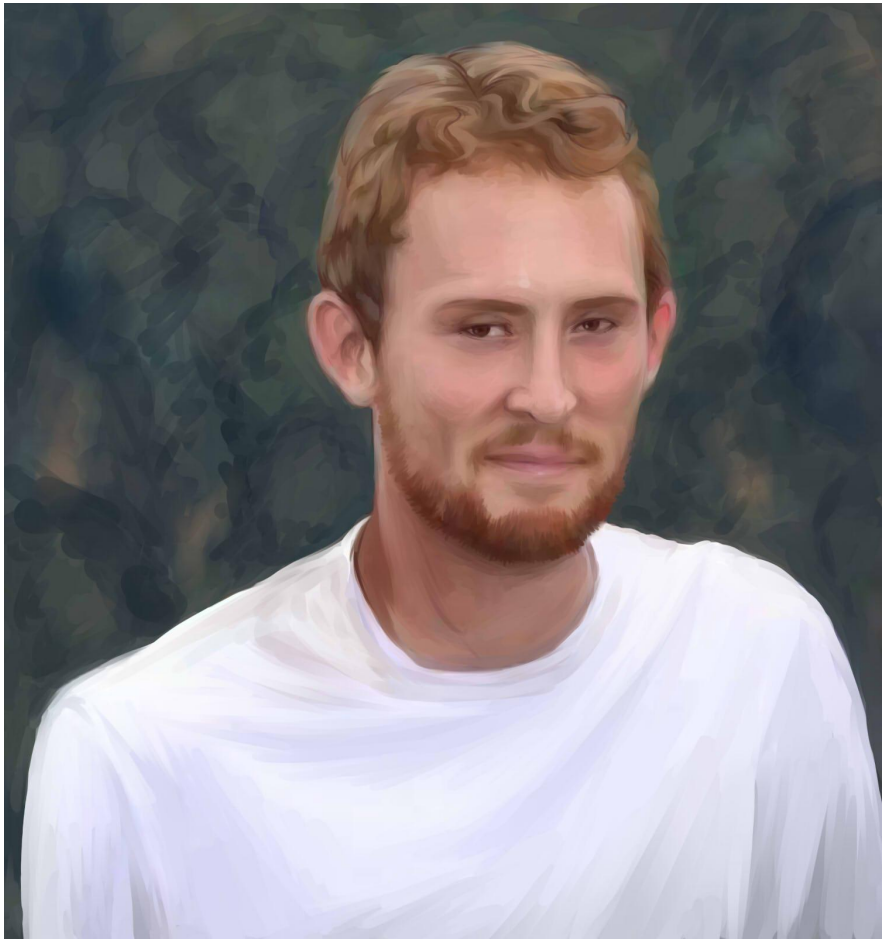
Developing a Multicurrency, Multisignature Wallet

Alex Melville



Alex Melville

Software Engineer

@BitGo

World Traveler

github.com/Melvillian/talks

```
const BitGoJS = require('bitgo');

// Read the user authentication section to get your API access token
const accessToken = process.env.ACCESS_TOKEN;
const bitgo = new BitGoJS.BitGo({ env: 'production', accessToken });

// Create the coin to send requests to BitGo's service
const coin = bitgo.coin('btc');

async function runBitGoExample() {
  // Create a BitGo multisignature wallet
  const walletParams = { label: 'BTC Wallet', passphrase: 'secretpassphase' };
  let response = await coin.wallets().generateWallet(walletParams);
}

runBitGoExample();
```

```
const BitGoJS = require('bitgo');

// Read the user authentication section to get your API access token
const accessToken = process.env.ACCESS_TOKEN;
const bitgo = new BitGoJS.BitGo({ env: 'production', accessToken });

// Create the coin to send requests to BitGo's service
const coin = bitgo.coin('eth');

async function runBitGoExample() {
  // Create a BitGo multisignature wallet
  const walletParams = { label: 'BTC Wallet', passphrase: 'secretpassphase' };
  let response = await coin.wallets().generateWallet(walletParams);
}

runBitGoExample();
```

Security

Multisignature

HD Wallets (BIP 32)

Why Multisignature?

- (Single-signature is less secure)

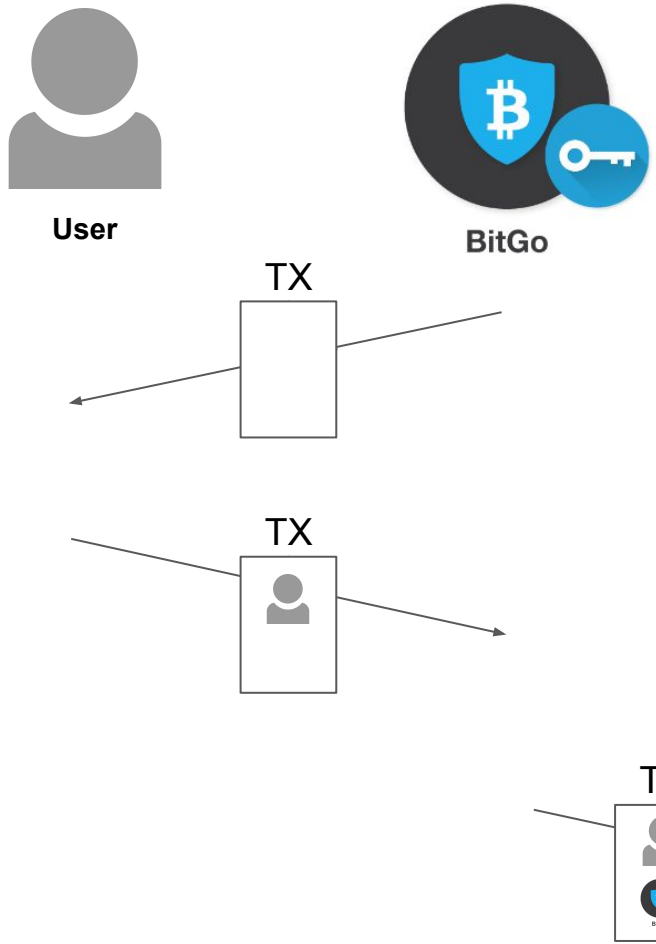
Why Multisignature?

- (Single-signature is less secure)

Coincheck

MT.GOX

- Cosigner and charge for your service



How does single-signature work?

```
var alice1 = bitcoin.ECPair.makeRandom({ network: regtest })
var alice2 = bitcoin.ECPair.makeRandom({ network: regtest })
var aliceChange = bitcoin.ECPair.makeRandom({ network: regtest, rng: rng })

var txb = new bitcoin.TransactionBuilder(regtest)
txb.addInput(unspent0.txId, unspent0.vout) // alice1 unspent
txb.addInput(unspent1.txId, unspent1.vout) // alice2 unspent
txb.addOutput('mwCwTceJvYV27KXBc3NJZys6CjsgsoeHmf', 8e4) // the actual "spend"
txb.addOutput(aliceChange.getAddress(), 1e4) // Alice's change
// (in)(4e4 + 2e4) - (out)(1e4 + 3e4) = (fee)2e4 = 20000, this is the miner fee

// Alice signs each input with the respective private keys
txb.sign(0, alice1)
txb.sign(1, alice2)

// build and broadcast our RegTest network
regtestUtils.broadcast(txb.build().toHex(), done)
```

How does multisignature work?

```
const halfSignedTransaction = '0100000...5e525152';

var bitgo1 = masterKey.derive(alice1.path)
var bitgo2 = masterKey.derive(alice2.path)

var txb = new bitcoin.TransactionBuilder(halfSignedTransaction)

// BitGo signs each input with the respective private keys
txb.sign(0, bitgo1)
txb.sign(1, bitgo2)

// build and broadcast our RegTest network
regtestUtils.broadcast(txb.build().toHex(), done)
```

Multicurrency Multisignature

- https://github.com/bitcoinjs/bitcoinjs-lib

```
const walletParams = { label: 'my wallet', passphrase: 'secretpassphase' };
const response = await coin.wallets().generateWallet(walletParams);
```

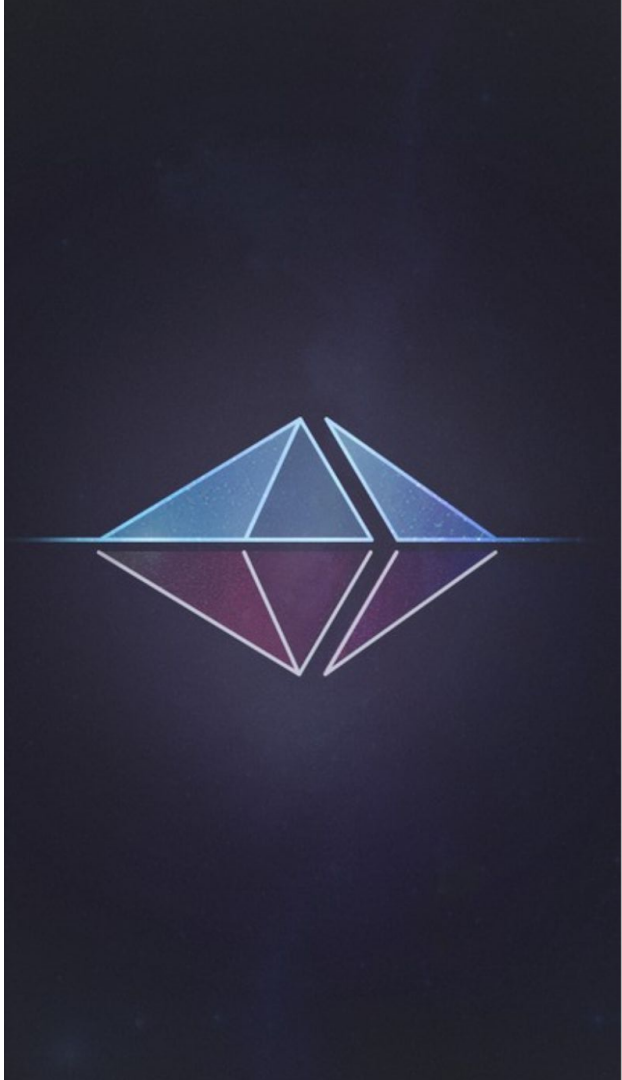
Multicurrency Multisignature

- https://github.com/ethereumjs/ethereumjs-tx
- https://github.com/ethereumjs/ethereumjs-abi
- https://github.com/ethereumjs/ethereumjs-util

- Requires 1 on-chain wallet deploy transaction
  - Makes wallet creation expensive and error-prone

Create Wallet

(1) Contract Deploy



Multicurrency Multisignature

- https://github.com/ripple/ripple-lib
- https://github.com/ripple/ripple-address-codec
- https://github.com/ripple/ripple-binary-codec
- https://github.com/ripple/ripple-keypairs
- https://github.com/ripple/ripple-hashes

- Requires 3 on-chain wallet deploy transactions
  - Makes wallet creation REALLY expensive and error prone

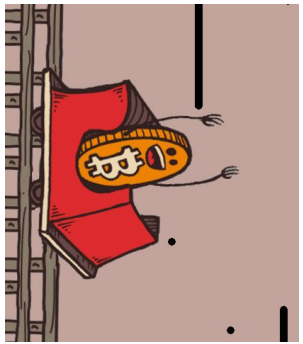
Create Wallet

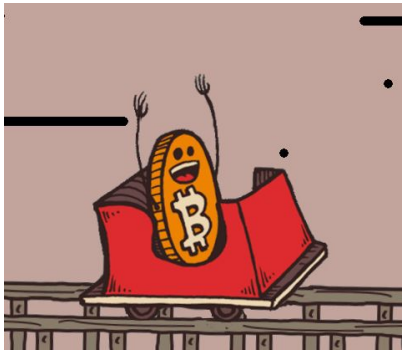
(1) Funding Tx

(2) Multisig Tx

(3) Disable
Master Key Tx

Poll: What is the most important part of any cryptocurrency wallet?





Key Management Problem

- Good wallet will create a new address every time it receives a transaction
  - Privacy

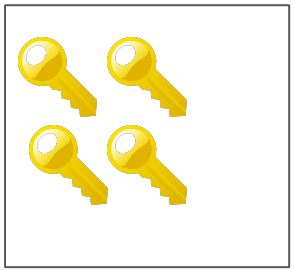
Key Management Problem

- Good wallet will create a new address every time it receives a transaction
  - Privacy

- This means a new private + public keypair for every new address

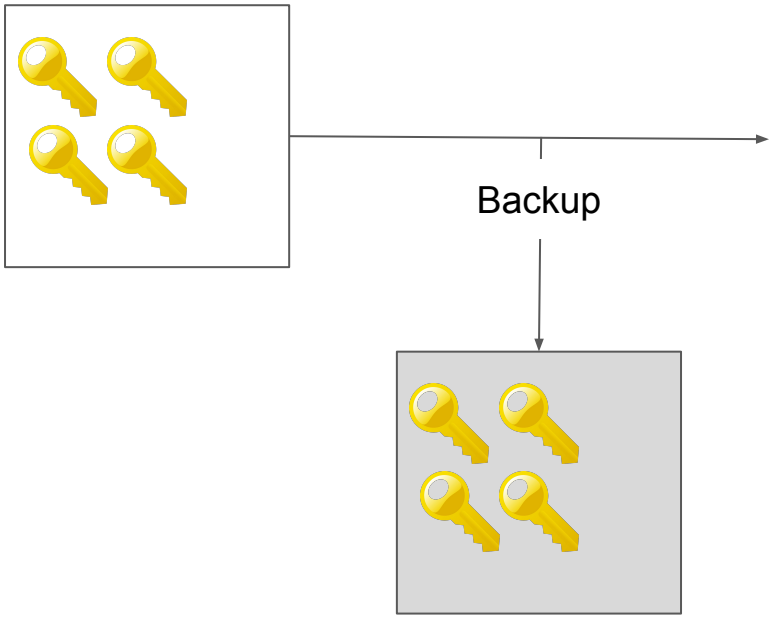
Key Management Problem

- Backups are a pain



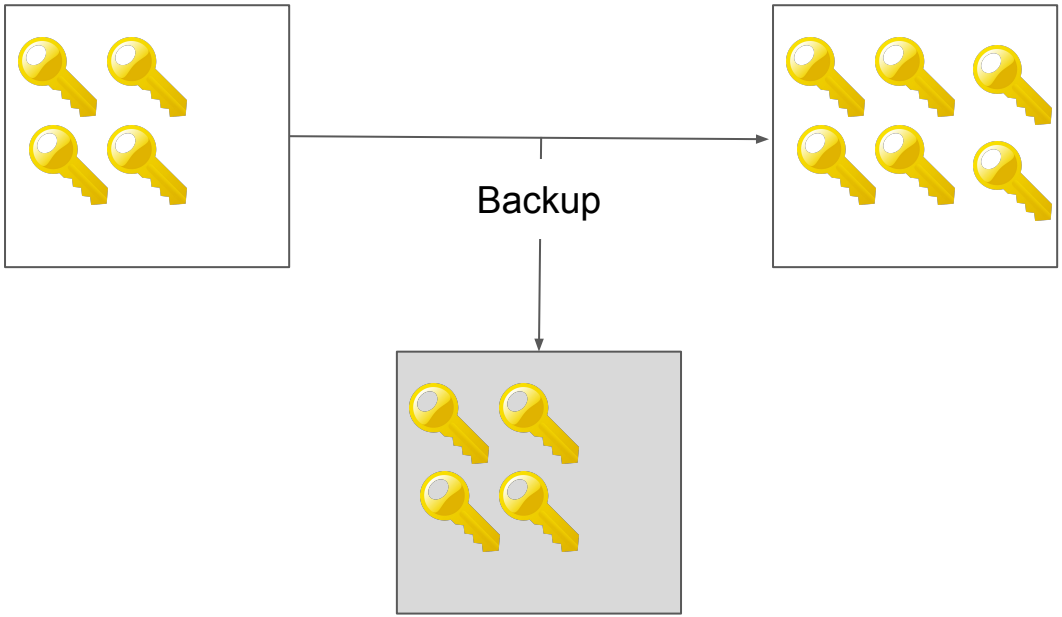
Key Management Problem

- Backups are a pain



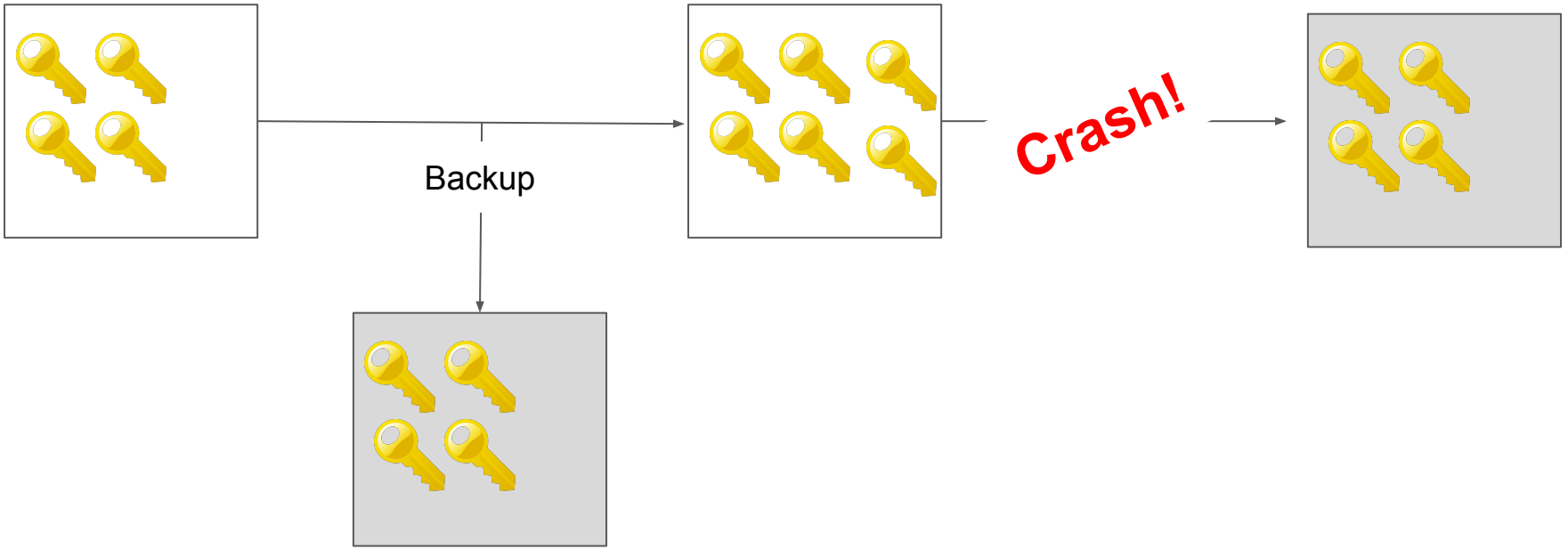
Key Management Problem

- Backups are a pain



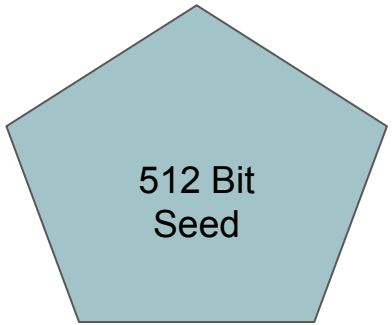
Key Management Problem

- Backups are a pain



Hierarchical Deterministic Keys (BIP32)

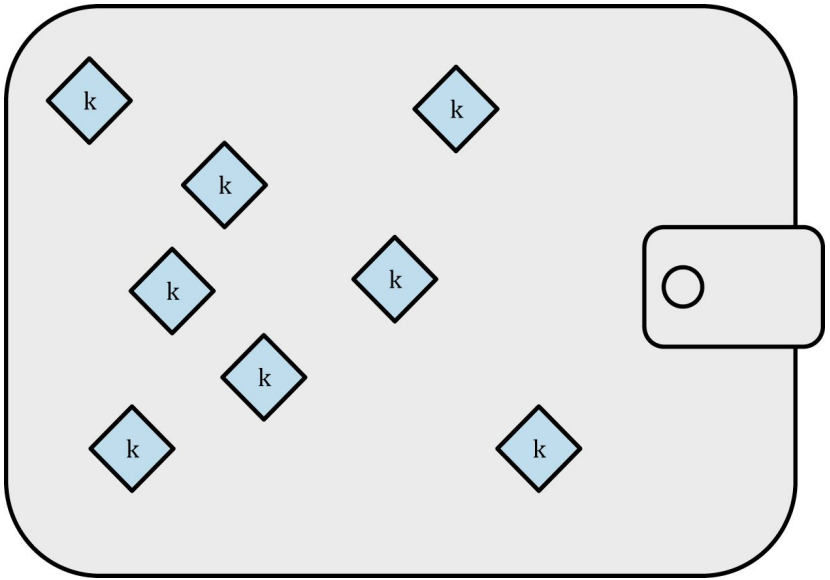
- 1 master seed
- Deterministic method (think hashing) for generating unlimited child keys
  - You only need to remember one 512-bit string



Seed (hex): 000102030405060708090a0b0c0d0e0f

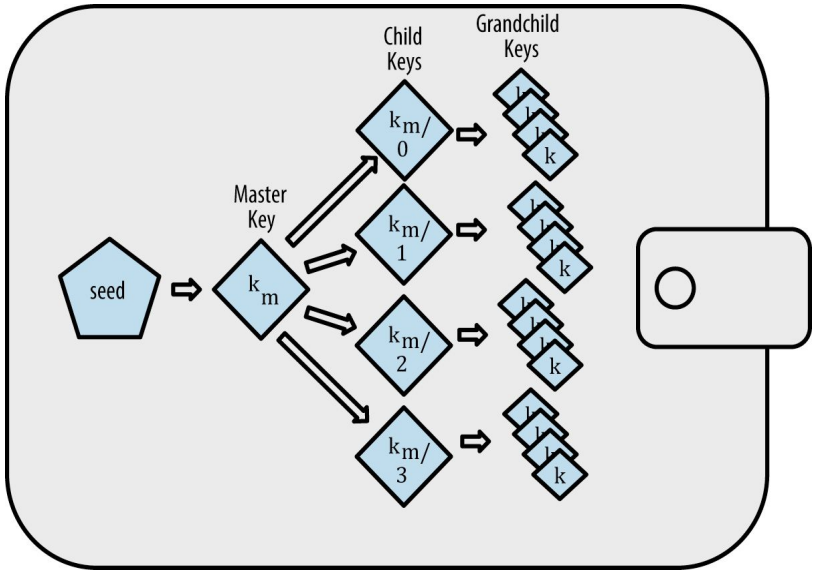
Hierarchical Deterministic Keys (BIP32)

Multiple Private Keys



Hierarchical Deterministic Keys (BIP32)

Single BIP32 Seed



Security Takeaway

- Prevent single key theft
- Be prepared for more development complexity
- Simplify key management with BIP32 keys


Thanks!
ありがとう！