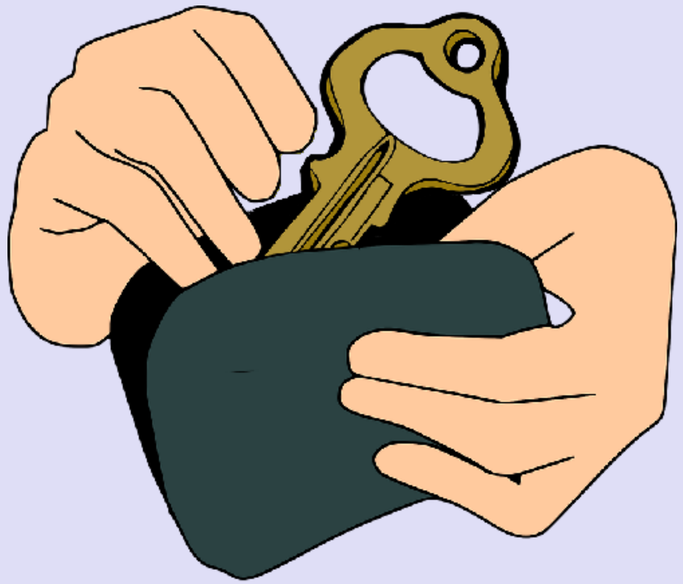
BLOCKCHAIN TUTORIAL 29

HIERARCHICAL DETERMINISTIC WALLET – BIP32 & BIP44

m/44'/60'/0'/0/0



m/44'/0'/0'/0/0

INTRO

- In this video I will explain:
- What a hierarchical deterministic wallet is.
- What BIP-32 and BIP-44 is.

DETERMINISTIC WALLET

- In blockchain tutorial 28 I have explained BIP-39.

- BIP-39 describes how the mnemonic words are created. These mnemonic words together with a password (optionally) is used to generate a 512 bit seed.

- In this video the 512 bit seed is also called the "BIP-39 seed".

- This seed is used as input to generate private and public keys for deterministic wallets.

- There are two types of deterministic wallets:
  - Sequential deterministic wallets.
  - Hierarchical deterministic wallets.

SEQUENTIAL DETERMINISTIC WALLET

- Sequential deterministic wallets generates private keys for example by taking SHA256(seed + n), where n is an index number that starts from 0 and increments as additional keys are needed (simplified explanation).

HIERARCHICAL DETERMINISTIC WALLET

- Nowadays most wallets are **H**ierarchical **D**eterministic (HD) wallets.
- This wallet type is described at:
  https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
- Most HD wallet vendors have implemented BIP-32, BIP-39 and BIP-44.
- These 3 Bitcoin Improvement Proposals are becoming an industry standard.
- If your HD wallet is BIP 32/39/44 compliant than you can "transfer" your private keys to another wallet from another vendor which also implemented these standards. However implementation of these standards can differ. For example a vendor implementing BIP-39 uses his own wordlist, making his wallet not compatible with other vendors.

BIP-39

- BIP-39 describes the implementation of mnemonic words to generate a 512 bit seed. This seed can be used to create a HD wallet.

- More information about BIP-39 can be found at:
  https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

- Or watch my YouTube video:
  "Blockchain tutorial 28: Bitcoin Improvement Proposal 39 (BIP-39) mnemonic words"
  https://youtu.be/hRXcY_tIlrw

BIP-32

- BIP-32 describes how you can build a **general** hierarchical deterministic wallet. These wallets can be shared partially or entirely with different systems, each with or without the ability to spend coins.

- More information about BIP-32 can be found at:
  https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
  https://bitcoin.org/en/developer-guide#hierarchical-deterministic-key-creation

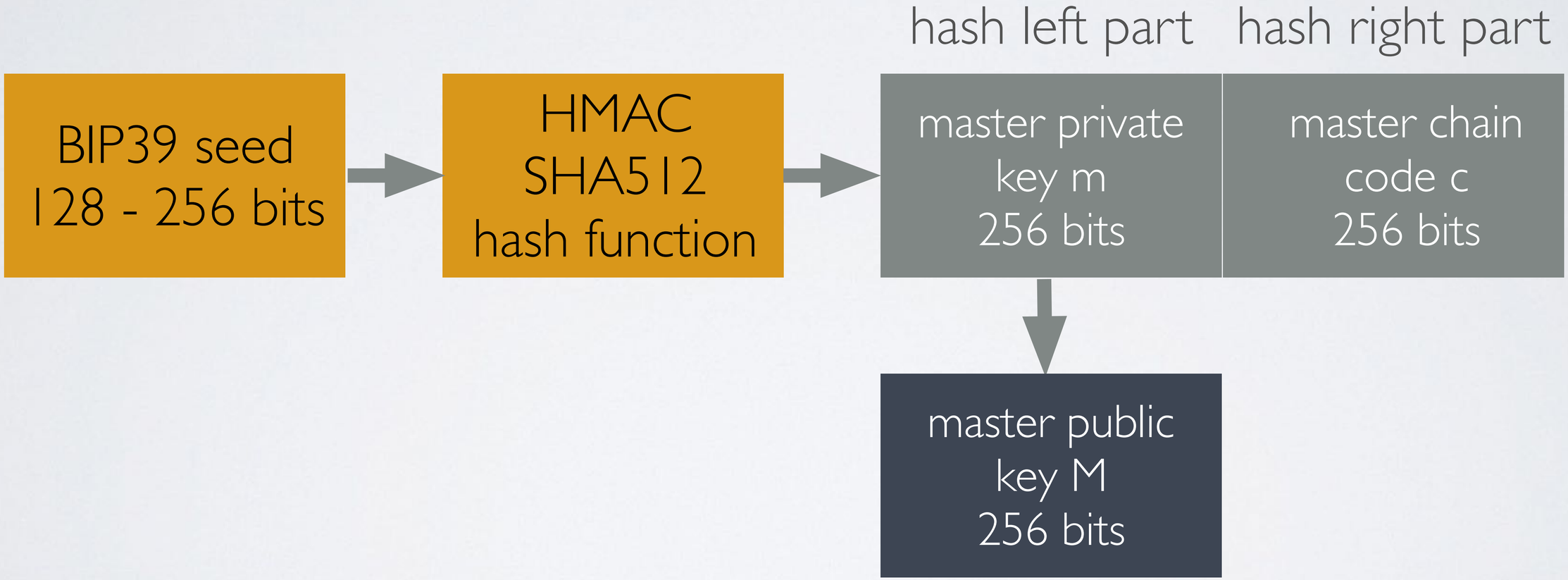
- If you want to see how BIP32 is implemented in the bitcoinjs library, see:
  https://raw.githubusercontent.com/bitcoinjs/bitcoinjs-lib/master/src/hdnode.js

BIP-32

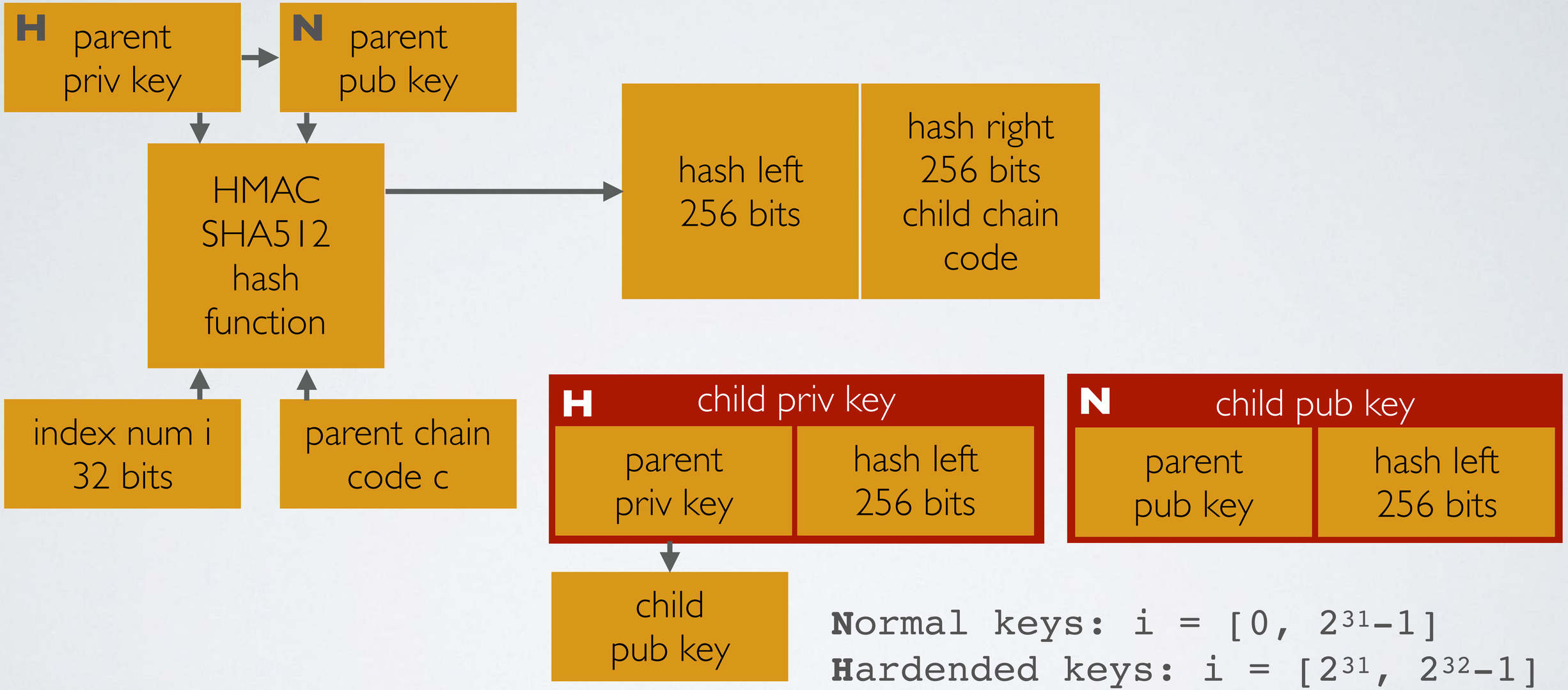
- Online web application:
  https://www.mobilefish.com/download/ethereum/hd_wallet.html

BIP-32

- BIP-32 explains how master keys and master chain code are created from a BIP-39 seed. The chain code is used as entropy in the **C**hild **K**ey **D**erivation (CDK) function.



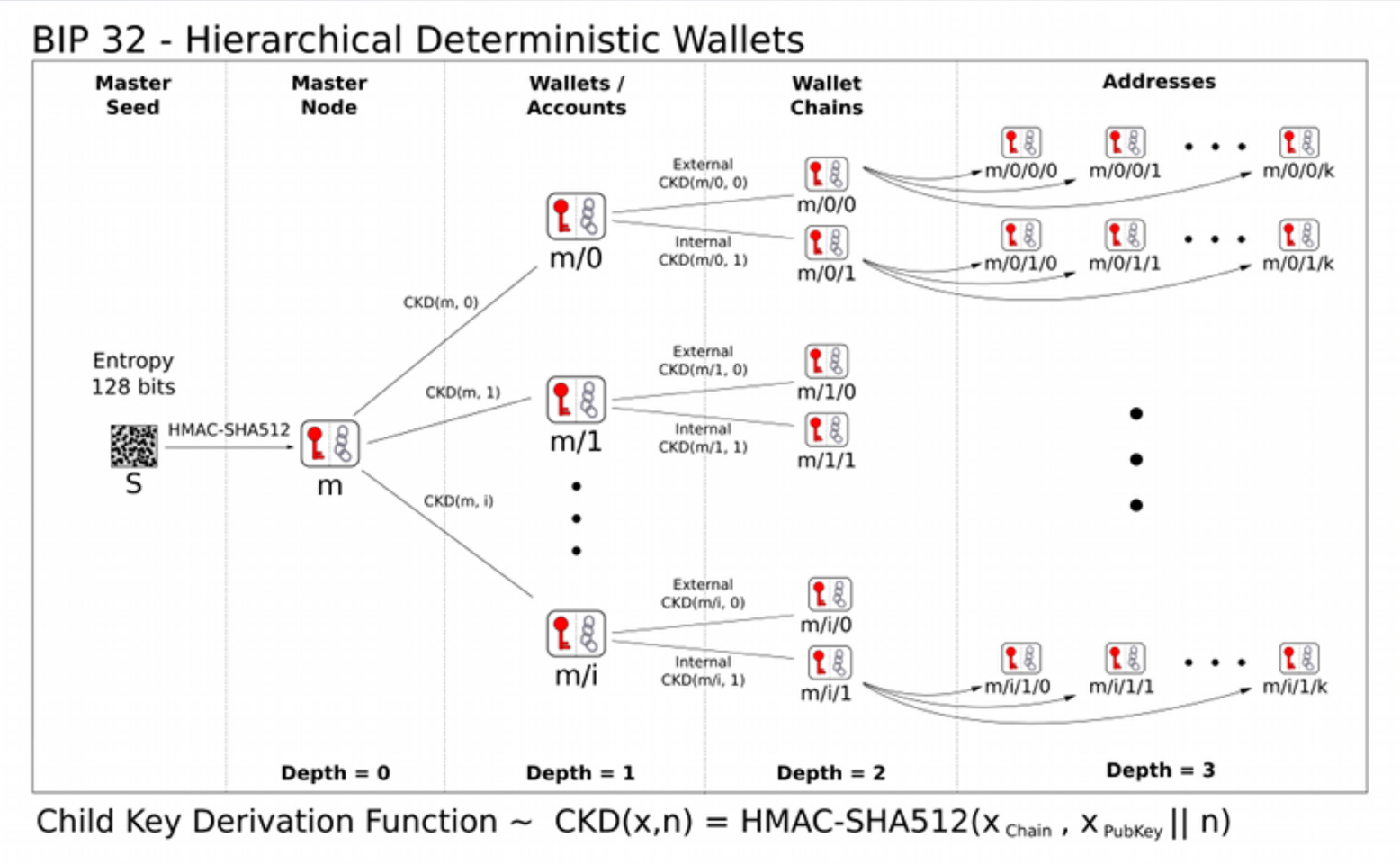
BIP-32 CHILD KEY DERIVATION (CKD)



Normal keys: $i = [0, 2^{31}-1]$

Hardended keys: $i = [2^{31}, 2^{32}-1]$

BIP-32 CHILD KEY DERIVATION (CKD)



Source: https://github.com/sipa/bips/blob/bip32update/bip-0032/derivation.png

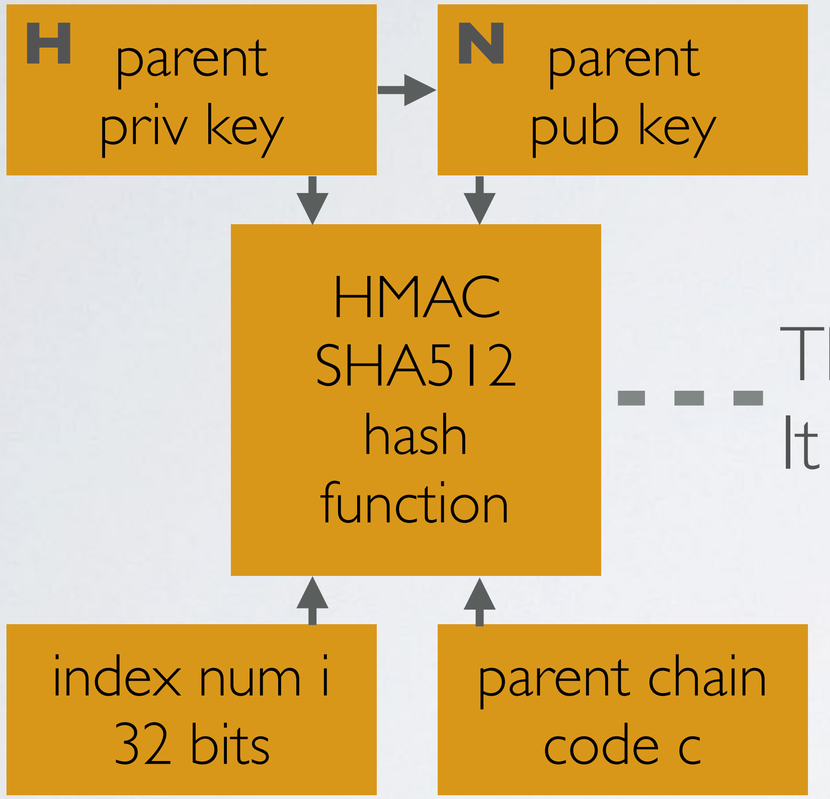
BIP-32

- Using different index numbers (i) will create different unlinkable child keys from the same parent keys.

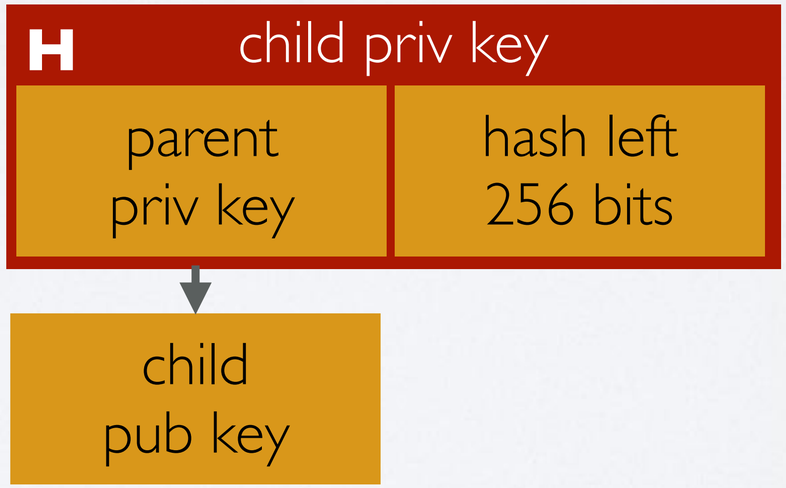
- Repeating the procedure for the child keys using the child chain code will create unlinkable grandchild keys.

- By changing the chain code, a new node (aka wallet) is created.

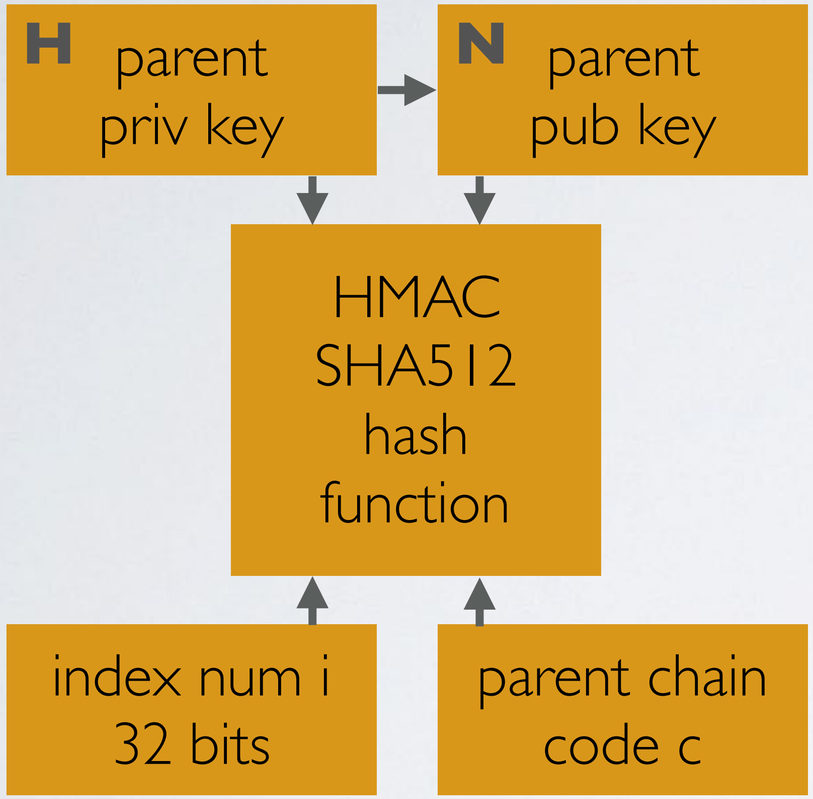
BIP-32 CHILD KEY DERIVATION (CKD)



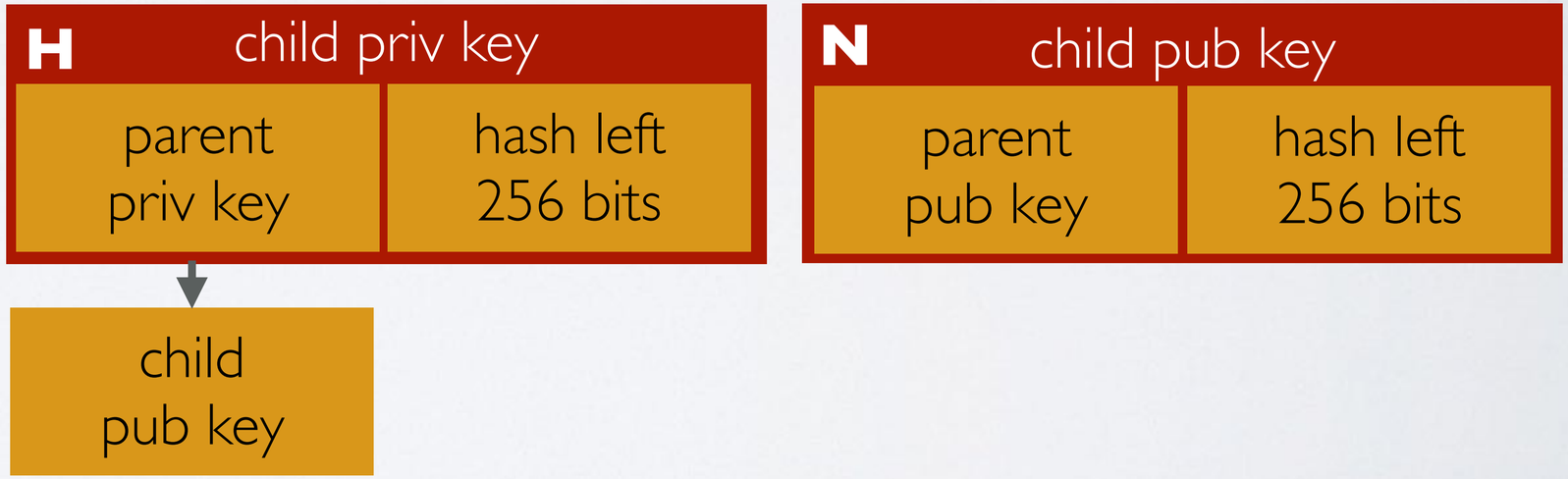
This function is also known as the CKD function.
It derives child keys from parent keys.



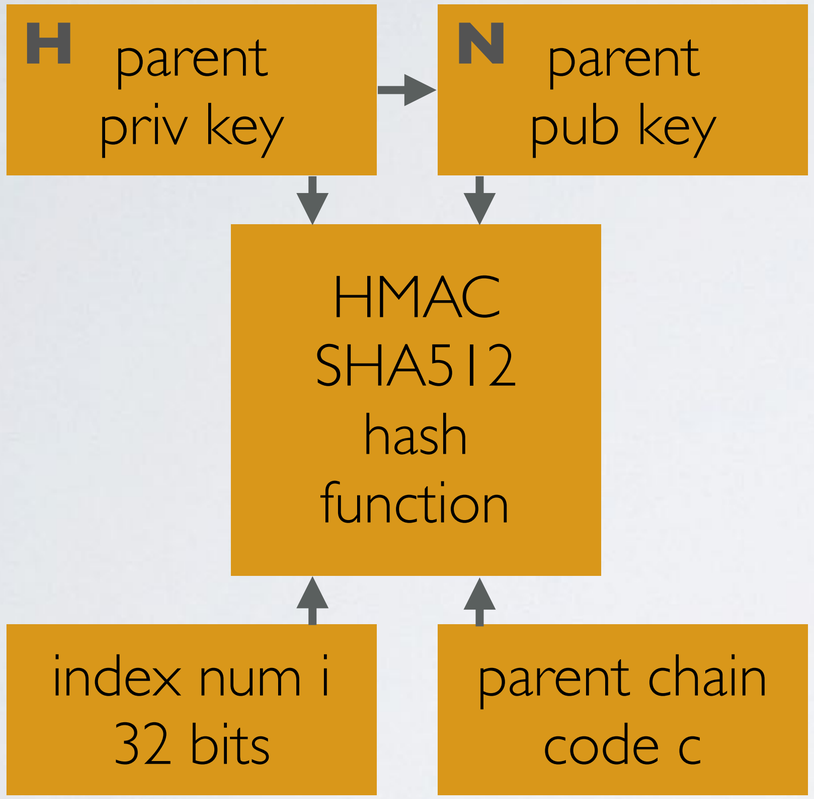
BIP-32 CHILD KEY DERIVATION (CKD)



These are also called derived child keys



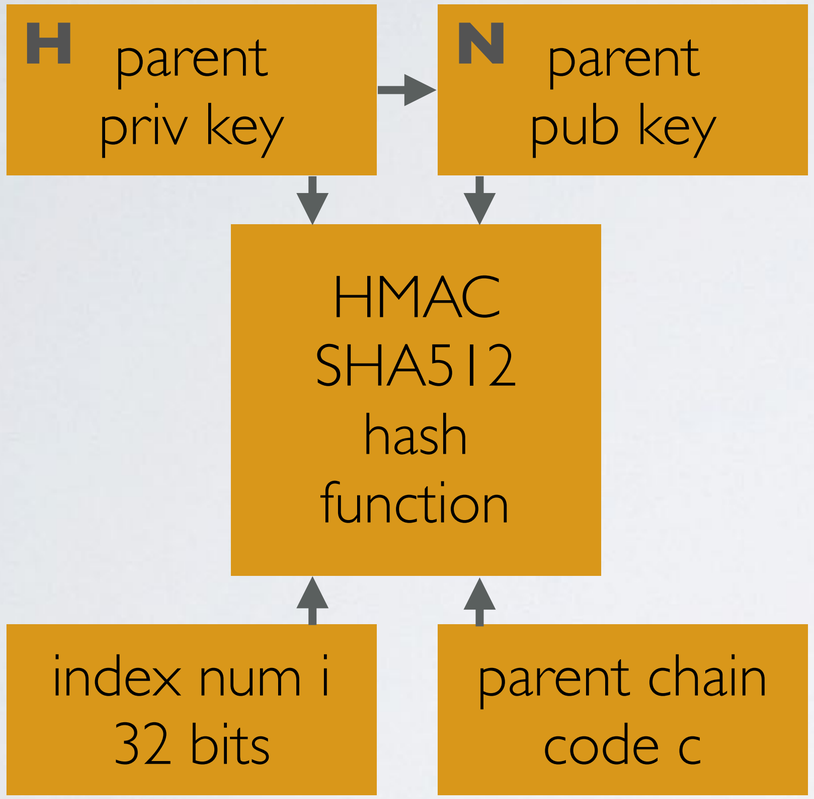
BIP-32 CHILD KEY DERIVATION (CKD)



Extended private key (**xprv**) =
parent private key + parent chain code

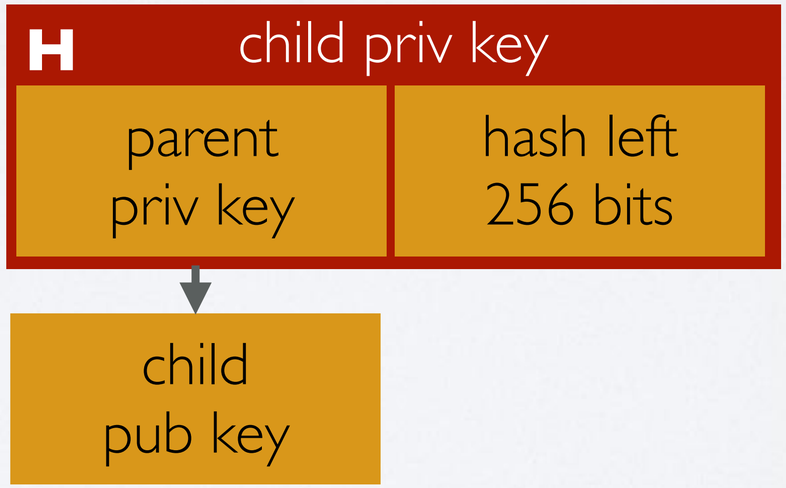
Extended public key (**xpub**) =
parent public key + parent chain code

BIP-32 CHILD KEY DERIVATION (CKD)



xprv keys can create a complete branch with child private keys and child public keys

xpub keys creates only a branch of child public keys. It can not create hardened keys.



BIP-32

- Extended public (xpub) keys can only generate public keys. This is perfect if you want a wallet which can only watch your account balances or receive coins but you can not sign any transactions because there are no private keys available.

- A wallet created with an extended private (xprv) key can generate public keys and private keys.

BIP-44

- BIP-44 defines a **specific** logical hierarchy for deterministic wallets based on an algorithm described in BIP-32.

- More information about BIP-44 can be found at: https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki

BIP-44

- BIP-44 uses the following derivation path:
  `m/purpose'/coin_type'/account'/change/address_index`

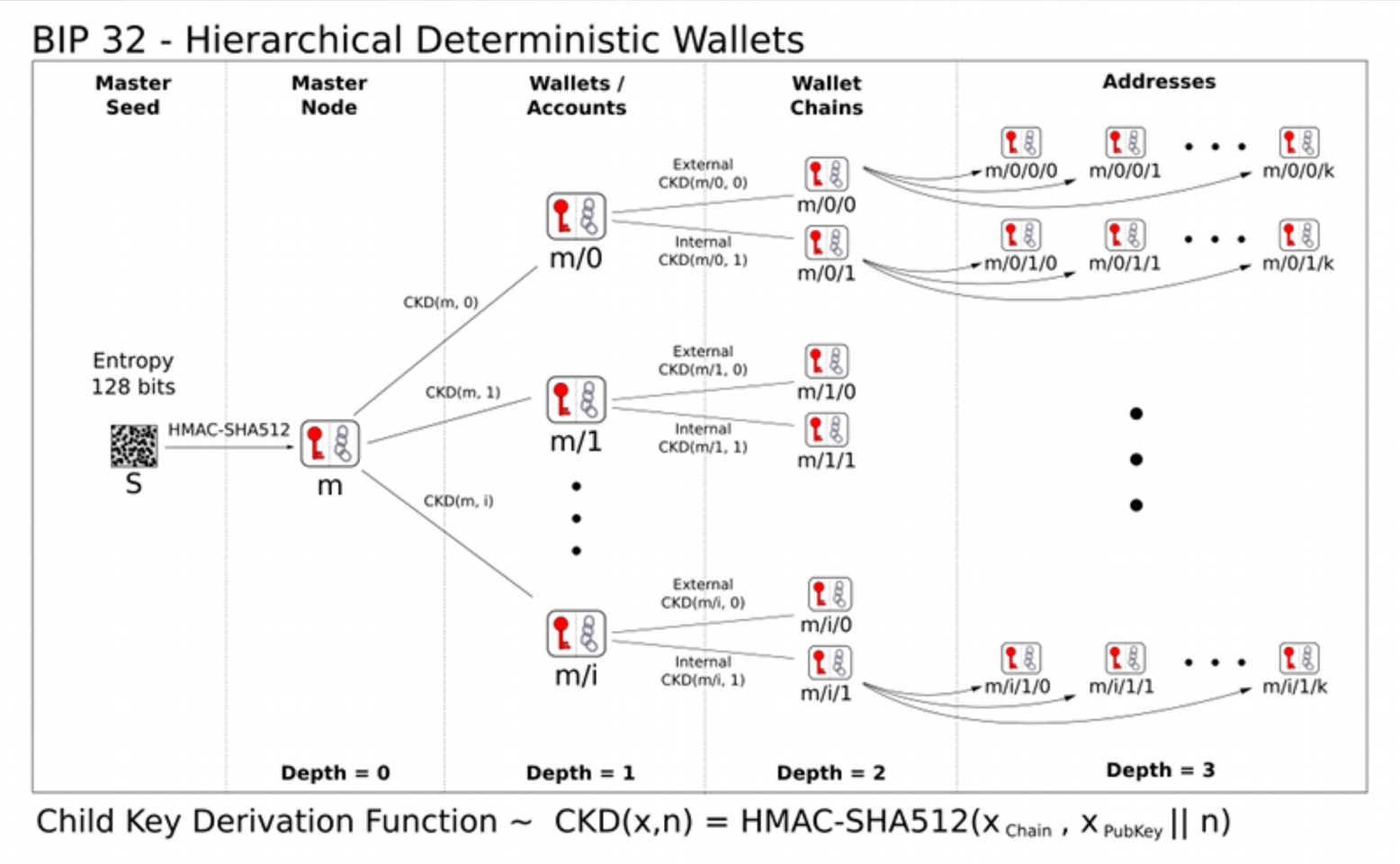
- The letter m denotes the master node and all hierarchical levels are separated with slashes (/). m is level 0, purpose is level 1, coin_type is level 2 etc.

- The purpose scheme is described in BIP-43.
  Because we are using BIP**44** scheme we should use **44'**.

- The apostrophe (for example in **`purpose'`**) indicates hardened derivation.

- The default registered coin types for usage in level 2 of BIP-44 can be found at:
  https://github.com/satoshilabs/slips/blob/master/slip-0044.md

BIP-44



Source: https://github.com/sipa/bips/blob/bip32update/bip-0032/derivation.png

BIP-44

- **`m/purpose'/coin_type'/account'/change/address_index`**
  account level can be seen as bank account types, for example payment account, savings account etc.

- change level is also known as "external / internal level" where external (0) is used for addresses that are meant to be visible outside of the wallet (receiving payments) and internal (1) is used for addresses which are not meant to be visible outside of the wallet (signing transactions).

- address_index is a sequence of addresses starting at 0.

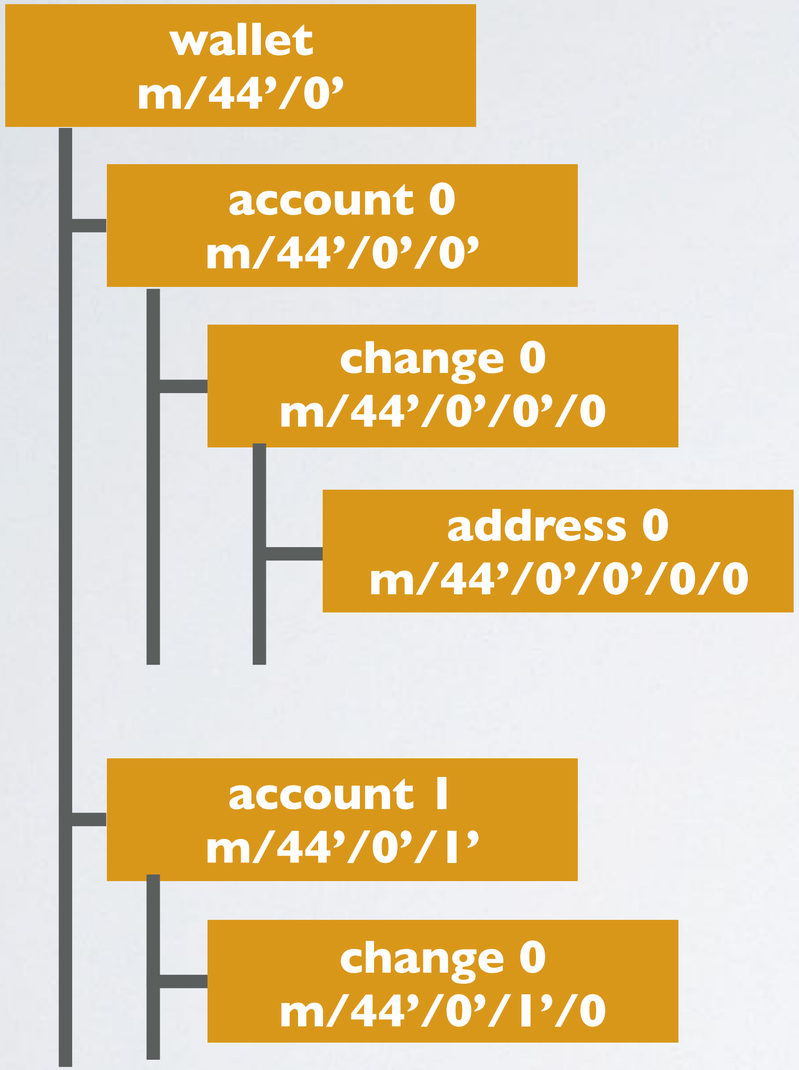
- The keys in HD wallets are identified using a path naming convention.
  For example: m/44'/60'/0'/1/4. The fifth address at change level 1.

BIP-44

m/44'/60'/0'/0/1

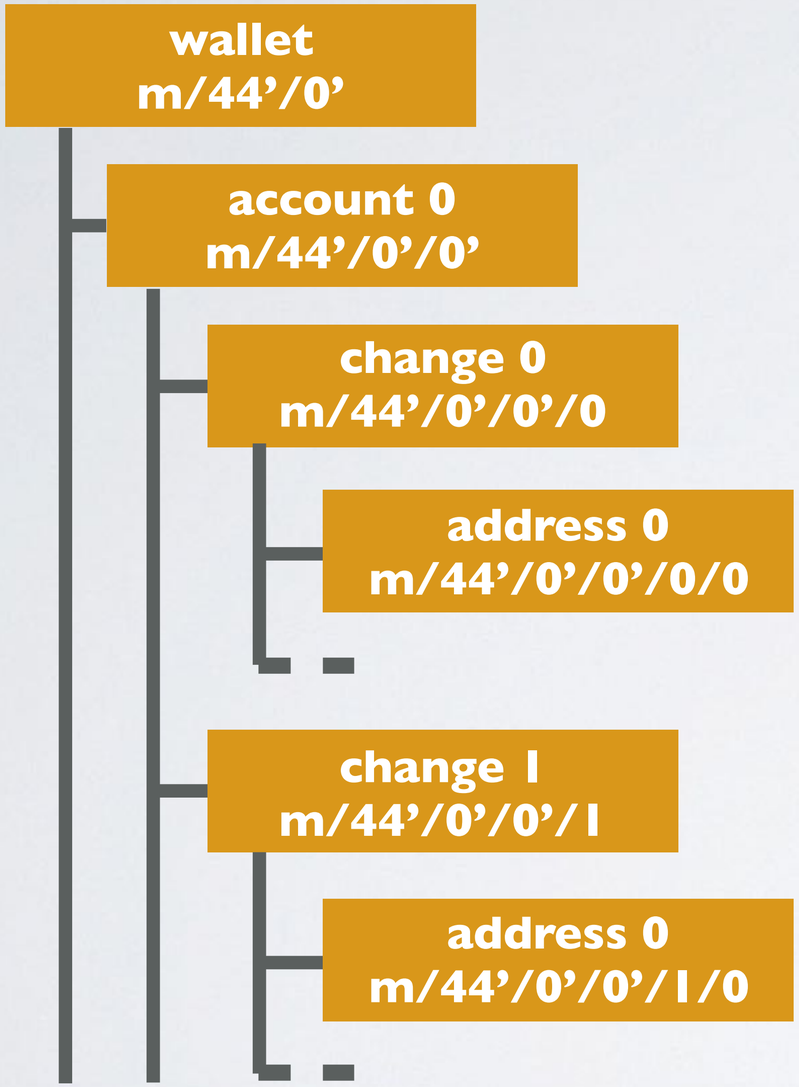
Hardened derivation Normal derivation

BIP-44 DERIVATION PATH EXAMPLE 1



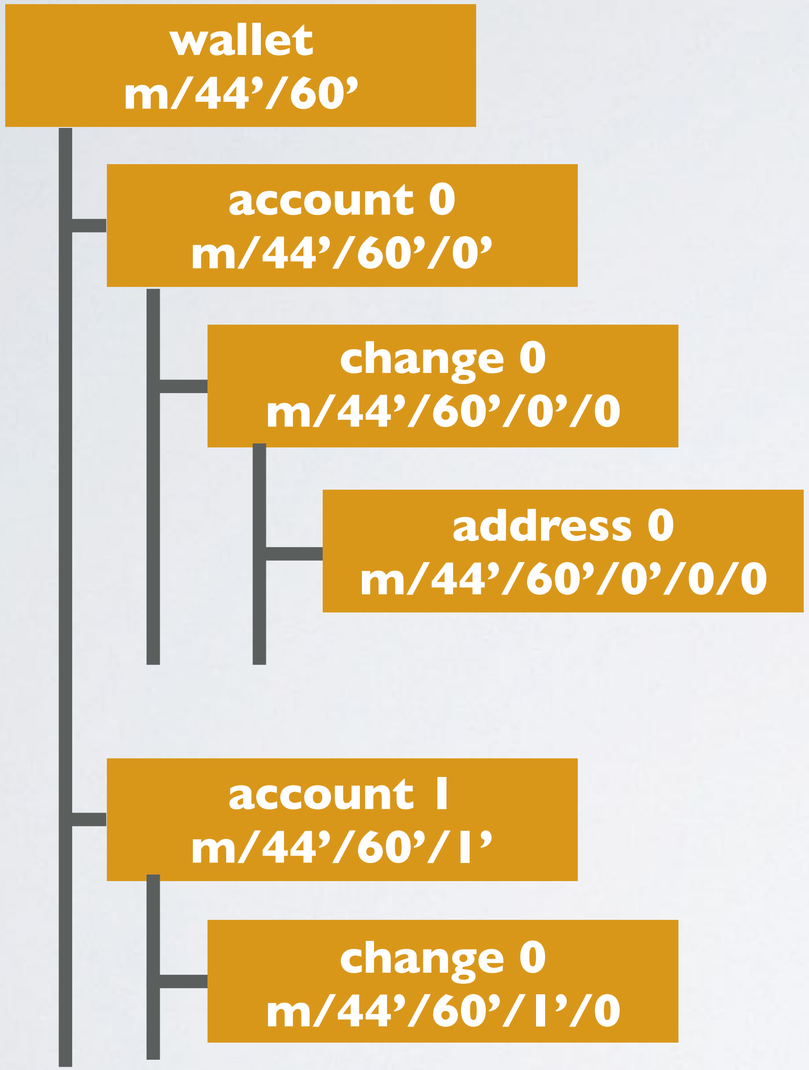
m/44'/0'/0'/0/0
purpose = 44
coin_type = 0 (Bitcoin)
account = 0
change = 0
address_index = 0

BIP-44 DERIVATION PATH EXAMPLE 1



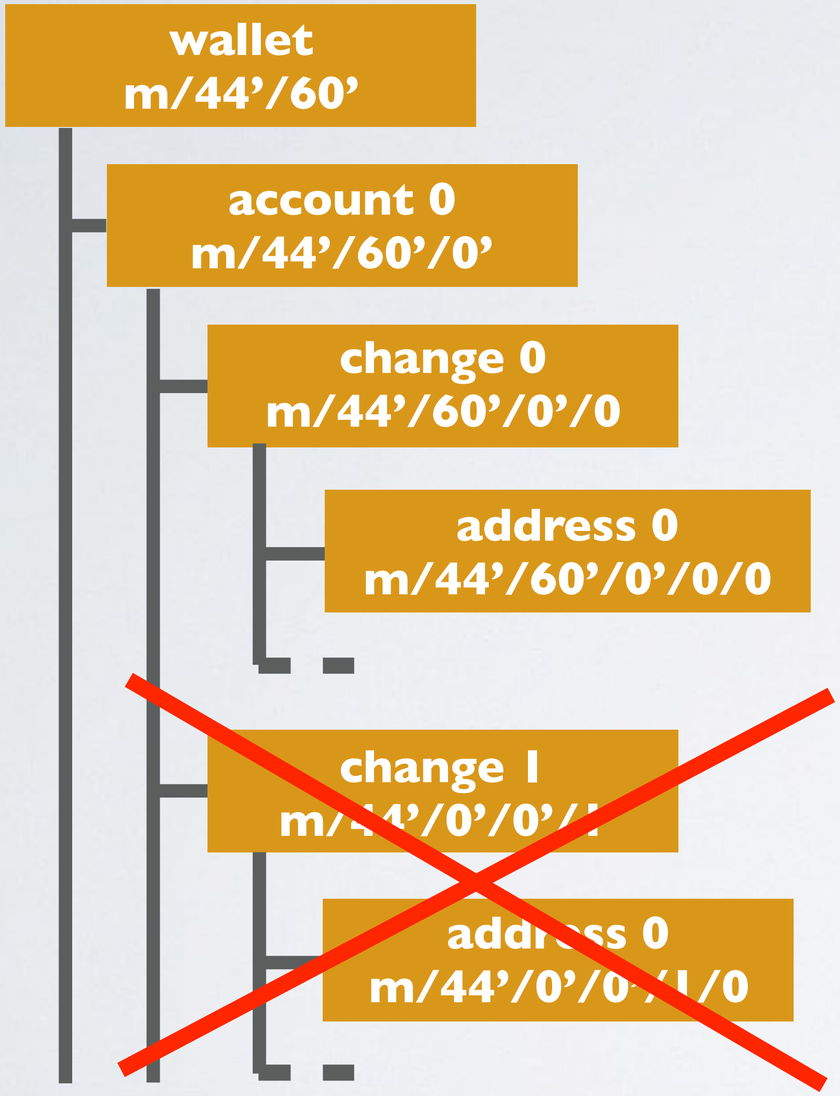
m/44'/0'/0'/0/0
purpose = 44
coin_type = 0 (Bitcoin)
account = 0
change = 0
address_index = 0

BIP-44 DERIVATION PATH EXAMPLE 2



m/44'/60'/0'/0/0
purpose = 44
coin_type = 60 (Ethereum ETH)
account = 0
change = 0
address_index = 0

BIP-44 DERIVATION PATH EXAMPLE 2



m/44'/60'/0'/0/0
purpose = 44
coin_type = 60 (Ethereum ETH)
account = 0
change = 0
address_index = 0

BIP-44 XPRV AND XPUB DEMONSTRATION

- Suppose a webshop wants to receive payments in only in ETH.
  For each payment received a different Ethereum address must be used.

- The webshop should use a wallet containing only public keys.
  The webshop wallet uses a xpub key with derivation path: **m/44'/60'/0'/0**

- This wallet creates the following addresses:
  m/44'/60'/0'/0/0
  m/44'/60'/0'/0/1
  m/44'/60'/0'/0/2
  m/44'/60'/0'/0/3
  m/44'/60'/0'/0/……

BIP-44 XPRV AND XPUB DEMONSTRATION

- The accounting department, separated from the webshop, uses another wallet containing the same public keys AND accompanied private keys.

- The accounting department can transfer payments made on these public addresses and transfer it to a separate accounting address. This is possible because they have access to the private keys.

- The accounting department wallet uses a xprv key with the same derivation path: **m/44'/60'/0'/0**

BIP-32 RISK

- **m/purpose'/coin_type'/account'/change**

- The change level is not hardened. If a hacker gets it hands on any child private key (for example: m/44'/60'/0'/0/2) AND the account xpub key, the hacker can recompute the account xprv key and thus have access to every private and public key descending from the account level.

- Read: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki#implications

- **xpub keys only generates public keys but you must be aware of the above mentioned risk.**