BIP-85

General purpose HD keychains

Aneesh Karve

@akarve

dowsing.seaport0d@icloud.com

https://github.com/akarve/bipsea

Derive millions of secrets from one seed

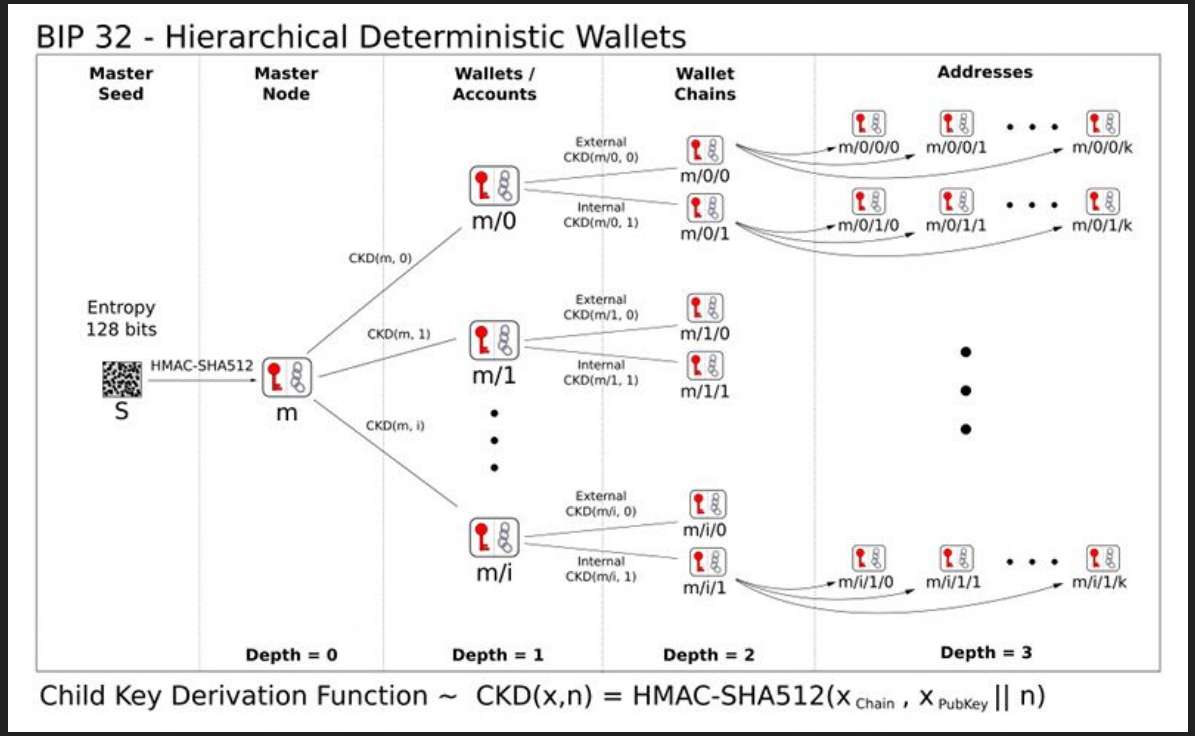
"One Seed to rule them all,
One Key to find them,
One Path to bring them all,
And in cryptography bind them."

Derivation paths (BIP-44)

```
m / purpose' / coin_type' / account' / change / address_index
```

- ' indicates hardened derivation
- Each segment is a 32-bit unsigned int; set high bit for hardened derivation

Hierarchical Deterministic Wallets (HDW, BIP-32)



HDW properties

- Irreversible (up to hardening)
- Reproducible (each derived key is a pure function of the master seed and path)

* Hardened derivation and irreversibility

- Compromised xpub = compromised privacy
  - Useful & dangerous: can derive all children (to watch addresses)
- Compromised xprv = comprised funds
- Non-hardened prv(child) + xpub(parent) = xprv(parent)

* HDW derivation steps (BIP-32, BIP-39)

1. Mnemonic =: seed words
2. PBKDF2(norm(seed words)) =: master seed
3. HMAC(master seed) =: root extended private key (xprv)
4. At each level of the derivation: HMAC_and_ECC(parent_xprv) =: child xprv

* Understanding BIP-32 functions

- CKD_priv(private_key) => hard or soft private key
- N(private_key) => public_key
- CKD_pub(public_key) => soft public key

```
pip install bipsea
```

```
$ bipsea mnemonic -t jpn -n 15
```

おかわり おっと ゆにゅう いこつ ろうそく げつれい
おかわり きらい ちたん にくまん でんわ ずぶぬれ く
ださる いらすと のみもの

```
$ bipsea mnemonic -n 12 --pretty

1) beach
2) tail
3) trial
4) design
5) lyrics
6) …
```

```
$ bipsea mnemonic -t spa -n 12 | bipsea
validate -f spa

1) beach
2) tail
3) trial
4) design
5) lyrics
6) …
```

```
$ bipsea validate -f free -m "$(cat input.txt)"
```

- steganographic seeds
- Free seed mnemonics for steganography and attack-resistance
- https://github.com/akarve/bip-keychain/blob/main/pre-bips/steganographic-seeds.md
- PKBDF2 is input agnostic (but validation is on you); could make a brain wallet this way, but more entropy is better (although secp256k1 is 256-bit)

```
$ bipsea mnemonic | bipsea validate |
bipsea xprv

xprv9s21ZrQH143K41bKPQ9XHbPoqfdCDmZLBorYHay
5E273HTu5yAFm27sSWRoCpisgQNH9vfrL9yVvVg5rBE
bMCk2UwQ8K7qCFnZAY7aXhuqV
```

```
$ bipsea mnemonic | bipsea validate |
bipsea xprv | bipsea derive -a mnemonic -n
12

rotate link six joy boss sock unveil
achieve charge sweet hidden regular
```

```
$ bipsea validate -f free -m "blah blah blah" | bipsea xprv | bipsea derive -a drng -n 1000
Warning: Relative entropy of mnemonic seems low (0.37). Consider a more complex --mnemonic.

14089826f0b88542ea926369494c720fa3c0e5ee4202865575d5fd3b933c65ca044a010f85a9e75122cb9a61045302111
08a1f1533e8eaeeaedb32dc7e8e78148982eb8cd627318e1557dd3eb0e9aeaedc7045dc97bc0128756e2c8c08479ccdff
9420e8b9052fb5dbcdc0a41e47a345b8330435686de507e5e999b5831ace88c81b2cbc33111c0c185d450a73f18383e2c
b4ad93079570ac65854bfdb50708fb7b77c8adaf535a35d602f83e911e54ca4029a44d79886b63230e28d3f4d7daf36e6
fa9d688f0ddc9e4f17a51e0155ef20906f3e4da3eda87ed6c420a981a3227b31becdaaa969834831a219d7c56dada3e8f
752971d6b8757dbea8bb5d25f69c2b79695425361e8edd03b1adcd724c618b237b46c4a593c1bef735f62f671b5df4aa3
49c0ee2cf1f0cade8626a2dd316e62c3065aa1d3d9cb6b2ec3b7bbff38f1bc737c2ed88f338e31ee441087784f6626074
f6e9cdd082cd7d69f383ddd5059c69f5d6a6035ffe247f5f622c22c33f11da28e96eb7fd1fbd92a45db8f72803c3311f7
1aa070effddfb0c5efe9e40f1d0fd1ecf38c2cad2fe91e078ce431fb8f85f839347164330b2e53c41f7577aed46050610
926c37b631a5cc7ce73edc48af7579a868941de6f6f9c819722fb26127ede2f8ea151e8a45bf566fe8fe9831f53a28edb
cfc84a3c0548a0f3658d075c6f8b6c6f445c57ad629e7ccdcb73e0e664c2f7e903e6ced9fc5c756894fa40bd7e7e42a53
7488b5093325e5334000efcecdce57faf45315fa272cad2ff09ac44e74a78c28b5374ab91b6db66196c49de21a6354ec3
3efdd72fb3aaf7e3838a59322b6658defc86f8dca0f3f72a0dd9e3c4494b00968fbf8dfcf640364d88e38c05c14acf669
ffe2278a95f380a21b49b4cb8dc0ae96535787d784638de868ad26b38e8b74ea919ac45540e2500f30b45ac8efa7f2ca4
f0ef07017bd1df18ba0bc5be0277af29c27573b78e594eacbc6d75d6bcbd54b33ac88b7711fe763f043d85c33e46314f1
948609d3040863fc13e59c9c6e81e6025e82bf8370cc84a451f406c0ed007306579e2d47dee55a39332c1458e297d46aa
4c78f20f3865c205772018e80e36335934154eb31dfd40080080f1bd600773d1d10870fb3efc53155b669812ef89fc40c
92e917b5cc590b313647f6de3efcb1d5355f2ccbe4d2962659aa77e20d1aa16740f8f5e685a8104e3ed063cf461d69e7b
1ef95186c0f98fbd06e54ca8fc32b036daddcbb367e65e102bcac44a263f196499fa9ab7b37584b7eb77a296d56a31ff2
827eabdd18d5e59e1500bfcb69e173a6d71f338dbef996dec297ff37fb1fd9a0af10bda1461fd94a57f723d1d5db5e468
a6d480e5cd5d521e533013138e47c3336f933fafd5ce5a855cc14392dc8b
```

```
$ bipsea validate -m "$MNEMONIC" | bipsea
xprv | bipsea derive -a dice -n 100 -s 6

4,2,5,3,4,4,4,5,0,3
```

```
$ bipsea validate -m $MNEMONIC | bipsea
xprv | bipsea derive -a base85 -i 0

iu?42{I|2Ct{39IpEP5zBn=0
```

- Increment index for fresh secrets

Heart of an extended key

- 512 bits (64 bytes) from HMAC
  - First 32 bytes := Secret Key
  - Second 32 bytes := Chain code
- Key insight
  - This is 512 bits of reproducible cryptographic entropy for any application (not just ECC)

Derivation paths for BIP-85 applications

- BIP-39
  - m/83696968'/39'/{language}'/{words}'/{index}'
- XPRV
  - m/83696968'/32'/{index}'
- Dice (in PR)
  - m/83696968'/89101'/{sides}'/{rolls}'/{index}'
- Base64 password
  - m/83696968'/707764'/{pwd_len}'/{index}'

So what?

- Simplified opsec (e.g. for multi-sig)
  - One key (can be multi-factor, incl. passphrase)
- Alternative or complement to SSKR
- Potential entropy from any source (not just BIP-39 mnemonics)
  - Deck of cards, book cipher, chess game, etc.
- Hardware wallets now have everyday use cases for superior password management

BIP-85 vs Apple Keychain

BIP-85	Apple Keychain
Cold root (hard to attack)	Hot root (easy to attack)
Back up short derivation path	Back up long cipher-text
Path reuse on compromise	Record reuse on compromise
No third-party (up to hw security)	Trusted third-party
Derived keys can be derived just-in-time	Derived keys must be stored on device
Root compromise compromises all children	Root compromise compromises all children
Password rotation via child index	Password rotation via OS

BIP-85 limitations and improvements

- Arbitrary application codes and return types
- Unclear derivation path structure for new applications
- Upcoming improvements (https://github.com/bitcoin/bips/pull/1600)
  - Add BIP-39 language
  - Corrected test vectors
  - New applications, e.g. dice for numeric pins
  - Guidance on path structure
- Doubles down on ECDSA
  - https://github.com/akarve/bipsea?tab=readme-ov-file#ecdsa-for-the-curious-and-paranoid

Future: semantic paths

- https://github.com/akarve/bip-keychain
- Instead of arbitrary integer path segments, hash JCS JSON-LD to an int

```
[
  {
    "@context": "https://schema.org",
    "@type": "WebSite",
    "url": "https://bitcoin.org/en/"
  },
  {
    "@context": "https://schema.org",
    "@type": "CreateAction",
    "name": "Password Derivation",
    "object": {
      "@type": "Thing",
      "name": "Password"
    },
    "result": {
      "@type": "PropertyValueSpecification",
      "valuePattern": "[a-zA-Z0-9]{8,16}",
      "minLength": 8,
      "maxLength": 16,
      "valueRequired": true
    }
  }
]
```

Future: steganographic “free” seed mnemonics

- Easier to hide
  - Deck of cards has ~21 words (225 bits) of entropy & easier to generate
- Easier to get wrong (no checksum, or user-provided checksum, no dictionary, more error prone)
- https://github.com/akarve/bip-keychain/blob/main/pre-bips/steganographic-seeds.md