je *provjera ispravnosti lanca*.
- Konsenzus je *implicitan* – ako je čvor prihvatio novi blok onda će njega nadograđivati ako njega sljedećeg odabere KBV.