CS 168: Blockchain and Cryptocurrencies

Wallets

Prof. Tom Austin
San José State University

Lab Review

What is a wallet?

- Collection of public/private key pairs
- Types of wallets:
  - Software wallets
  - Paper wallets
  - Hardware wallets

Wallet in HW 2

- Array of keys/addresses
  - `[{ address, {public,private} }]`
- Addresses derived from keys
  - Storing key pairs alone is sufficient
- JBOK wallet
  - JBOK: "Just a Bunch of Keys”

`setupWallet` Method
(from utxo-mixin.js)

```
function() {
   this.wallet = [];
   this.wallet.push({
      address: this.address,
      keyPair: this.keyPair });
}
```

How Keys Are Generated

- Nondeterministic (random) wallets
  - "Just a Bunch of Keys" (JBOK)
  - Hard to manage
  - All keys must be stored
  - Leads to key reuse
- Determinitistic (seeded) wallets
  - Derived from master key (the seed)
  - Only need to remember seed

Random Seeds

- Random 512-bit seed is generated.
- Process uses PBKDF2 key stretching algorithm.
- Salt is mixed in:
  - Optionally includes a passphrase
  - Can create an arbitrary number of (all valid) seeds

Mnemonics

From https://xkcd.com/992/

Bitcoin Mnemonic

- Proposed in Bitcoin Improvement Proposal 39 (BIP 39)
- Rand sequence is generated
- Sequence expressed as list of English (or other human lang) words
  - Each word represents 11 bits.
  - 24 words encodes 256-bit key + 8-bit checksum

Benefits of mnemonics/passphrases

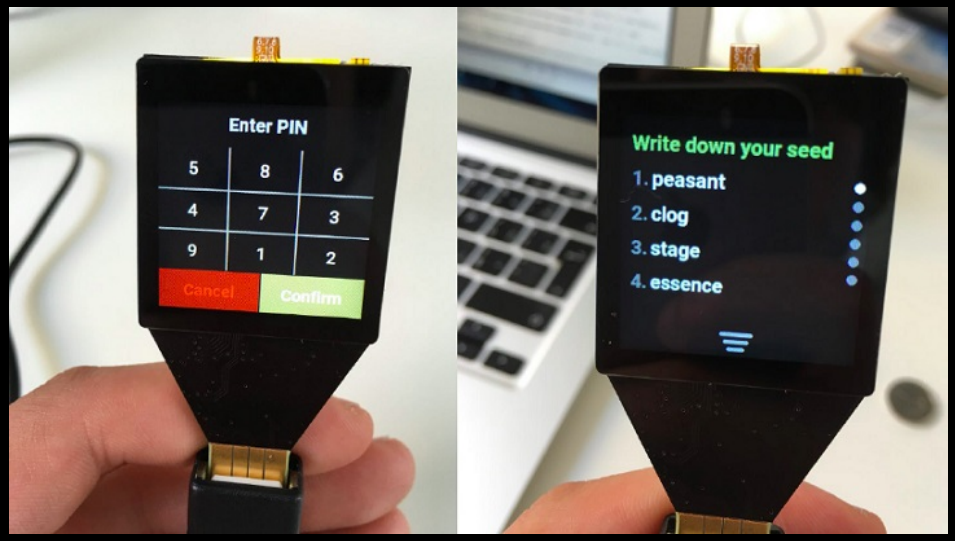
- Can be written down
  - Backup (aka “cold storage”)
  - Avoid storing on computer
- Can memorize passphrase (if desired)
- Passphrases can create many valid wallets with the same mnemonic
  - “Duress wallet”

Simplified Mnemonic
(in class)

- Let R = A37 (hex)
- Let wordlist =
  - apple
  - banana
  - cherry
  - durian
  - eggplant
  - fig
  - grapefruit
  - honey
- **What is the mnemonic phrase?**

Hardware (HD) Wallets

- Private keys never leave device
- Protected with PIN
- 2-factor auth:
  Must have wallet to write transactions
- Protection against malware

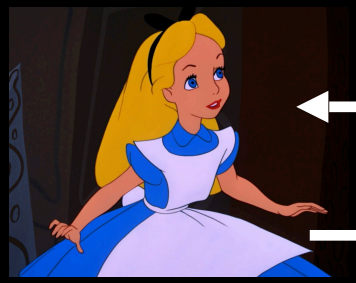


Hardware Wallet Process

1) Issue transaction

2) Forward transaction

5) Send signed transaction



3) Verify transaction

4) Confirm transaction

BIP-32: Hierarchical Deterministic Wallets (Motivation)

- In BTC, need new keypair for every transaction
  - Hard to backup
  - Hard to protect
- Alternate strategy: “chain” of keypairs
  - Deterministic
  - Allows sharing of keys, but …
  - … sharing is all-or-nothing.
- Goal:
  - Selectively share portion of keypairs
  - Require only single seed

BIP-32 design

- Supports multiple chains from single root.
  - Key tree
- Defines how to
  1. Generate keys
  2. Organize wallet
- https://en.bitcoin.it/wiki/BIP_0032

BIP-44: Multi-Account Hierarchy for Deterministic Wallets

- Defines how to handle
  - Multiple coins
  - Multiple accounts per coin
  - Multiple addresses per account
- Designed to be used with BIP-32
- https://en.bitcoin.it/wiki/BIP_0044

Lab: Create a Mnemonics tool

Details in Canvas.

Node Buffer class

- Fixed-length sequences of bytes
- Good for low-level data management
- https://nodejs.org/api/buffer.html

Buffer static methods

- Key static methods:
  - `Buffer.alloc(n)`: creates a buffer of `n` bytes
  - `Buffer.from(d)`: creates a buffer to fit the data `d`
- Key instance methods
  - `buff.writeUInt8(b,p)`: writes byte `b` to position `p` of buffer `buff`.
  - `buff.readUInt8(p)`: reads byte from position `p` of buffer `buff`.