Table of contents

GoofyCoin

- Imaginary cryptocurrency
- One main character Goofy
- Two simple rules
  - CreateCoin
  - Sending coins

GoofyCoin

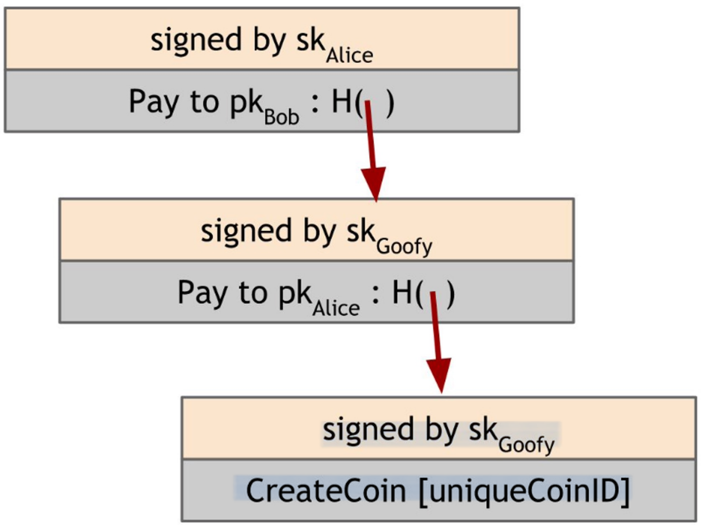
- Rule 1: CreateCoin
  - Goofy can generate new coins when he wants
    - Generates a unique+fresh coin ID “uniqueCoinID”
    - String *CreateCoin [uniqueCoinID]*
    - Digitally signs the string with his signature
      - New coin created

GoofyCoin

- Rule 2: Paycoin
  - Whoever owns a coin can transfer it on to someone else
    - *Pay this coin to X’s public key*
      - *this* is a hash pointer to the coin

GoofyCoin

- Verification
  - Follow the chain of hash pointers until creation by Goofy
    - Verify all of signatures along the way



GoofyCoin

- Fundamental security problem with GoofyCoin

- One can double spend a coin
  - Tx are not advertised

- Perfectly valid Tx for both recipients
  - Signed
  - Traceback to genesis $\longrightarrow$ **As it can be traced back to the genesis , means when goofy first created it. It simply means every branch of the tree will eventually land up to the root for sure.**
  - Can claim to be the owner

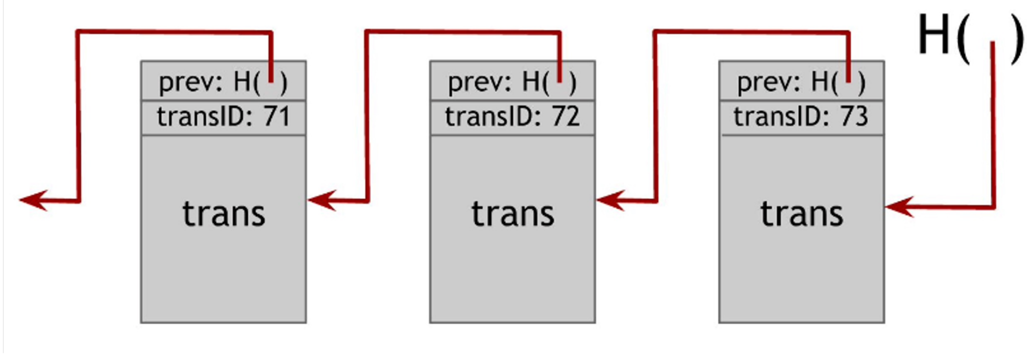
- GoofyCoin is not secure

ScroogeCoin

- ScroogeCoin is built over of GoofyCoin
  - Scrooge can also generate new coins when he wants
- Scrooge publishes an append-only ledger
  - History of all Tx
- If the ledger is truly append-only
  - Use it to defend against double spending
  - Each Tx is written in ledger before accepting them

ScroogeCoin

- Scrooge can build a blockchain
  - Series of data blocks (one Tx/block, for simplicity)
  - Each block has TxID, Tx contents, hash pointer to previous block
  - Scrooge digitally signs each data block
    - Final hash pointer
    - Publishes the signature along blocks



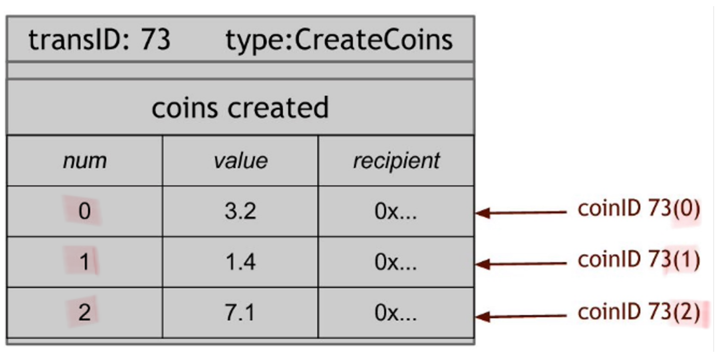
ScroogeCoin

- Scrooge verifies double spending Tx
- What if Scrooge manipulates old data
    - Hash pointers signed by him

The manipulation of old data means manipulation of hash pointers signed by him. This is what the sub bullets mean in these presentations , i.e. what does the sentence above really mean.

ScroogeCoin

- Rule 1: CreateCoin
    - Multiple coins in one Tx
    - Each coin has
        - a serial number within Tx
            - CoinID = ID of Tx (coin's serial # in that Tx)
        - a value
        - a recipient, first recipient's public key



transID: 73 type:CreateCoins

coins created

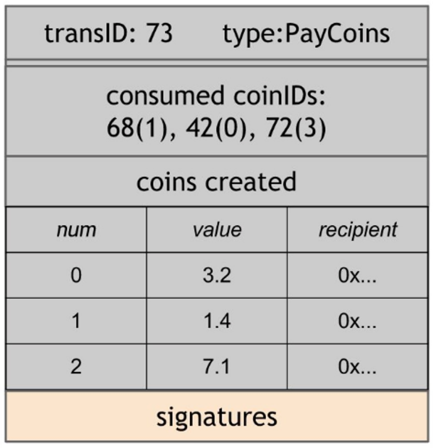
num	value	recipient	
0	3.2	0x...	coinID 73(0)
1	1.4	0x...	coinID 73(1)
2	7.1	0x...	coinID 73(2)

ScroogeCoin

- Rule 2: Paycoin
  - Coins are immutable (never changed, subdivided, or combined)
    - Consume some coin (destroy them) and create new coins of same total value
  - New coins might belong to different people (public keys)

ScroogeCoin

- PayCoin Tx is valid if
  - Consumed coins are valid
    - They were indeed created
  - Consumed coins were not already consumed
    - Prevent double spending
  - Total input coins value = Total output coins value
    - No new "value" is created here
  - All owners of input coins validly signed Tx
    - Owner’s permission



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

ScroogeCoin

- If all conditions are met
  - PayCoin Tx is valid, Scrooge accepts it
  - Scrooge write/appends it into blockchain
    - Only now others can see/accept that this Tx has happened
- Until a Tx is published
  - It can be preempted by a double spending Tx

ScroogeCoin

- ScroogeCoin prevents double spending
- Scrooge has too much influence (centralization)
  - Can’t create fake Tx
    - Can’t forge other people’s signatures
  - Can create new coins for himself
  - Could stop endorsing Tx from some users
    - Ask for a fee
  - Could get bored of the system and stops operating blockchain

ScroogeCoin

- Solution: de-Scrooge-ify
  - Decentralization

- Bitcoin is about decentralization
  - Decentralization is an important concept
  - Not unique to Bitcoin
    - Who maintains the ledger of Tx?
    - Who has authority over which Tx are valid?
    - Who creates new Bitcoin?
    - Who determines how the rules of the system change?
    - How do Bitcoin acquire exchange value?