

Transactions and UTXO

Digital Signatures

public key and secret key pair (pk, sk)
(private key) $(pk, pri k)$

$$G \leftarrow \text{Sign}(m, sk)$$

$$\{true, false\} \leftarrow \text{Verify}(m, G, pk)$$

→ Exercise 1

Idea: Why not secret key = password
 $G = H(m || \text{password})$
verify/

Accounts

Idea: Use public key as identity.

Bank $\text{map}[pk] \text{ balance}$

deposit (pk, value) :

$\text{balances}[pk] += \text{value}$

transfer $(\text{from-pk}, \text{to-pk}, \text{value}, \text{signature})$

if: $\text{value} > 0$ and

$\text{balance}[\text{from-pk}] \geq \text{value}$ and

$\text{verify}(\text{to-pk} || \text{value}, \text{signature}, \text{from-pk})$

then

$$\text{balance}[\text{from}] -= \text{val}$$
$$\text{bal}[\text{to}] += \text{val}$$

Idea: Everybody runs the bank.

- Put transfers publicly on the blockchain.
- Everybody verifies and applies them

Problem: Replay attack!

UTXO: Unspent transaction output

Idea: No balances but coins:

- for each coin:
store pk of the owner.
- transfer:
provide signature with owners key
provide pk of new owner

Data structure:

Array $[pk_0, pk_1, pk_2]$

Q: What should you sign:

Ex: Send coin 0 to pk_4

$\text{sign}(0 \parallel pk_4)$

Problem: Replay if pk_1 owns coin 0 again.

$[pk_4 \mid pk_1 \mid pk_3]$

Solution: Datastructure map/dict

$\{a \rightarrow pk_1, b \rightarrow pk_1, c \rightarrow pk_3\}$

$\downarrow \text{transfer}(a, pk_4, \underset{\substack{\text{sig}_{pk_1} \\ \text{sig}_p}}{\text{sig}("a" \parallel pk_4)})$

$\{d \rightarrow pk_4, b \rightarrow pk_1, c \rightarrow pk_3\}$

Now: Have coins with different value

$\{a \rightarrow (v, pk)\}$ - called output

value

spending condition

output (v, pk)
value v spending condition, usually public key

input (id, sig)
identifier points to one output id redeeming argument (signature) sig

transfer (inputs, outputs)

Example: $\{a \mapsto (2, pk_1), b \mapsto (2, pk_1)\}$

Send 3 to pk_3

inputs $(a, sig_1), (b, sig_1)$

two sigs: can be with different key.

outputs $(3, pk_3), (1, pk_1)$

change coming back.

Q: What to sign?

$$sig_1 = sk_1 \cdot \text{sign}("a" || "b" || 3 || pk_3 || 1 || pk_1)$$

Addresses vs. UTXO

+ One balance for one user

- Easy to track one user

- One user has multiple outputs

+ Easy to use multiple private keys.

(Default \rightarrow create new pk for change.)

+ No storage used for empty accounts.

Spending condition:

- Small redeeming conditions save storage
- Programmable redeeming conditions allow advanced payment control.

Examples:

	Redeeming condition	argument
a)	public key	signature
b)	hash	public key (hashes to condition and signature)
c)

c)	5 public keys	signature with 5 of the 5 public keys.
d)	script	argument that makes script return true

Note: Base 56 encoding ← for noting down
 on paper
 Error code