



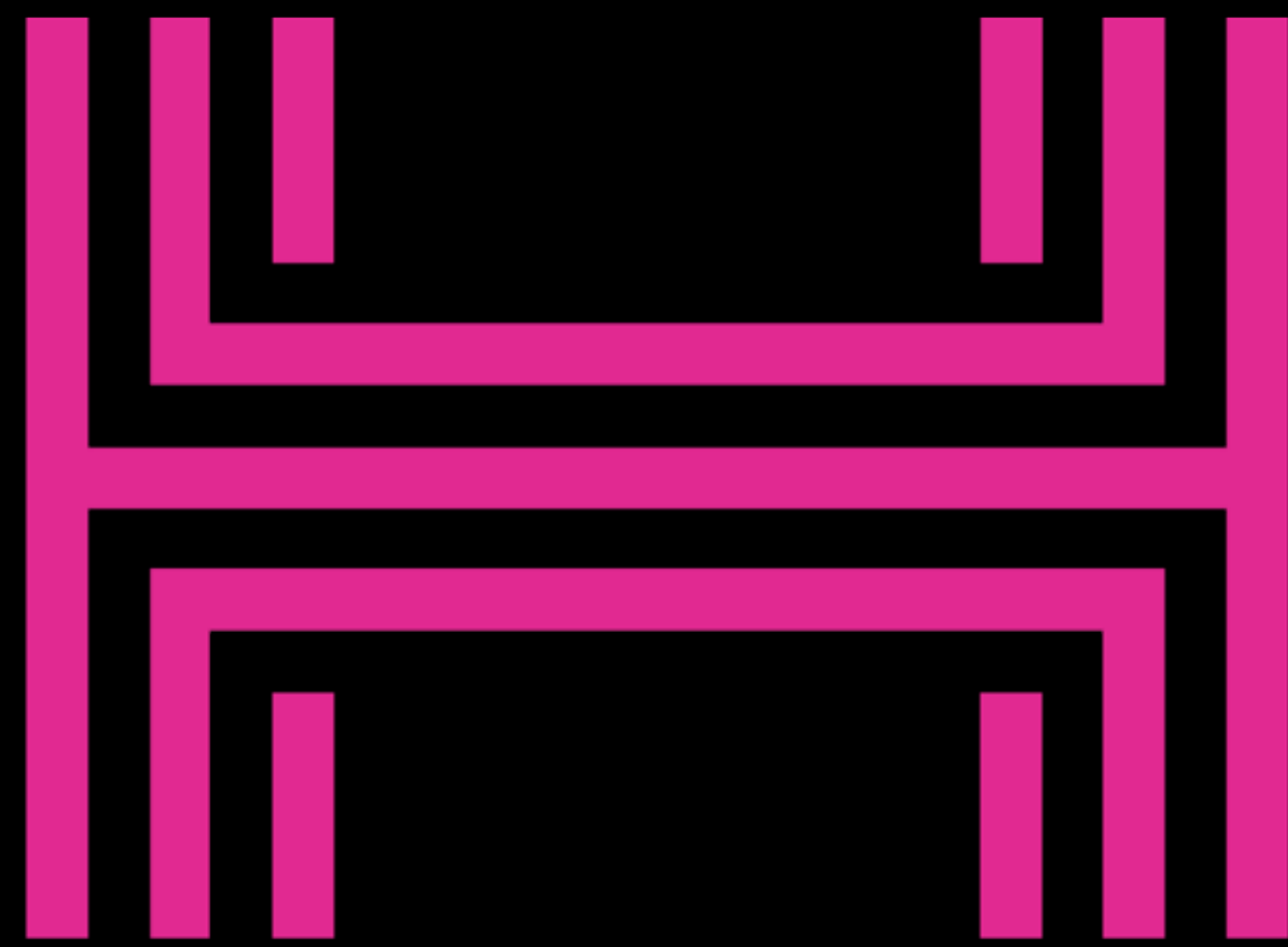
BLOCKCHAIN COMMONS

**SHARDING SECRETS
TO PROTECT YOUR BITCOIN**

WHAT IS BLOCKCHAIN COMMONS?

- ▶ We are a community that brings together stakeholders to collaboratively build open & interoperable, secure & compassionate infrastructure.
- ▶ We design decentralized solutions where everyone wins.
- ▶ We are a neutral “not-for-profit” that enables people to control their own digital destiny.





Human
Rights
Foundation

Thanks to our 2026 Learning Bitcoin Sponsor



Learning Bitcoin from the Command Line

Learning Bitcoin from the Command Line 3.0

Chapter One: Introduction to Learning Bitcoin Core (& Lightning) from the Command Line

Interlude: Introducing Bitcoin

Chapter Two: Creating a Bitcoin-Core VPS

2.1: Setting Up a Bitcoin-Core VPS with Bitcoin Standup

2.2: Setting Up a Bitcoin-Core Machine via Other Means

Chapter Three: Preparing Your Bitcoin Wallet

3.1: Verifying Your Bitcoin Setup

3.2: Knowing Your Bitcoin Setup

3.3: Setting Up Your Wallet

3.4: Understanding the Descriptor Wallet

Chapter Four: Preparing Your Bitcoin Addresses

4.1: Understanding the Address

4.2: Integrating Addresses and Descriptors

4.3: Creating QR Codes for Addresses

4.4: Receiving a Transaction

Interlude: Using Command-Line Variables

Chapter Five: Sending Bitcoin

Learning Bitcoin from the Command Line 3.0

by Christopher Allen and Shannon Appelcline

A Blockchain Commons Course



This work is currently **in process**, but the files linked here have been revised and are in a fairly complete state.

Part One: Preparing for Bitcoin

- 1.0: Introduction to Programming with Bitcoin Core and Lightning
 - Interlude: Introducing Bitcoin

CATEGORY

OTHER RESOURCES

Table of contents

by Christopher Allen and Shannon Appelcline

A Blockchain Commons Course

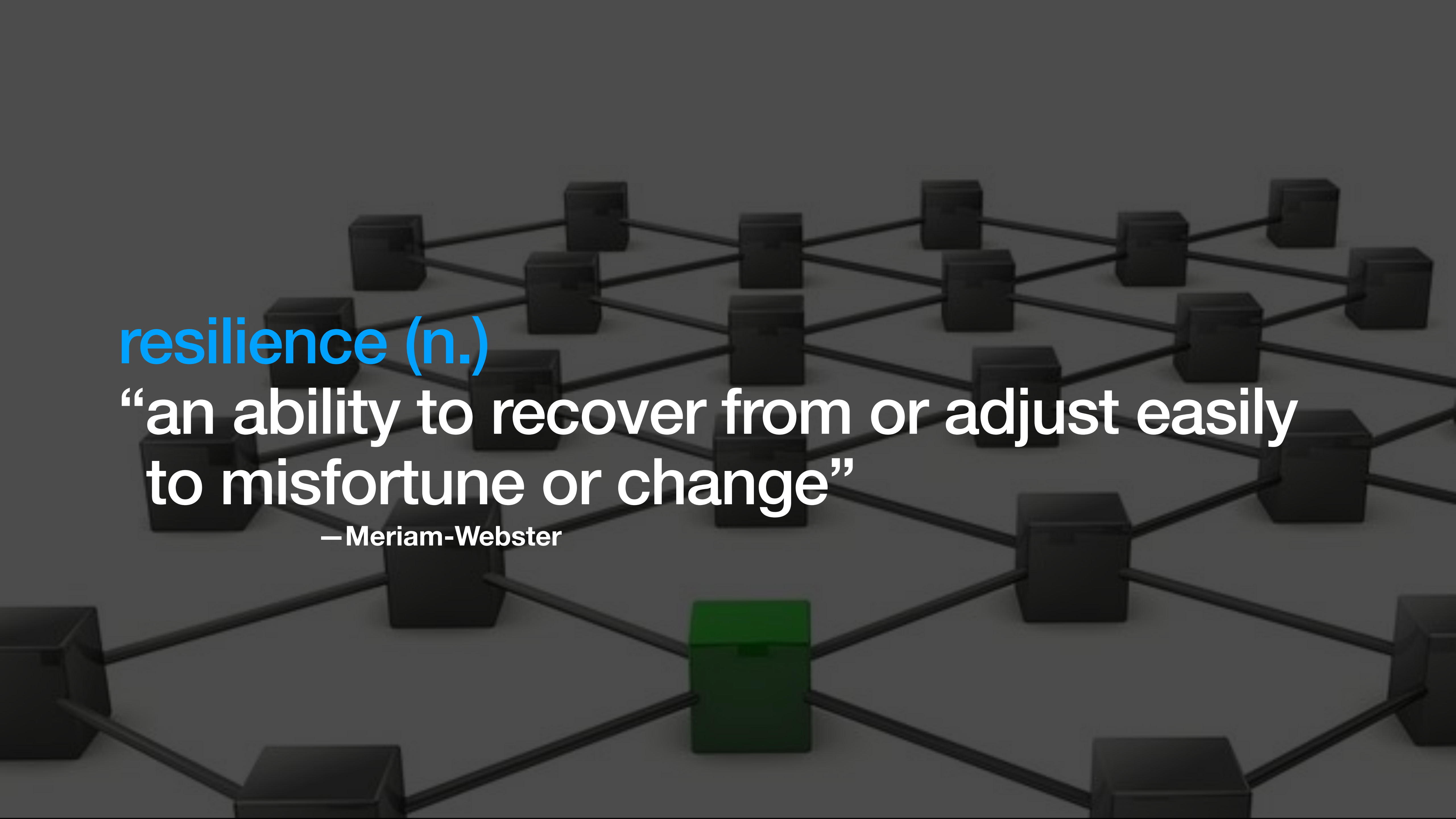
Part One: Preparing for Bitcoin

Part Two: Sending Transactions

Part Three: Expanding Bitcoin

Part Four: Exploring the Ecosystem

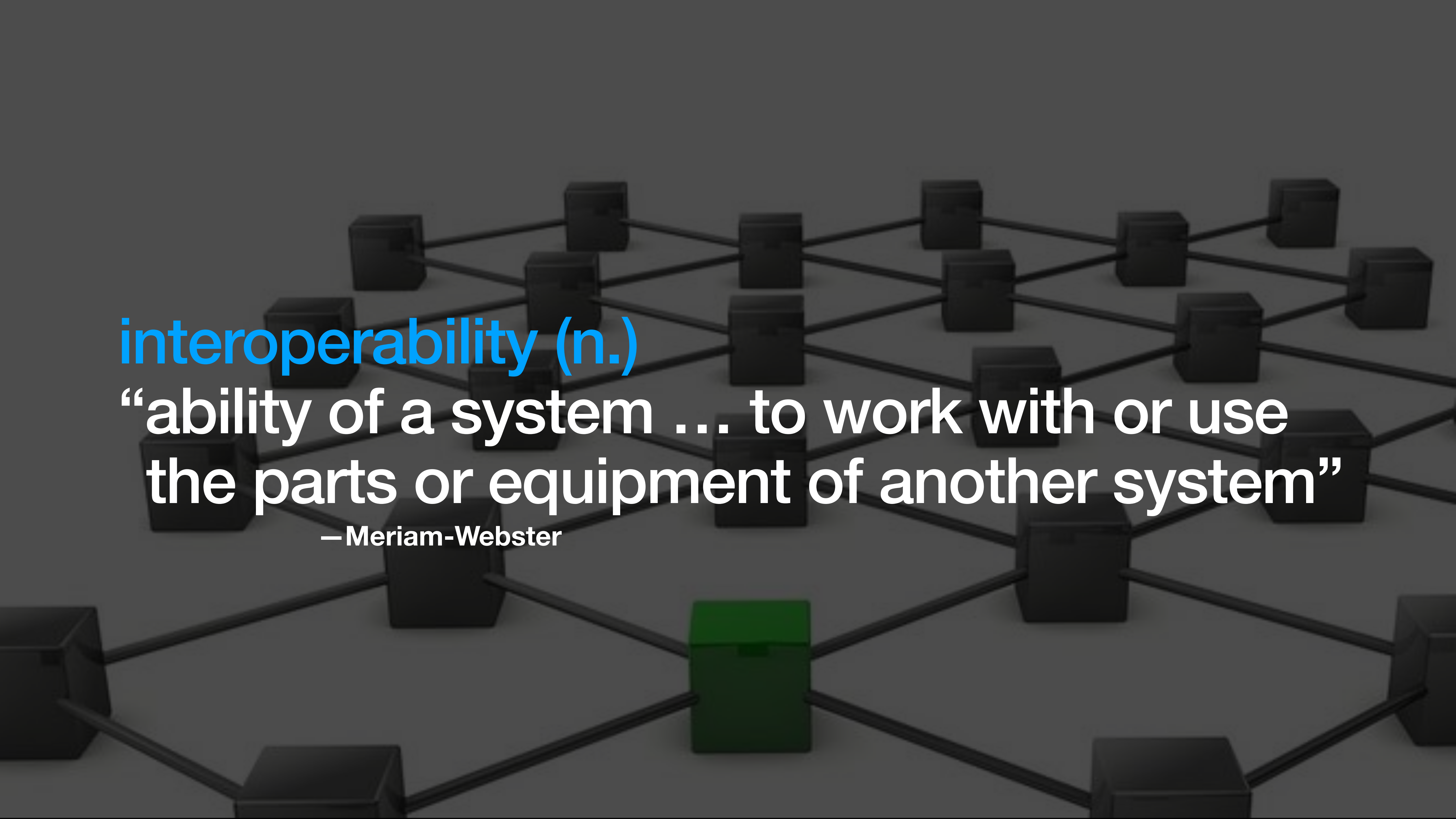
Part Five: Scripting with Bitcoin

A network diagram consisting of numerous black cubes connected by thin black lines. The cubes are arranged in a complex, interconnected pattern. One cube, located near the bottom center of the image, is highlighted in a solid green color, standing out from the rest of the network.

resilience (n.)

“an ability to recover from or adjust easily to misfortune or change”

— Meriam-Webster

A network diagram consisting of numerous black cubes connected by thin black lines. The cubes are arranged in a complex, interconnected pattern. One cube, located in the lower-middle section of the image, is highlighted in a solid green color, making it stand out from the others.

interoperability (n.)

“ability of a system ... to work with or use the parts or equipment of another system”

— Meriam-Webster

Problem #1

Single Points of Failure Threaten Bitcoin

- ▶ Accidental loss of a seed or key is the biggest threat to self-sovereign control of digital assets
- ▶ If you have a single seed or key, that's a single point of failure (SPOF)
- ▶ Better protection is needed

10 years ago, a man lost Bitcoin that's now worth \$100 million

HOME > NEWS > AFTER 12 YEARS OF SEARCHING, HE GIVES UP ON RECOVERING HIS HARD DRIVE WITH \$742 MILLION IN BITCOIN

After 12 years of searching, he gives up on recovering his hard drive with \$742 million in bitcoin

Man has two guesses to unlock bitcoin worth \$240m

13 January 2021

Share

Save

Add as preferred on

Florida Man Loses \$354M Bitcoin Fortune After Lying to Feds; Hard Drive Destroyed

Bitcoin

Florida

Hardware wallet

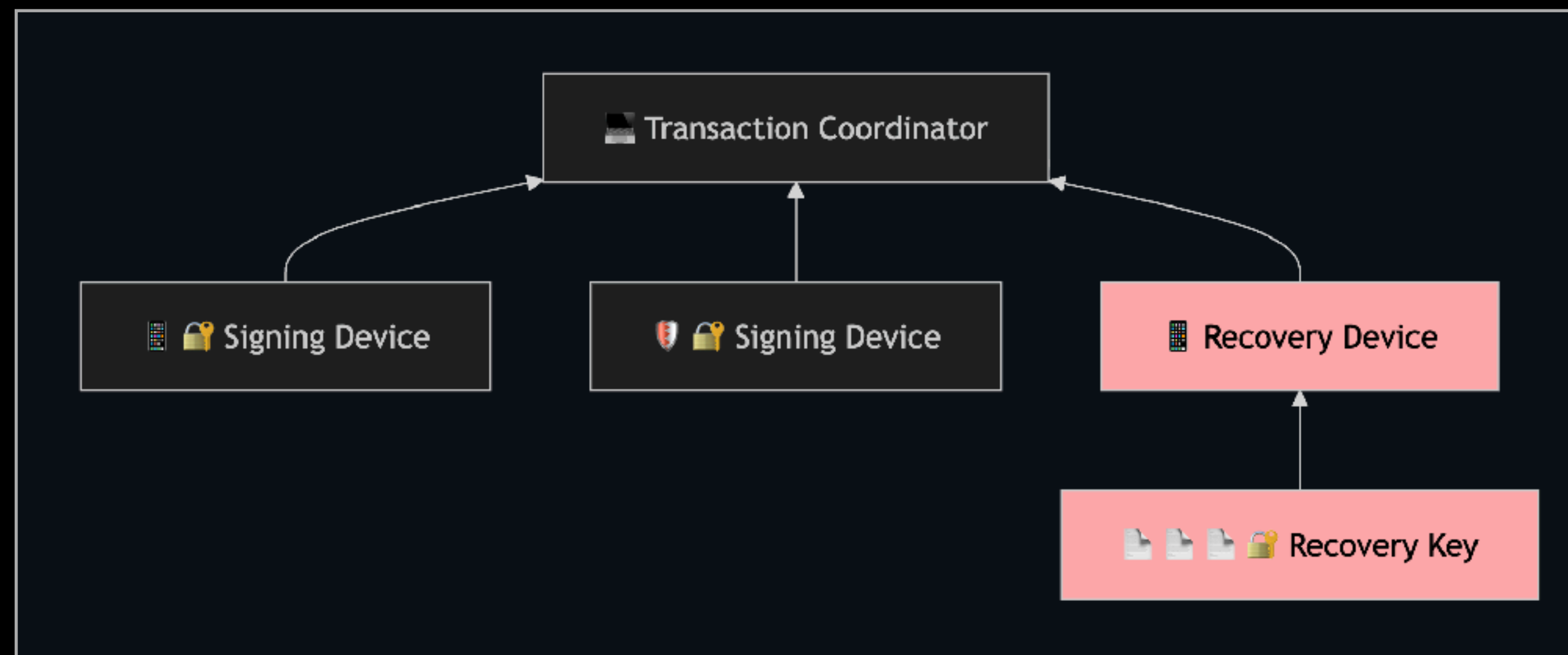
Legal

"For years, Prime denied that he had much Bitcoin at all. And Bitcoin was not on the list when he sought

Multisigs are a Great Solution

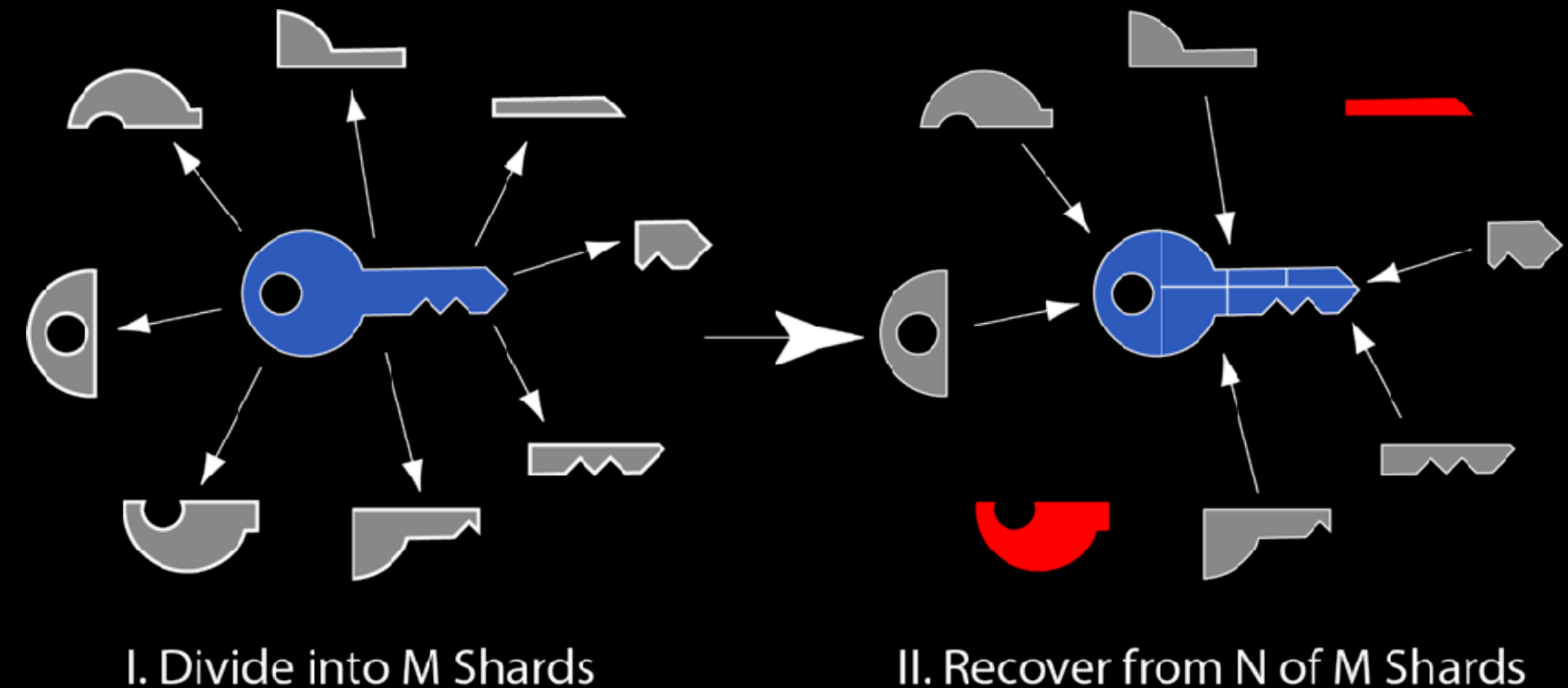
But They're a Pain in the Neck

- ▶ Multisig is a Preferred Answer
- ▶ Create a 2-of-3 multisig
- ▶ Place keys on two signing devices
- ▶ Store a backup key elsewhere



Shamir Offers an Easier-to-Use Alternative

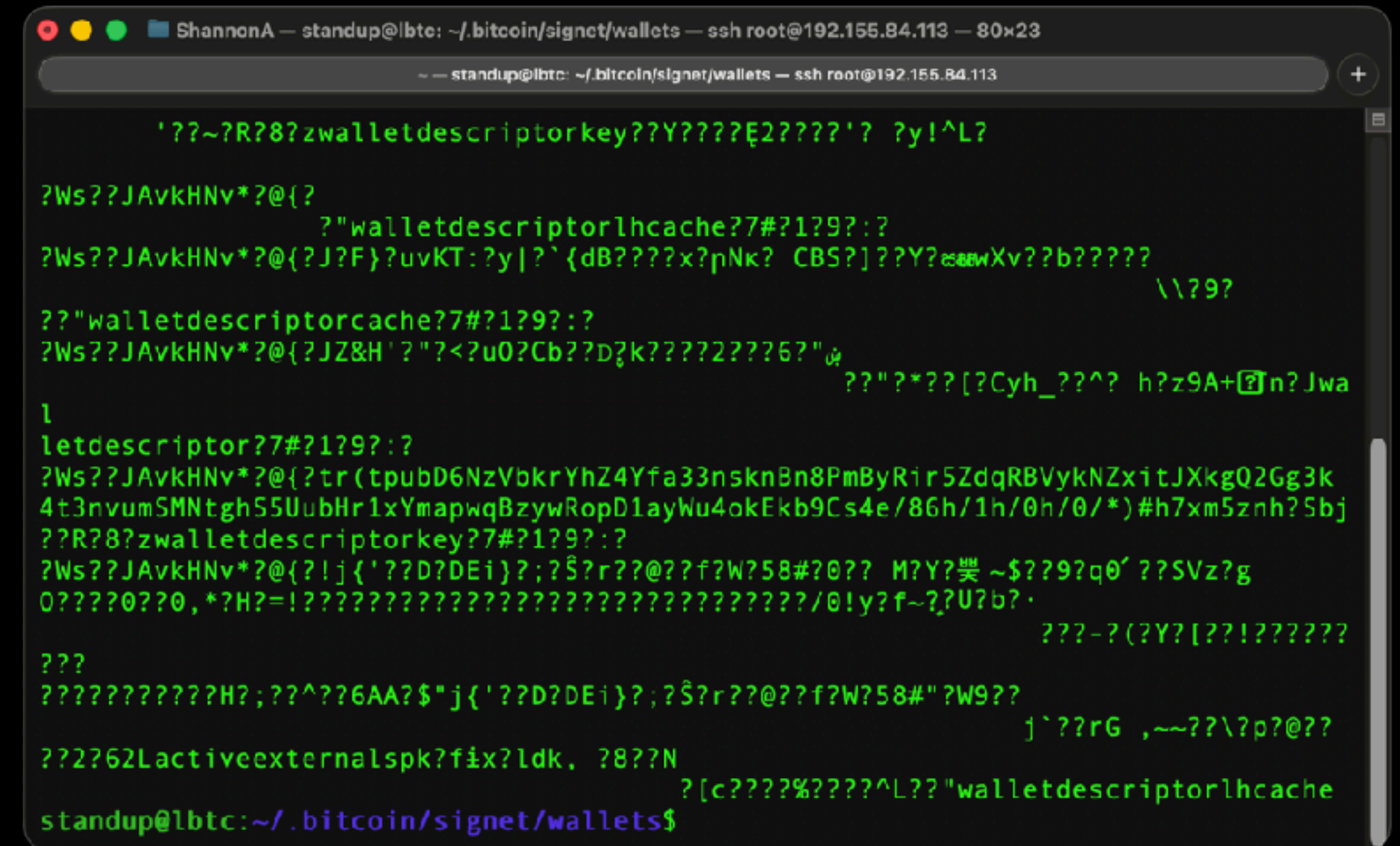
- ▶ That leaves us with Shamir's Secret Sharing
- ▶ You back up your seed as shares
- ▶ You set a threshold for recovery (e.g., 2-of-3)
- ▶ You can lose a share and still recover
 - ▶ But a thief can't do anything with one
- ▶ Reconstruction is dangerous ...
 - ▶ But you can't have everything



Problem #2

Bitcoin Core Doesn't Support Resilience Features

- ▶ No BIP-39 mnemonic or entropy
- ▶ No Shamir's Secret Sharing
- ▶ Only interoperable output is descriptors
- ▶ Keys and derivation paths are kept in an SQL file in Bitcoin Core
 - ▶ You can back it up, sure ...
 - ▶ But you're dependent on file format



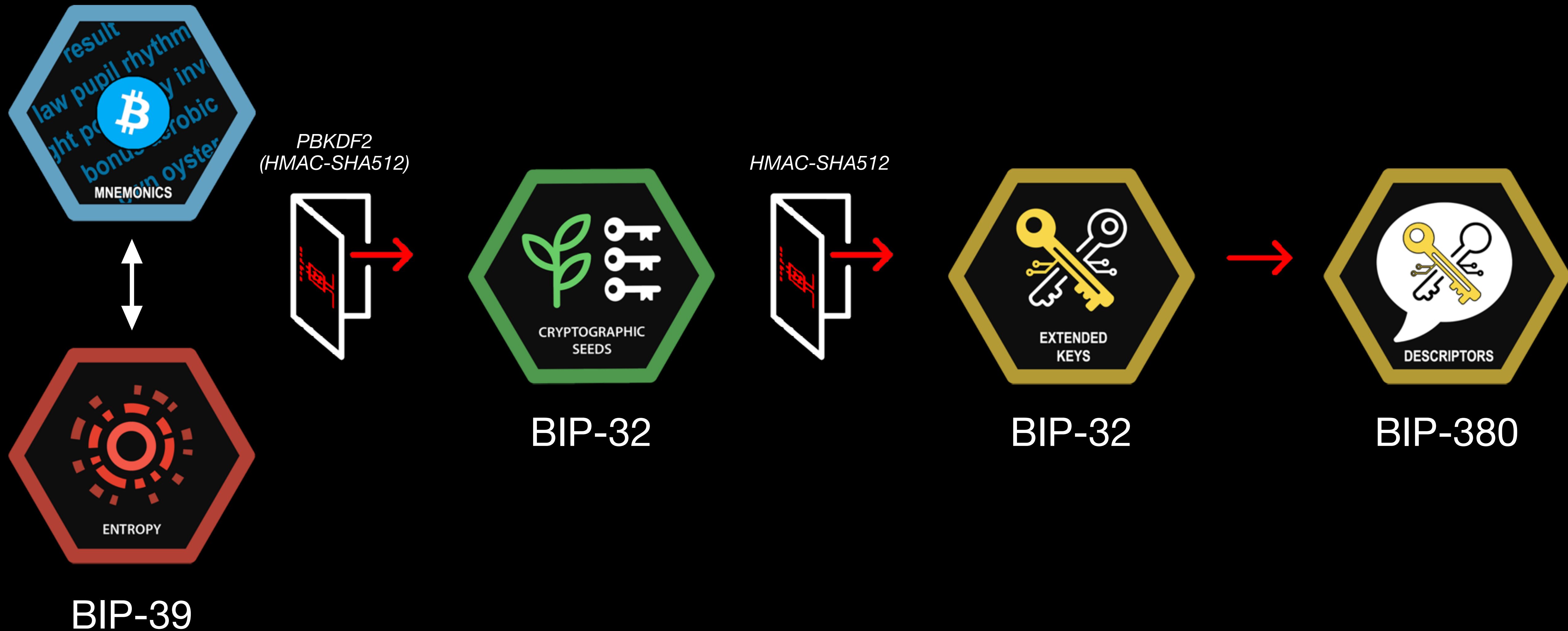
```
ShannonA — standup@lbt: ~/.bitcoin/signet/wallets — ssh root@192.155.84.113 — 80x23
~ — standup@lbt: ~/.bitcoin/signet/wallets — ssh root@192.155.84.113

'??~?R?8?zwalletdescriptorkey??Y????E2?????' ?y!^L?

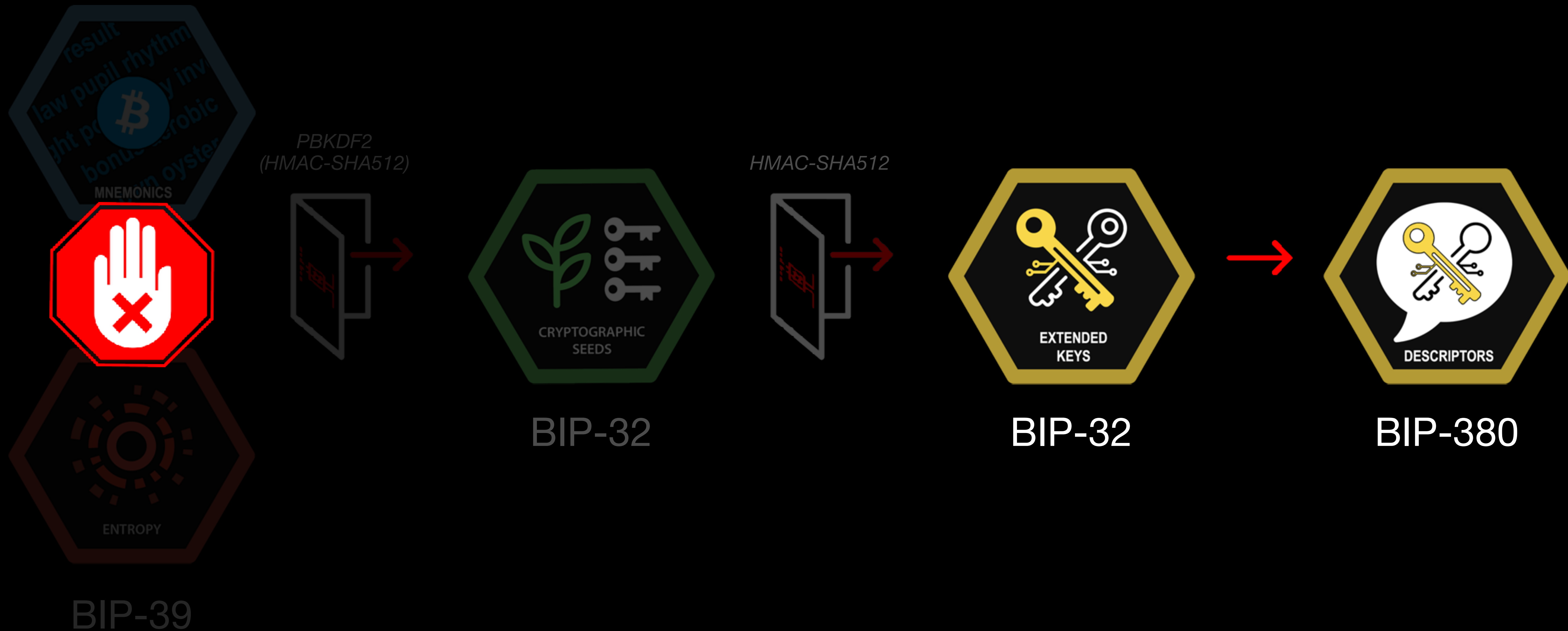
?Ws??JAvkHNv*?@{?
    ?"walletdescriptorlhcach?7#?1?9?:?
?Ws??JAvkHNv*?@{?J?F}?uvKT:?y|?`{dB????x?pNk? CBS?}??Y?asawXv??b?????
    \?9?

??"walletdescriptorcach?7#?1?9?:?
?Ws??JAvkHNv*?@{?JZ&H'? "?<?u0?Cb??D?k????2??6?"?
    ??"?*??[?Cyh_??^? h?z9A+[?]n?Jwa
1
letdescriptor?7#?1?9?:?
?Ws??JAvkHNv*?@{?tr(tpubD6NzVbkrYhZ4Yfa33nsknBn8PmByRir5ZdqRBVyknNZxitJXkgQ2Gg3k
4t3nvumSMNtgh55UubHr1xYmapwqBzywRopD1ayWu4okEkb9Cs4e/86h/1h/0h/0/*)#h7xm5znh?5bj
??R?8?zwalletdescriptorkey?7#?1?9?:?
?Ws??JAvkHNv*?@{?!j{'??D?DEi}?;?S?r??@??f?W?58#?0?? M?Y?龔~$??9?q0'??SVz?g
0????0??0,*?H?=? ?????????????????????????????????????/0!y?f~?U?b?
    ???-(?Y?[??!??????
???
??????????H?;??^??6AA?$"j{'??D?DEi}?;?S?r??@??f?W?58#"?W9??
    j`??rG ,~~??\?p?@??
???262Lactiveexternalspk?fîx?ldk. ?8??N
    ?[c????%????^L??"walletdescriptorlhcach
standup@lbt: ~/.bitcoin/signet/wallets$
```

How the Bitcoin Ecosystem Works



How Bitcoin Core Works



Solutions

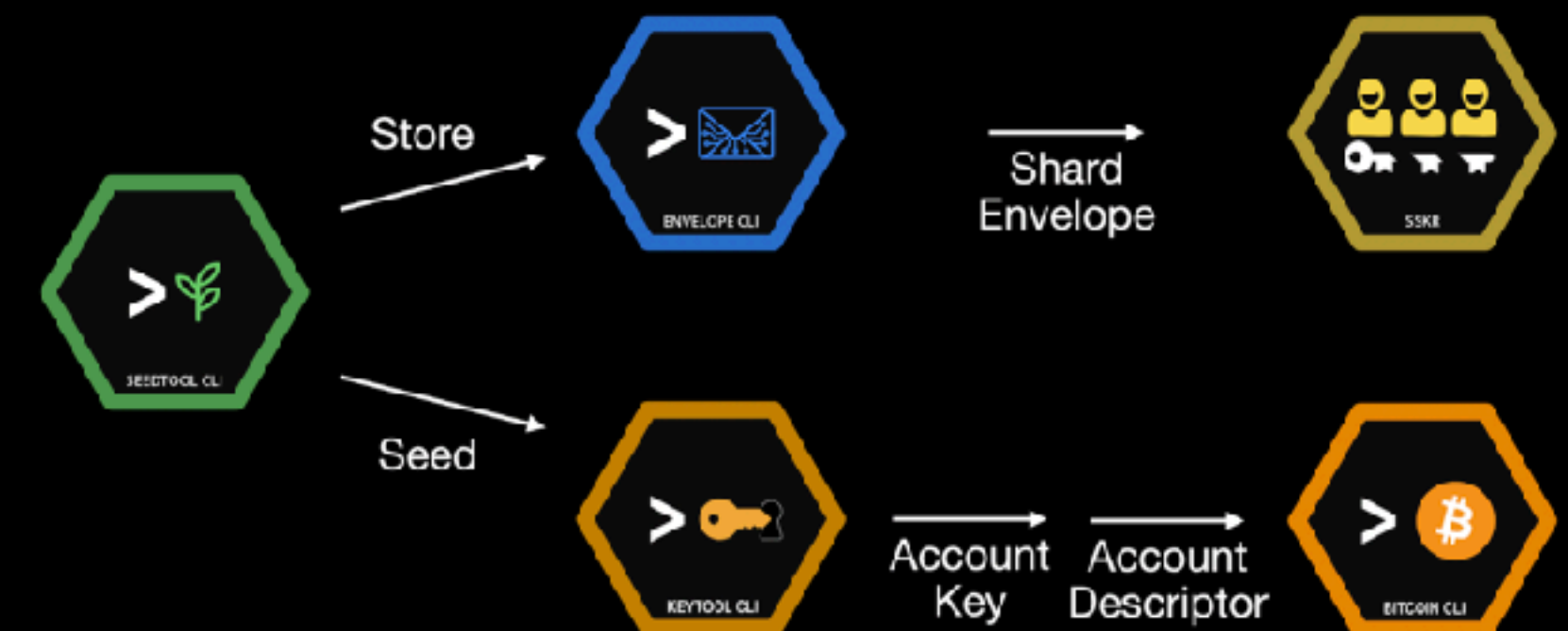
Import a Sharded Secret; or Export a Secret to Shard

- ▶ So how do we make Bitcoin Core interoperable and resilient?
- ▶ If we can't shard in Bitcoin Core
 - ▶ We shard outside of Bitcoin Core
- ▶ Either We:
 - ▶ Export a secret from Core; or
 - ▶ Create a secret outside of Core and import it

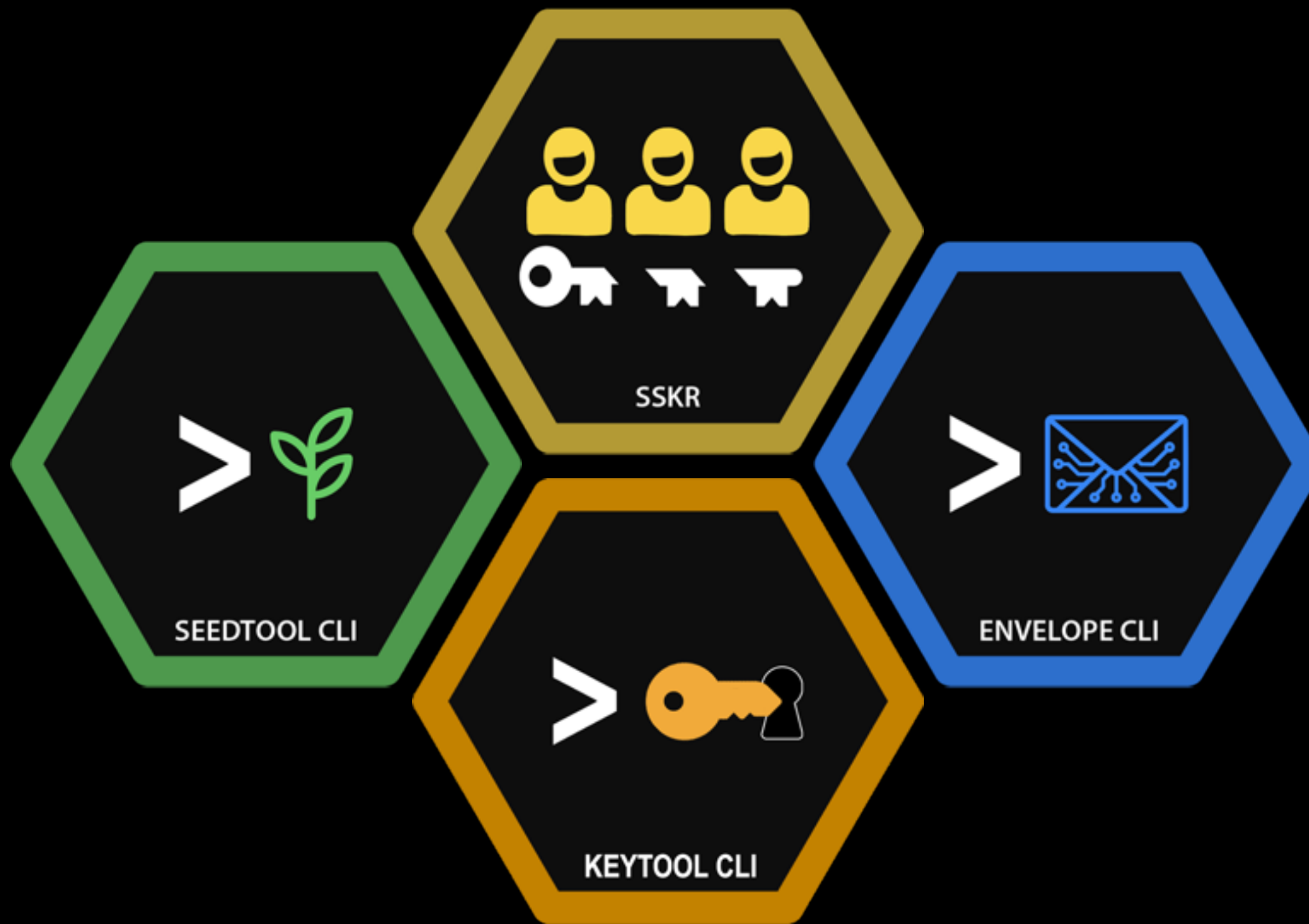
Game Plan: Export



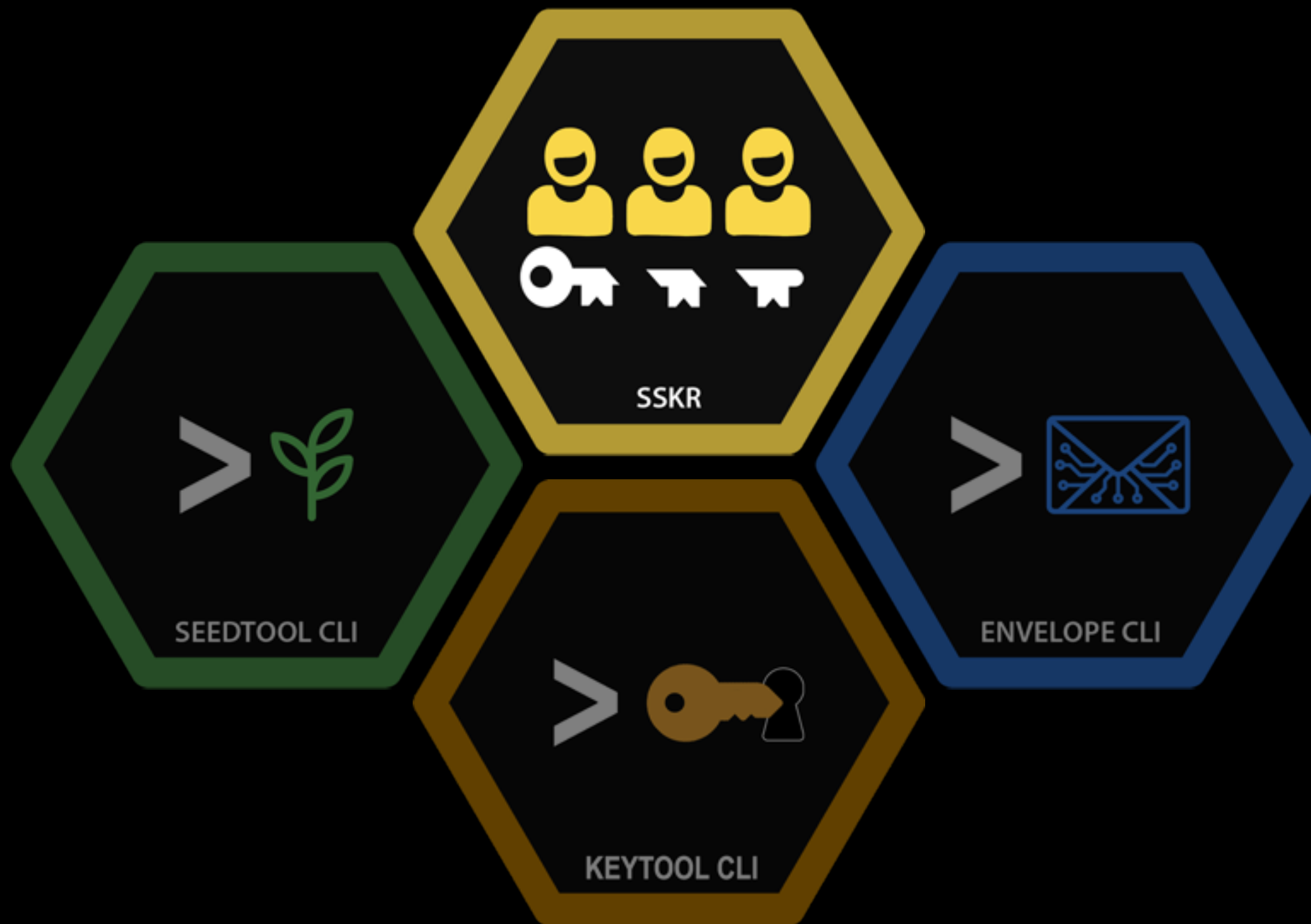
Game Plan: Import



Our Tools



Our Shamir



A network diagram consisting of numerous black cubes arranged in a grid-like pattern, connected by thin black lines. One cube, located in the lower-middle section of the grid, is highlighted in a solid green color, making it stand out from the others.

Solution #1

Exporting a Secret from Bitcoin Core

Game Plan: Export





Step 1: Understand the Descriptor

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ bitcoin-cli listdescriptors true | jq -r '.descriptors[].desc'
pkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRB
yBaY1qNGkEqumEHY/44h/0h/0h/0/*)#s0gh07wt
pkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRB
yBaY1qNGkEqumEHY/44h/0h/0h/1/*)#pmdkjt7n
sh(wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRt
gxRByBaY1qNGkEqumEHY/49h/0h/0h/0/*))#0xcgdu1g
sh(wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRt
gxRByBaY1qNGkEqumEHY/49h/0h/0h/1/*))#f9sdk35u
tr(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRB
yBaY1qNGkEqumEHY/86h/0h/0h/0/*)#ftzrqrf
tr(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRB
yBaY1qNGkEqumEHY/86h/0h/0h/1/*)#cl8zakep
wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxR
ByBaY1qNGkEqumEHY/84h/0h/0h/0/*)#tdj5nsz1
wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxR
ByBaY1qNGkEqumEHY/84h/0h/0h/1/*)#6eh4w9j8
standup@lbt:~$
```




Step 2: Extract Descriptors

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ DESC=$(bitcoin-cli listdescriptors true | jq -r '.descriptors[].desc')
standup@lbt:~$ DESC_ARRAY=( $DESCS )
standup@lbt:~$ for ((i = 0; i < 8; i++)); do
  echo "$i: ${DESC_ARRAY[$i]}";
done
0: pkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtg
xRByBaY1qNGkEqumEHY/44h/0h/0h/0/*)#s0gh07wt
1: pkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtg
xRByBaY1qNGkEqumEHY/44h/0h/0h/1/*)#pmdkjt7n
2: sh(wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85i
yRtgxRByBaY1qNGkEqumEHY/49h/0h/0h/0/*))#0xcgdulg
3: sh(wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85i
yRtgxRByBaY1qNGkEqumEHY/49h/0h/0h/1/*))#f9sdk35u
4: tr(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgx
RByBaY1qNGkEqumEHY/86h/0h/0h/0/*)#ftzqrfe
5: tr(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgx
RByBaY1qNGkEqumEHY/86h/0h/0h/1/*)#cl8zakep
6: wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRt
gxRByBaY1qNGkEqumEHY/84h/0h/0h/0/*)#tdj5nszl
7: wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRt
gxRByBaY1qNGkEqumEHY/84h/0h/0h/1/*)#6eh4w9j8
standup@lbt:~$
```




Step 3: Extract Master Key

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ MP_KEY=$(echo ${DESC_ARRAY[0]} | awk -F"[" '{print $2}' | awk -F"/" '{print $1}'
)
standup@lbt:~$ echo $MP_KEY
xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY
1qNGkEqumEHY
standup@lbt:~$
```




Step 4a: Store Master Key

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ KEY_ENVELOPE=$(envelope subject type string "$MP_KEY")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj known 'isA' known 'MasterKey' "$KEY_ENVELOPE")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj string "createdBy" string "`bitcoin-cli --version | head -1`" "$KEY_ENVELOPE")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj string "usedBy" string "`bitcoin-cli --version | head -1`" "$KEY_ENVELOPE")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj known 'DerivationPath' string "m/44h/0h/0h" "$KEY_ENVELOPE")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj known 'DerivationPath' string "m/84h/0h/0h" "$KEY_ENVELOPE")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj known 'DerivationPath' string "m/49h/0h/0h" "$KEY_ENVELOPE")
standup@lbt:~$ KEY_ENVELOPE=$(envelope assertion add pred-obj known 'DerivationPath' string "m/86h/0h/0h" "$KEY_ENVELOPE")
standup@lbt:~$ █
```




Step 4a: Store Master Key

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ envelope format $KEY_ENVELOPE
"xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBa
Y1qNGkEqumEHY" [
  'isA': 'MasterKey'
  "createdBy": "Bitcoin Core RPC client version v30.2.0"
  "usedBy": "Bitcoin Core RPC client version v30.2.0"
  'DerivationPath': "m/44h/0h/0h"
  'DerivationPath': "m/49h/0h/0h"
  'DerivationPath': "m/84h/0h/0h"
  'DerivationPath': "m/86h/0h/0h"
]
standup@lbt:~$
```


Why We Store “Created By”

- ▶ Seed/Key Creation Can Be Buggy
- ▶ In an Interoperable World, Where You Move Your Bitcoin Around, You Need To Be Able to Look Up Where Seeds Came From
- ▶ ColdCard Proved This Last Week
- ▶ But It's Not the First Case: [blockchain.info](#) (2014), `secureRandom` in JavaScript (2018)

Bitcoin owners rocked by \$116 million hack: What we know about the Coldcard exploit

By Camila
Crypto Fellow
August 3, 2018

**Coldcard wallet attack
\$89M in Bitcoin from**

**White Hat Hacker Returns Missing
Bitcoins to Blockchain.info**

Latest News Published Dec 11, 2014

On December 8, Blockchain.info—a Bitcoin web wallet provider and block explorer—announced that they'd misfired a software update. The faulty update resulted in insecure private key generation for some users, and consequently, some users' bitcoins were at risk.

Share

**Vulnerability in JavaScript Function May Mean
term Bitcoin Hodlers are at Risk**

By Priyeshu Garg Apr 20, 2018 at 05:45 AM HST This article is more than 4 years old

Prefer us on Google



Step 4b: Store Descriptors

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ DESC_ENVELOPE_1=$(envelope subject type string "descriptors-for-bitcoin")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'isA' string "collectionOfDescriptors" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj string "createdBy" string "`bitcoin-cli --version | head -1`" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj string "usedBy" string "`bitcoin-cli --version | head -1`" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[0]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[1]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[2]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[3]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[4]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[5]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[6]}" "$DESC_ENVELOPE_1")
DESC_ENVELOPE_1=$(envelope assertion add pred-obj known 'OutputDescriptor' string "${DESC_ARRAY[7]}")
standup@lbt:~$
```




Step 4b: Store Descriptors

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ envelope format $DESC_ENVELOPE_1
"descriptors-for-bitcoin" [
  'isA': "collectionOfDescriptors"
  "createdBy": "Bitcoin Core RPC client version v30.2.0"
  "usedBy": "Bitcoin Core RPC client version v30.2.0"
  'OutputDescriptor': "pkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/44h/0h/0h/0/*)#s0gh07wt"
  'OutputDescriptor': "pkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/44h/0h/0h/1/*)#pmdkjt7n"
  'OutputDescriptor': "sh(wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/49h/0h/0h/0/*))#0xcgdu1g"
  'OutputDescriptor': "sh(wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/49h/0h/0h/1/*))#f9sdk35u"
  'OutputDescriptor': "tr(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/86h/0h/0h/0/*)#ftzrqrf"
  'OutputDescriptor': "tr(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/86h/0h/0h/1/*)#c18zakep"
  'OutputDescriptor': "wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/84h/0h/0h/0/*)#tdj5nsz1"
  'OutputDescriptor': "wpkh(xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWfd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBaY1qNGkEqumEHY/84h/0h/0h/1/*)#6eh4w9j8"
]
standup@lbt:~$ █
```




Step 5: Shard Envelope

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ KEY_SHARES=$(envelope sskr split --group "2-of-3" $KEY_ENVELOPE)
standup@lbt:~$ KEY_ARRAY=($KEY_SHARES)
standup@lbt:~$
```




Step 6a: Check Your Work

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ echo ${KEY_ARRAY[0]}
ur:envelope/lftansfwlrhkadehfsbsfpjnkikgnncweeongygorstpgtahoxostyaevwaxbsatamsrsnntgshsjotdolesuop
ehdlnehchptjycwtarygmoybyytlvnybyhytapmimwelrolbdolkbmswfkpcxttaawswtldaxiafnrtimjycaryoentwzvdbdgu
fzcxhntiasosmwdpprhycdynfxwnckpsinwfcfweaoaxzorflmtbgwmpyteqzwzynmekproskqzjzgyylfzplttmntshpvljtc
tyarhylsknelfinltzokgtdjlptclkogstakghngdjyoxgoadlafzfeknjpjykkkeykfhdlsfrscgazoayeysgwkrpjzyltlko
uoinjsnbrpwefsaerftpkorornhlehyzmayaxsbiekoihisrornimlasptaesadlbdllufmoxlybspdsecxldndttuohlmdar
ncwecvycajyvabekgcyfylrsg hkpyaxfetbgtcwgrjkvdwlnlamvdgmqdcstkchvdwlkplydlrleegslglkgringlwnpdrykigr
tlkshtrefeceswythlbwsgotntkolkkkwdcwchbzkgsbgsrochwzsarfjlltkigetipsfggdekgvwceguhddygdrkweatjeuoh
pptgmhddatansfphdcxtifmionhfgwedyueeegmroyaplpdktrpcxmwtqdrdjygodepswftlrstdiymoooyamtpso tantkphd
daqzpk aeadaewkjtldglpdlkosoycymoctpfvysemtaopmwzzelymyjnbsvacmvluo onmnesfmysadydtge

[standup@lbt:~$ echo ${KEY_ARRAY[1]}
ur:envelope/lftansfwlrhkadehfsbsfpjnkikgnncweeongygorstpgtahoxostyaevwaxbsatamsrsnntgshsjotdolesuop
ehdlnehchptjycwtarygmoybyytlvnybyhytapmimwelrolbdolkbmswfkpcxttaawswtldaxiafnrtimjycaryoentwzvdbdgu
fzcxhntiasosmwdpprhycdynfxwnckpsinwfcfweaoaxzorflmtbgwmpyteqzwzynmekproskqzjzgyylfzplttmntshpvljtc
tyarhylsknelfinltzokgtdjlptclkogstakghngdjyoxgoadlafzfeknjpjykkkeykfhdlsfrscgazoayeysgwkrpjzyltlko
uoinjsnbrpwefsaerftpkorornhlehyzmayaxsbiekoihisrornimlasptaesadlbdllufmoxlybspdsecxldndttuohlmdar
ncwecvycajyvabekgcyfylrsg hkpyaxfetbgtcwgrjkvdwlnlamvdgmqdcstkchvdwlkplydlrleegslglkgringlwnpdrykigr
tlkshtrefeceswythlbwsgotntkolkkkwdcwchbzkgsbgsrochwzsarfjlltkigetipsfggdekgvwceguhddygdrkweatjeuoh
pptgmhddatansfphdcxtifmionhfgwedyueeegmroyaplpdktrpcxmwtqdrdjygodepswftlrstdiymoooyamtpso tantkphd
daqzpk aeadaadtkgdktdlabgcklkguiacwfeuo oegwcn ytnsttflrsbasaktwzsespwftplrsiol tcyrlki

[standup@lbt:~$ echo ${KEY_ARRAY[2]}
ur:envelope/lftansfwlrhkadehfsbsfpjnkikgnncweeongygorstpgtahoxostyaevwaxbsatamsrsnntgshsjotdolesuop
ehdlnehchptjycwtarygmoybyytlvnybyhytapmimwelrolbdolkbmswfkpcxttaawswtldaxiafnrtimjycaryoentwzvdbdgu
fzcxhntiasosmwdpprhycdynfxwnckpsinwfcfweaoaxzorflmtbgwmpyteqzwzynmekproskqzjzgyylfzplttmntshpvljtc
```




Step 6b: Check Your Work

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

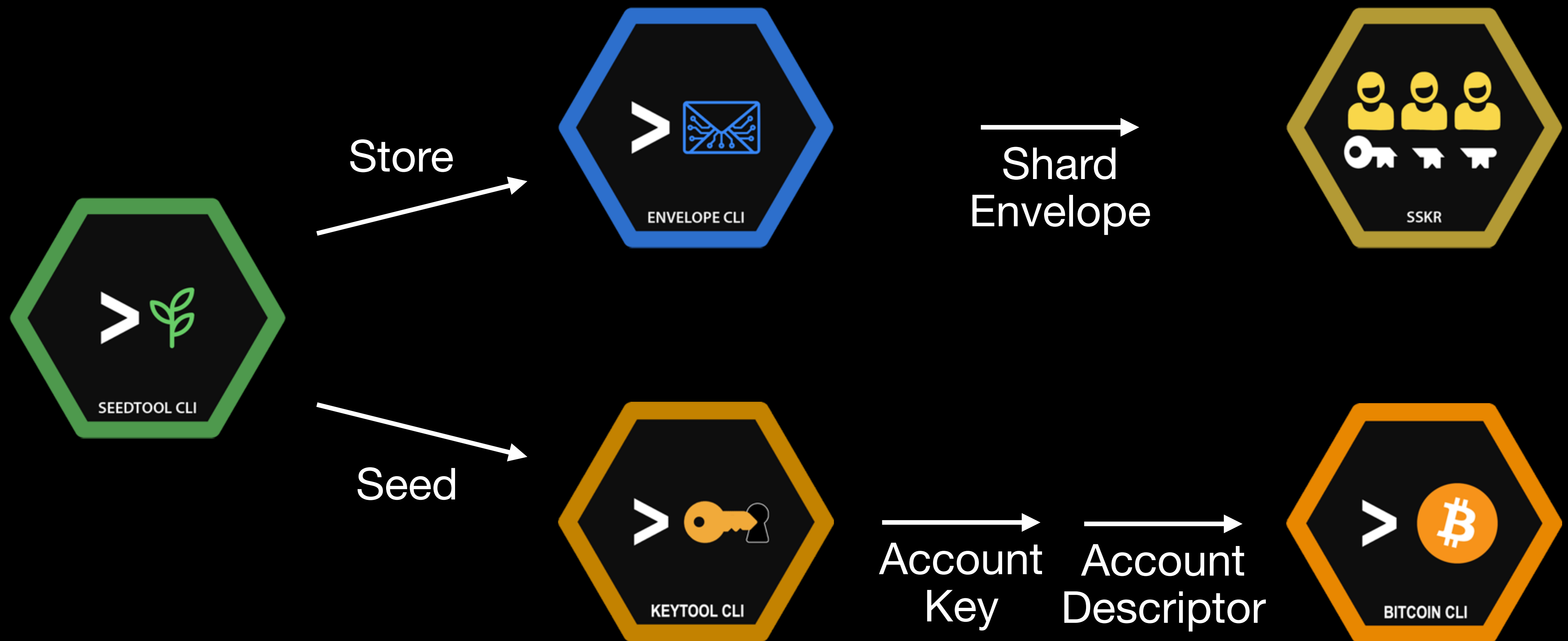
standup@lbt:~$ RESTORED_KEY=$(envelope sskr join "${KEY_ARRAY[0]}" "${KEY_ARRAY[1]}")
standup@lbt:~$ envelope format $RESTORED_KEY
"xprv9s21ZrQH143K2ozsiv9udXZcCfan65WtoxznEM3kJonhD8p6vWFd8Lp3LoQ8hvejFJAqHsyQGEfS3CBdL85iyRtgxRByBa
Y1qNGkEqumEHY" [
  'isA': 'MasterKey'
  "createdBy": "Bitcoin Core RPC client version v30.2.0"
  "usedBy": "Bitcoin Core RPC client version v30.2.0"
  'DerivationPath': "m/44h/0h/0h"
  'DerivationPath': "m/49h/0h/0h"
  'DerivationPath': "m/84h/0h/0h"
  'DerivationPath': "m/86h/0h/0h"
]
standup@lbt:~$
```




Solution #2

Importing a Secret into Bitcoin Core

Game Plan: Import





Step 1: Create a Seed

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ seedtool
7eee22944bae4290ffffbfd601bb9601
standup@lbt:~$ SEED=$(seedtool)
standup@lbt:~$ echo $SEED
b7cfbeb7dc646f4fd123af196086784f
standup@lbt:~$
```




Step 2: Create a Fingerprint

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ keytool --seed $SEED master-key-fingerprint
346bba8d
[standup@lbt:~$ FINGERPRINT=$(keytool --seed $SEED master-key-fingerprint)
[standup@lbt:~$ echo $FINGERPRINT
346bba8d
standup@lbt:~$
```




Step 3: Create Account Key

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ keytool --seed $SEED --account-derivation-path "m/84h/0h/0h" account-key-base58
xprv9yuLiuGTeoMXReVJ9S5wd3SAox8ducu6ejRT3g4ZWm8gyqHUwEv9EcftmPRbuYHAF9bp5VzkKVG7ydSn4zFZ6v1A9BCqGB
Que2uQUsk5LR
[standup@lbt:~$ AKEY=$(keytool --seed $SEED --account-derivation-path "m/84h/0h/0h" account-key-base58)
[standup@lbt:~$ echo $AKEY
xprv9yuLiuGTeoMXReVJ9S5wd3SAox8ducu6ejRT3g4ZWm8gyqHUwEv9EcftmPRbuYHAF9bp5VzkKVG7ydSn4zFZ6v1A9BCqGB
Que2uQUsk5LR
standup@lbt:~$
```




Step 4: Create Descriptor

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ DESC="wpkh([$FINGERPRINT/84h/0h/0h]$AKEY/0/*)"
standup@lbt:~$ DESC_CS=$(bitcoin-cli getdescriptorinfo $DESC | jq -r '.checksum')
standup@lbt:~$ DESC_WITH_CS=$DESC#$DESC_CS
[standup@lbt:~$ echo $DESC_WITH_CS
wpkh([346bba8d/84h/0h/0h]xprv9yuLiuGTeoMXReVJ9S5wd3SAox8ducu6ejRT3g4ZWm8gyqHUwEv9EcftmPRbuYHAF9bp5
VzkKVG7ydSn4zFZ6v1A9BCqGBQue2uQUsk5LR/0/*)#6kxddmw7
standup@lbt:~$
```




Step 5: Import Descriptor

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ bitcoin-cli -named createwallet wallet_name="seedtool" blank=true
{
  "name": "seedtool"
}
[standup@lbt:~$ bitcoin-cli -rpcwallet=seedtool importdescriptors '[{"desc": "'$DESC_WITH_CS'",
"timestamp":1780329126, "active": true, "range": [0,100] }]'
[
  {
    "success": true
  }
]
standup@lbt:~$
```




Step 6: Check Your Work

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ bitcoin-cli -rpcwallet=seedtool listdescriptors
{
  "wallet_name": "seedtool",
  "descriptors": [
    {
      "desc": "wpkh([346bba8d/84h/0h/0h]xpub6Cth8QoMVAupe8ZmFTcwzBNuMyy8K5cx1xM3r4UB56ffrdcdUnEPnQz
NZ3tRpWLMaruw3hgVffmgje6oFQv4ivzLaeBY2grRPZxJbWcVxMA/0/*)#ju2f3q35",
      "timestamp": 1780329126,
      "active": true,
      "internal": false,
      "range": [
        0,
        999
      ],
      "next": 0,
      "next_index": 0
    }
  ]
}
standup@lbt:~$
```




Step 7a: Back Up with Seedtool

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ seedtool -i hex $SEED -o bip39
result law pupil rhythm eight pony duty invest bonus aerobic own oyster
standup@lbt:~$ seedtool -i hex $SEED -o sskr --groups 2-of-3 --sskr-format ur
ur:sskr/gosgtyaeadaegylnahsomsfxstamonssdwotasynuofxoybdnshd
ur:sskr/gosgtyaeadaadktfrechfbgrdayrkvozelemycfselstevarydyce
ur:sskr/gosgtyaeadaocavdihwplnpgfwiodnpfkgzodtmkidkssbndhgwl
standup@lbt:~$ █
```




Step 7b: Test Your Backup

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151
[standup@lbt:~$ echo "ur:sskr/gosgtyaeadaegylnahsomsfxstamonssdwotasynuofxoybdnshd ur:sskr/gosgtyae]
adaocavdihwplnpkfwiodnpfkgzodtmkidkssbndhgw1" | seedtool -i sskr
b7cfbeb7dc646f4fd123af196086784f
standup@lbt:~$ █
```




Step 8a: Back Up with Envelope

```
ShannonA — standup@lbtc: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbtc: ~ — ssh root@96.126.98.151

standup@lbtc:~$ SEED_ENVELOPE=$(envelope subject type string "$SEED")
[SEED_ENVELOPE=$(envelope assertion add pred-obj known 'isA' known 'Seed' "$SEED_ENVELOPE")
[standup@lbtc:~$ envelope format $SEED_ENVELOPE
"b7cfbeb7dc646f4fd123af196086784f" [
    'isA': 'Seed'
]
standup@lbtc:~$
```




Step 8b: Add Metadata to Seed

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

standup@lbt:~$ SEED_ENVELOPE=$(envelope assertion add pred-obj string "createdBy" string "`seedtool -V`" "$SEED_ENVELOPE")
SEED_ENVELOPE=$(envelope assertion add pred-obj string "usedBy" string "`bitcoin-cli --version | head -1`" "$SEED_ENVELOPE")
SEED_ENVELOPE=$(envelope assertion add pred-obj known 'DerivationPath' string "m/84h/0h/0h" "$SEED_ENVELOPE")
standup@lbt:~$ envelope format $SEED_ENVELOPE
"b7cfbeb7dc646f4fd123af196086784f" [
  'isA': 'Seed'
  "createdBy": "seedtool-cli 0.4.0"
  "usedBy": "Bitcoin Core RPC client version v30.2.0"
  'DerivationPath': "m/84h/0h/0h"
]
standup@lbt:~$
```




Step 8c: Shard Seed

```
ShannonA — standup@lbt: ~ — ssh root@96.126.98.151 — 99x24
~ — standup@lbt: ~ — ssh root@96.126.98.151

[standup@lbt:~$ SEED_SHARES=$(envelope sskr split --group "2-of-3" $KEY_ENVELOPE)
[standup@lbt:~$ SEED_ARRAY=($SEED_SHARES)
[standup@lbt:~$ echo ${SEED_ARRAY[0]}
ur:envelope/lftansfwlrhkadehvslgtamwtdjljnvdenmkpldlaeplimkigwkibdatwyluhgbdmkkbtpfrlomtmtsglnjybnf
wndsfatrpfsskkevautmevakeoxsscklrjzjydmrpwndwkesrbbsceltgwsewtadsfsoyardpsgotkwyfrhknbjolsbknekiqz
uytlischaacsjzdtlnlfnolrfdkcsslrmeryamgtesnteojszmfpdwckiewtbzvd lunevyspglmenbhflpkbnlisvlfzjkfdm
shshkjoayyaspbzgrwdrezcrhhtbsfpinrntyflrllgpeinstvleewtmnkscezt hlgdajlmeps ltonlgaesetntbbeurkndnen
bzlpo xwlsabbbbhhqdbkykgtkkbwsnwptdlpfelejlb tasisondnmulumkdnutnybyfglojyssdn nscasbp liyhnoygynefwvss
ataglknlupfv srftbceidvyamlogyutplkiosuyjkdaptwladj nfxihktylkbrdsej egunlbbdauebsbzc fweldj slkgansonyl
vwctykatrfcfmejn lpttvsbegr dptyjnhfosvlbgpfgsgsnlv yuthdktmycpnbhglfclsbgd k k zssbhfv ydrhlvedeqzltamyaz
mmkrohddatansfphdcxtifmiomhfgwedyueeeegmroyaplpdktrpcxm wutqdrdjygodepswftlrst diyndmooyamtpso tantkphd
dafepkaeadaecfgawkf wlgty motyeor nnelonloxfzwnwlvonbvlbytnvlp ltpbgndlutowsjpimpfhtltfx
standup@lbt:~$
```


A 3D network diagram consisting of numerous black cubes connected by thin black lines, forming a complex, interconnected mesh. The cubes are arranged in a way that suggests a distributed system or a network topology. One cube, located in the lower-middle section of the image, is highlighted in a vibrant green color, making it stand out from the rest of the network.

Addendum Storing Your URs as QRs

Seeds, Keys, and Shards

Can All Be Stored as QR Codes



68028db28e55fe56035cc332267cd1d8



ur:sskr/gogomnaeadaennenaotleedesgguwtspaooszohtdmssdtvawegm
ur:sskr/gogomnaeadadiadyeejnscsmusffhhsvyutmojlrsdlmydebsbdlr
ur:sskr/gogomnaeadaolbftjtrnjzfeswlusonyossnspludwgmtadtptsg

Our Tools (The QRs)





"Sharding Secrets to Protect Your Bitcoin"

[https://learningbitcoin.blockchaincommons.com/
10_0_Working_with_Secrets/](https://learningbitcoin.blockchaincommons.com/10_0_Working_with_Secrets/)



@BlockchainComns



company/blockchain-commons

"Advocating for the creation of open, interoperable, secure & compassionate digital infrastructure to enable people to control their own digital destiny and to maintain their human dignity online"

