Writing Smart Contracts
03 Accounts

Peter H. Gruber

Supported by the Algorand Foundation

Algorand Adesses

(1) Private key

- A very long number . . .
- 256 Bit $= 2^{256} \approx 10^7 7$ different possibilities
- "Master password to account", "Single Factor Authentication"

(2) Mnemonic

- 25 words out of a list of $2048 = 2^{11}$ words
- 1 word $=$ 11 Bit
- 24 words $= 264 > 256$ Bits
- Algorand uses 25th word as checksum

(3) Address = public key

- Hash (Ed25519) of private key
- Algorand: 256 Bit $+$ 32 Bit Checksum
- Easy: private → public
- (Almost) impossible: public → private

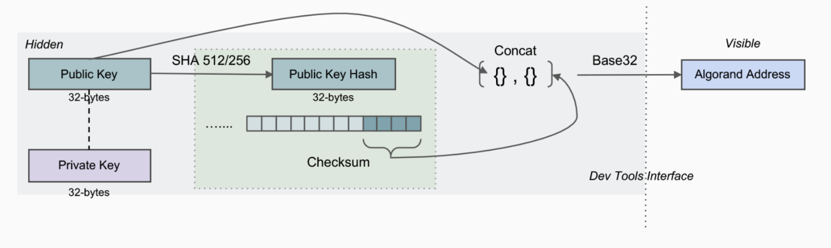
(4) **Wallet** = collection of keys

Public Key = Address

Transformations

- Add 4 Bytes = 32 Bit Hash
- Encode as numbers/letters for readability
- 56 numbers/letters, 5 Bytes each = 280 Bits

`N72FLVBF2PW6SKXNDW6JLZT5WUACHGIDVZI3OPUCK2ALFUHO3KURCNRODE`



Public Key to Algorand Address

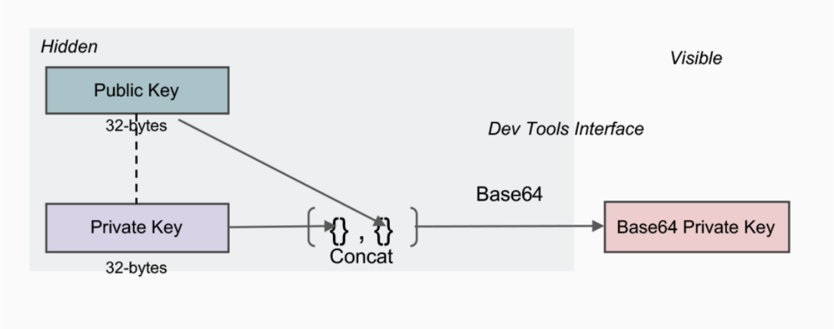
Private Key

Transformations

- Store Public and Private Keys
- Encode as numbers/letters for readability
- 80 numbers/letters, 6 Bytes each = 280 Bits
- For developpers

VwrmAkisLya/0H+HALB13XRpLNGfkoMY4mgUXYL6FURv

9FXUJdPt6Srt HbyV5n21AC0ZA65Rtz6CVoCy007aqQ==



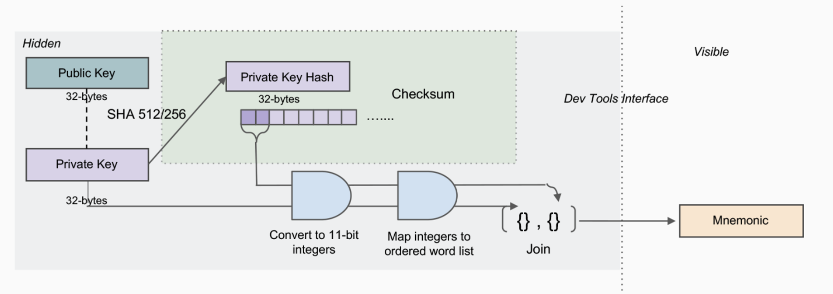
Base64 Private Key

Passphrase = Mnemonic

Transformations

- Store Private Key
- Encode as words from a list
- 25 words, 11 Bytes each = 275 Bits
- For end users

```
enough oblige accident setup gap sister magnet lemon axis scale river
evidence spray enrich write myth away mask crucial spend again leaf camera
able athlete
```



Accessing the blockchain

Where is the Algorand chain?

- On approx. 120 relay nodes (Nov 2021)
- One of them at USI
- On (many) indexer nodes world-wide

How large is the Algorand Chain?

- Approx. 920GB
- Up-to-date: `https://howbigisalgorand.com/`

How can we access the chain?

- Required to make transactions and verify them
- Set up our own indexer node
- Access via API, e.g. purestake.io

An Algorand transaction

```
{
  "txn": {
    "amt": 5000000,
    "fee": 1000,
    "fv": 6000000,
    "gen": "mainnet-v1.0",
    "gh": "wGHE2Pwdvd7S12BL5FaOP20EGYesN73ktiC1qzkkit8=",
    "lv": 6001000,
    "note": "SGVsbG8gV29ybGQ=",
    "rcv": "GD64YIY3TWGDMCNPP553DZPPR6LDUSFQOIJVFDPPXWEG3FVOJCCDBBHU5A",
    "snd": "EW64GC6F24M7NDSC5R3ES4YUVE3ZXXNMARJHDCCCLIHZU6TBEOC7XRSBG4",
    "type": "pay"
  }
}
```

Python commands

Transactions

- Local
  1. Prepare/create transaction → `txn`
  2. Sign transaction → `stxn`
- On Chain
  3. Send transaction → `txid`
  4. Verify transaction → `txinfo`

Accounts

- Local
  - Create key pair
- On Chain
  - Get account balance