whoami

- Bryan Bishop
- Software development background
- Previously @ LedgerX (4 years!)
- Bitcoin Core contributor
- Biotech projects
- Transcripts - https://diyhpl.us/wiki/transcripts
- Follow me @ https://twitter.com/kanzure
Conflict of interest slide (Just kidding)

Academic support

digital currency initiative
Stanford Cyber Initiative
BSafe network
BASE ALLIANCE
Imperial College London

TU Wien Security & Privacy Research Division
Technion Israel Institute of Technology
IC3 The Initiative For CryptoCurrencies & Contracts

Other supporting orgs

BC²
DG LAB
bitembassy.org

101.lab

Be'er Sheva Blockchain community
What is an HD wallet, briefly?

- Indeterministic wallet: random keys every time
  - Lots of horror stories of lost keys
  - Remember to backup your wallet!
  - Sweep to new change address -> ded.

https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
Deterministic wallet: random key, but only once

- Backup the random master seed
- Armory originally implemented deterministic wallets, but it was a giant pool of addresses
Hierarchical deterministic (HD) wallets

- Backup the master private key (master seed)
- Many deterministic subwallets
- Address reuse bad
Previously on Bitcoin Edge Dev++ (2018)

- James Chiang gave an excellent presentation on bip32:
  - video: https://www.youtube.com/watch?v=OVvue2dXkJo

- Most of the diagrams in this presentation are from his slides.
bip32 overview

BIP 32 - Hierarchical Deterministic Wallets

Master Seed → Master Node → Wallets / Accounts → Wallet Chains → Addresses

Entropy 128 bits

Child Key Derivation Function: \( \text{CKD}(x, n) = \text{HMAC-SHA512}(x_{\text{Chain}} , x_{\text{PubKey}} \| n) \)
bip32 Child Key Derivation Paths (bip32 paths)

Hierarchical Deterministic Wallets (BIP32)

Master Keys

0 → Child Keys /0

1 → Child Keys /1

2 → Child Keys /2

0 → Child Keys /0/0

1 → Child Keys /0/1

2 → Child Keys /0/2

0 → Child Keys /1/0

1 → Child Keys /1/1

2 → Child Keys /1/2

0 → Child Keys /2/0

1 → Child Keys /2/1

2 → Child Keys /2/2

3, 4, ...

HD wallets (BIP32) can deterministically derive an indefinite number of fresh addresses from a single wallet secret.

HD Tree
- Fresh addresses to improve privacy.
- HD Tree is derived from Master Keys.
- HD Tree can be reconstructed from master Keys (given tree structure).

Master keys
- Derived from HD root secret.

Subtrees
- Allow separation of keys for accounts/usages.
- Selective key sharing.
More bip32 paths

- m/0
- m/1/2/3/1, m/1/2/3/2, m/1/2/3/3, m/1/2/3/4, etc.
- m/1/2'
- m/1/2'/3'
- m/1/2'/3/4'
- m/1'/2/3'/4/5'
- etc...
Master key pair derivation

HD Root Seed (128 - 512 Bits)

HMAC-SHA512 ("Bitcoin Seed", HD Root Seed)

Master Private Key (L256bits)  Master Chain Code (R256bits)

Private Key * G

Master Public Key

The master key pair is derived from the HD root secret, and together with the chaincode, provides the basis for deriving subsequent child key generations.

HMAC-SHA512
- 512 bit hash digest is split into left and right 256 bits.
- Right 256 bits are chaincode, used in child key derivation.
Hierarchical deterministic (child) private keys are derived from parent private keys.

**HMAC SHA512**
- **Key:** Parent chaincode
- **Data:** Parent public Key || Index

**Addition of two 256bit scalars**
- Private key + L256
- Result: Child private key

**Parent public key to child public key**
- HD child public key derivation without parent private key.
Child key pair derivation, part 2

Hierarchical deterministic (child) private keys are derived from parent private keys.

HMAC SHA512
- Key: Parent chaincode
- Data: Parent public Key || Index

Addition of two 256bit scalars
- Private key + L256
- Result: Child private key

Parent public key to child public key
- HD child public key derivation without parent private key.
Hardened HD children

- **Parent Public Key**
- **Parent Chain Code**
  - **HMAC SHA-512** (Parent Chain Code, Parent Public Key || Child Index)
- **Child Index**
- **Parent Private Key**
  - L256 bits
- **Scalar Addition**
- **Child Private Key**
  - (Parent Private Key + L256bits) * G
- **Child Public Key**
- **Child Chain Code** (R256 bits)

**Child private key hardening**
- Parent public key replaces private key.
- HMAC512:
  - Key: parent chaincode
  - Data: 0x00 || private key || index
- Hardened Index Notation:
  - \( i' = i + 2^{31} \)

**Hardened public keys.**
- Cannot derive any children.
- Derived only from hardened parent child key.
Hardened HD children

Child private key hardening
- Parent public key replaces private key.
- HMAC512:
  - Key: parent chaincode
  - Data: 0x00 || private key || index
- Hardened Index Notation:
  - $i' = i + 2^{31}$

Hardened public keys.
- Cannot derive any children.
- Derived only from hardened parent child key.
Vulnerability: Private key exposure

- This is where hardened keys are useful.
- Note that exposing any child private key and the chaincode results in leaking the private key for the higher levels.
- Hardened keys mitigate this. Hardened keys protect their parent nodes from "downstream" leaks.
Wallets & gap limits

- Gap limit – number of addresses in bip32 tree to check when scanning the blockchain.
- When to stop scanning?
- What if a user hands out a receiving address for a specific purpose, and then, since the user is a good user and doesn't reuse addresses, never uses it again? There can be large "gaps".
- Gap limit is usually set to 20 but it's arbitrary.
Quaint "use cases" specified in bip32 spec

- Some of these seem almost historical, has anyone read these in a while? Take a look....
- "Full wallet sharing" between nodes that both need to spend coins (what?)
- Audits: share extended public keys and the auditor can derive child addresses.
- Recurrent business-to-business transactions: Use an incrementing bip32 path off of some bip32 key.
- Receive-only wallet, like for a merchant's online webserver selling some items.
Software libraries

- bx tool (libbitcoin)
- pycoin (python)
- bcwallet (python)
- hdkey (javascript)

- I usually go with "pycoin". Buyer beware for the others; I don't have specific experience with those.
bip44

- "Multi-account hierarchy for deterministic wallets"
- Basically, a standard that recommends a specific hierarchy and labeled purposes for each level in the hierarchy.
- It's basically this:
  - m / purpose' / coin_type' / account' / change / address_index
- https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki
Hierarchical Deterministic Wallets

Tel Aviv, Israel
September 2019

Bryan Bishop <kanzure@gmail.com>

0E4C A12B E16B E691 56F5 40C9 984F 10CC 7716 9FD2

https://twitter.com/kanzure
https://twitter.com/thebitcoinedge