Wallet Security

Wallets

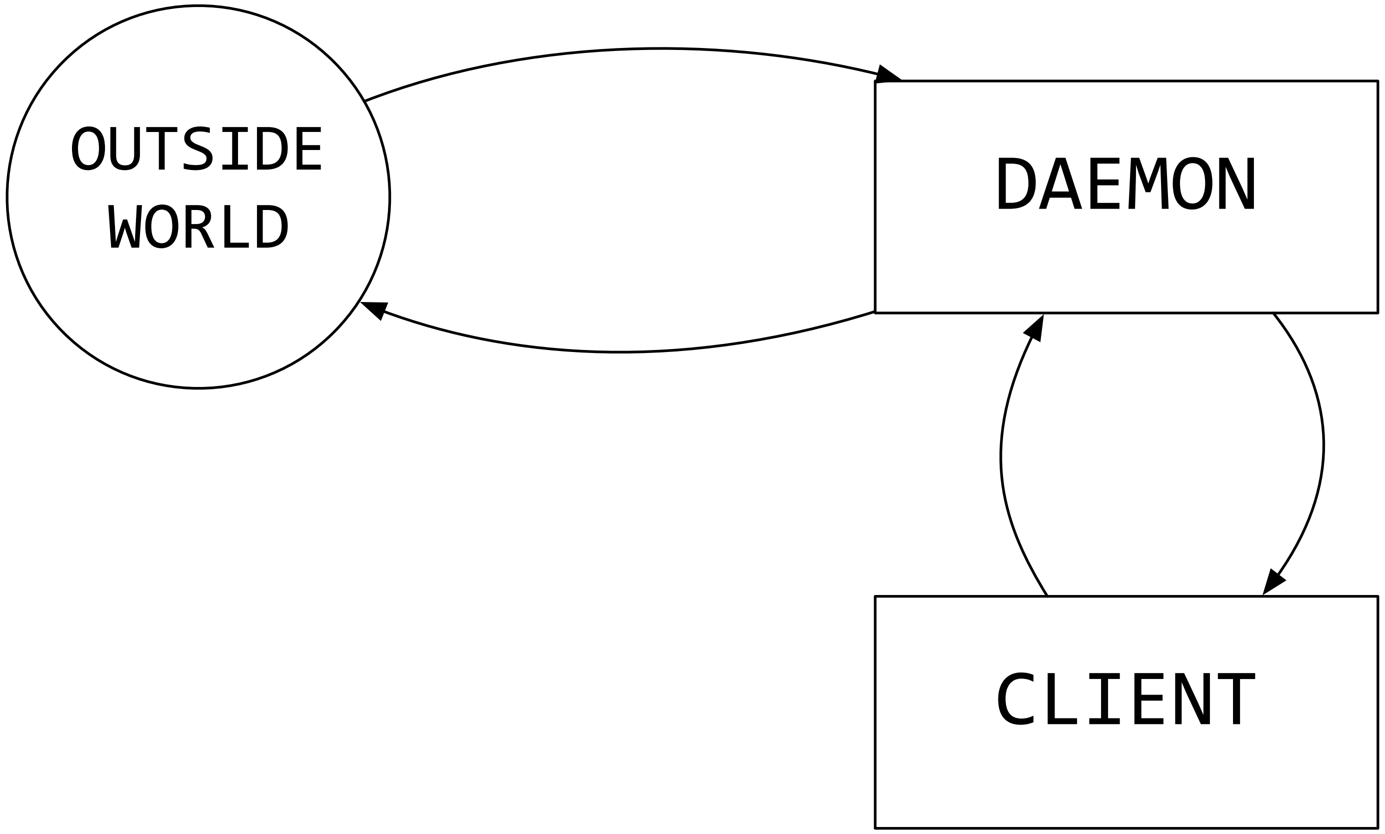
- Keep track of the world
  - If you want
- Synchronize with the network if you fall behind
- Address end user needs
  - Send coin
  - Receive coin
  - Answer queries
    - What is my balance?
    - What is my activity history in this network?

This Lecture

- How do you engineer safe wallets?

Architecture

- Daemon, client architecture
- Daemon:
    - Long running
- Client:
    - CLI or GUI that talks to daemon
    - Short lived process


OUTSIDE WORLD
DAEMON
CLIENT

Followed By

- Armory
- Coinbase
- Parity Daemon

Attack Surface

- Key handling:
  - Client / daemon responsible
- Communication:
  - Are messages designed correctly
- Crypto:
  - Are you doing things right

Daemon Client Communication

- How do they communicate?
  - IPC
    - TCP, Sockets, Message queues…

What About HTTP

- A small example:
  - GET http://localhost:8000/balance
  - POST http://localhost:8000/send
  - GET http://localhost:8000/history

Flow

- Client makes HTTP requests to Daemon
- Issues?

Issues?

- **Anyone** can make those requests
- If you load a webpage, that webpage can issue requests to **http://localhost:8000**

History

- Zoom:
  - Video conferencing product
  - Recent successful IPO

Zoom Daemon

- The Zoom software ran a daemon on http://localhost:PORT
- Visiting https://zoom.us/j/meeting-id
  - Would cause the webpage to issue a request to the localhost server
  - Which would:
    - Join the user to a call
    - Update the zoom client
    - etc.

Zoom Daemon

- Further:
    - Buffer overflows in this undocumented web-server

Zoom Daemon

- Users figured this out
- Vuln was demonstrated using a third party website that:
  - Could join a random user into a zoom meeting that they didn’t want to join
  - Install a zoom client without their interaction

For Your Wallet

- Any third party page can figure out:
  - What’s your balance
  - What sort of transactions you’ve conducted in the past
  - Etc.

Doing It Right

- Well tested architectures:
  - Docker daemon + client:
    - Unix domain socket for IPC on OS X, Linux
    - TCP on windows
    - Avoids our http exploit

Links

- https://medium.com/bugbountywriteup/zoom-zero-day-4-million-webcams-maybe-an-rce-just-get-them-to-visit-your-website-ac75c83f4ef5

Protocol

- You can secure comm layer
- But what you send over the wire can still cause problems

Example

- Daemon / Client
- Client issues request:
  - {recipient: ABC-DEF-…, AMOUNT: 100}
- Daemon signs and broadcasts

Protocol

- Any other process can do that too

MISC

- You can log things like keys
  - Happens even now at large firms
- Coredumps

Coredumps

- You can dump a running process to disk
- And inspect the state
- If you keep the keys loaded in memory, they can be found there

Crypto

- Bitcoin wallet
  - Private keys stored in wallet.dat
  - AES-256 encryption of these private keys
  - Master key:
    - Passphrase -> SHA 512

Deterministic Wallet

- Seed Phrase
    - Wallets contain a wordlist:
        - 2048 words mapped to integers
        - Pick 12 random words from this wordlist

Seed Phrase

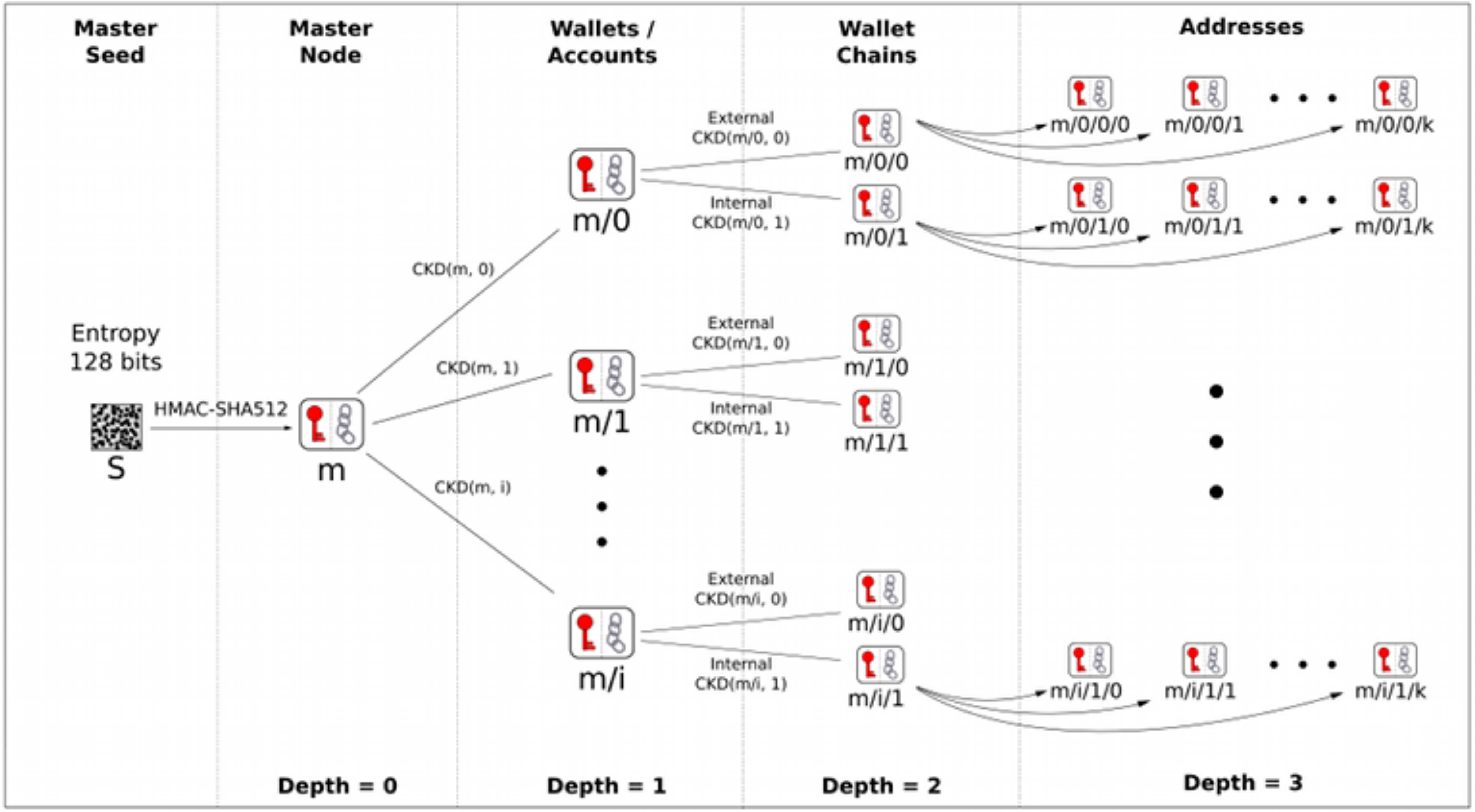
- This is your seed phrase:
  - 2048 ^ 12 combinations
  - 12 word seed phrase has about 128 bits of security

Seed Phrase

- Write down this 12 word list
- It is sufficient to recover your bitcoin

HD Wallet

BIP 32 - Hierarchical Deterministic Wallets



Child Key Derivation Function ~ CKD(x,n) = HMAC-SHA512(x_{Chain} , x_{PubKey} || n)

HD Wallet

- Single Seed Phrase for all private keys
- Master Public Key:
  - Generate from Master Private Key
  - Can generate all additional public keys but not their private keys
- Private Keys from the Master Private Keys are Master Private Keys themselves.

HD Wallet

- Computing n^th private key:
  - Compute an offset: H(n, Master PubKey)
  - Master Private Key + offset

HD Wallet

- Computing n^th Master Public Key:
  - Compute an offset: H(n, Master PubKey)
  - Master Public Key + get_pubkey(offset)

Hierarchy

- Root of pub / priv keys

Key Best Practices

- Brand new address to receive each payment
- Ask for a brand new address from the recipient

Threshold Signatures

- Constructing a single signature is:
  - Split between two devices
  - Single device won’t be at risk

Threshold Signatures

- Each party (device) creates a key independently
- A signing protocol
    - Each share does part of the signing

Hardware Wallets

- BitFI “Unhackable” Wallet

Exploits

- Can easily read finger movements on device
  - Taps etc.
- Can read out data being sent
- Can easily tamper with the device