BLOCKCHAIN TUTORIAL 28

BIP39 MNEMONIC WORDS

emotion

develop

win

pave

upgrade

allow

junior

box

volcano

dirt

athlete

question

INTRO

- In this video I will explain:
- What a wallet is.
- What the difference is between a non-deterministic wallet and a deterministic wallet.
- What mnemonic words are.
- What BIP-39 is.

WHAT IS A WALLET

- A wallet stores private keys.
- The public addresses are automatically derived from the private keys.
- A wallet does not store coins (Bitcoin, Litecoin, Ether etc.).
- If you open your Bitcoin wallet and one of your Bitcoin addresses shows that is has a balance of 5 BTC, than these bitcoins are not actually stored in your wallet. It means that these 5 bitcoins were transferred to your Bitcoin address during a transaction.This transaction (**TX**) information is stored on the blockchain.
- Your wallet queries the blockchain and searches for **U**nspent **TX O**utputs (UTXO) for all your Bitcoin addresses to display their balances.

WHAT IS A WALLET

- The bitcoins on these UTXO can be unlocked and transferred to another Bitcoin address using the private keys stored in the wallet.

- The word wallet is misleading, it just stores private keys and not the coins.

- If you lose your wallet, you lose your private keys and if you lose your private keys you can not unlock UTXO. This means you have lost access to your coins.

- However if you can restore your private keys (for example you have made a backup) you can always access your coins.

NON-DETERMINISTIC WALLET

- Wallets stores private keys but they also **create** these private keys.

- A non-deterministic wallet does the following:
  It generates private key 1 which in turn creates a corresponding public address 1
  It generates private key 2 which in turn creates a corresponding public address 2
  etc…

- The private keys are randomly generated numbers which are not related to each other.

- You can not derive these private keys with an algorithm.
  Hence the words "non-deterministic".

NON-DETERMINISTIC WALLET

- If you use a non-deterministic wallet you must make regular backups of these private keys.

- If you have problems with your wallet, you can restore your wallet by importing the backupped private keys.

- Explaining a non-deterministic wallet (for educational purpose only), see: https://www.mobilefish.com/services/cryptocurrency/cryptocurrency.html

DETERMINISTIC WALLET

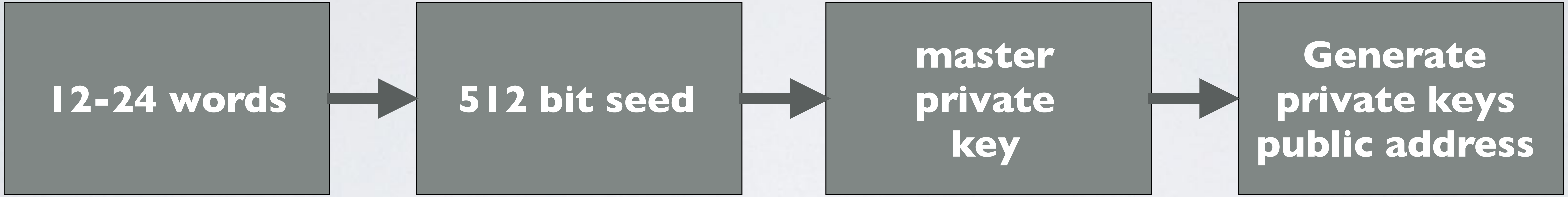
- A deterministic wallet uses 12 - 24 words to create a 512 bit seed.
  For example: choice fatal slab rookie …

- These words are called mnemonic words, because they are more easily to remember than this long hexadecimal string "BF8526205D0B2E227C52E411472FAD5CA8CAE0285BBEBD566F2B".

- The 512 bit seed is used to create a master private key.

- This master private key in turn is used to create private keys and corresponding public addresses.

DETERMINISTIC WALLET



12-24 words

512 bit seed

master private key

Generate private keys public address

DETERMINISTIC WALLET

- Generally speaking using these 12 - 24 words will complete restore your wallet with exactly the same private keys and corresponding public addresses.
  Hence the word "deterministic".

- It is imperative that you safely store these 12 - 24 words, without it you have no access to your private keys.

- To see how an Ethereum deterministic wallet works, see YouTube movie:
  "MetaMask: How to restore your accounts"
  https://youtu.be/cqz8-hOz_nk

BIP-39

- The acronym BIP means **B**itcoin **I**mprovement **P**roposal.

- BIPs are design documents for introducing features or information to Bitcoin.
  An overview can be found at: https://github.com/bitcoin/bips

- BIP-39 describes the implementation of mnemonic words for the generation of deterministic wallets.

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

- BIP-39 is becoming an industry standard which is not only used for Bitcoin wallets but it is also used in Ethereum, Dash and other Altcoin wallets.

BIP-39

- ENT = Random number (Allowed lengths: 128, 160, 192, 224, 256 bits) multiple of 32.
  Example: ENT (128 bits, 16 bytes) =
  [10101111, 00110011,.., 11110000, 01011110]
  More bits means better security but means more mnemonic words.

- Checksum Length CL (bits) = ENT length in bits / 32
  Example: CL = 128 / 32 = 4 bits

- HASH = SHA256(ENT)
  Example:
  HASH = SHA256([10101111, 00110011,.., 11110000, 01011110])
  HASH =
  321e9b91a5647270522e87959d1a56ea3f7601f0e32e837aa8bf420558a2df6f

BIP-39

- CHECKSUM CS = Take the first CL bits of the HASH
  Example: HASH = **3**21e9b91a5647270522e87959d1a56ea3f7601f0e3…
  CS = **0011**0010…

- ENT_CS = Append the checksum at the end of the random number = ENT + CS
  Example: `[10101111, `**`001`**`10011,.., 11110000, 0`**`1011110`**`]` **0011**

- Split ENT_CS in groups of 11 bits.
  Example: `[10101111`**`001`**`, 10011.., ..111100000, `**`1011110`** **0011**`]`

- Word Index = Convert each 11 bits into integers.
  Example: 1401,…..1507

BIP-39

- Number of combinations with 11 bits = 2^{11}= 2048
  The value range is: 0 - 2047

- The wordlist can be found at:
  https://github.com/bitcoin/bips/blob/master/bip-0039/bip-0039-wordlists.md

- The wordlist consists of 2048 words. These words and the order must not be changed.
  The wordlist is also available in other languages.

- Use the wordlist to find the words for each word index value.
  Example: 1401 (= quality),…..1507 (round)
  Mnemonic words = [quality, …, round]

BIP-39

ENT (bits)	# of words	combinations	combinations
128	12	2048^{12}	$\sim 5.4 \times 10^{39}$
160	15	2048^{15}	$\sim 4.6 \times 10^{49}$
192	18	2048^{18}	$\sim 4.0 \times 10^{59}$
224	21	2048^{21}	$\sim 3.4 \times 10^{69}$
256	24	2048^{24}	$\sim 2.9 \times 10^{79}$

BIP-39

- To give you an idea how big these numbers are, the number of atoms in the entire observable universe is estimated to be within the range of 10^{78} to 10^{82}.

- Concatenate these words into one long string. Mnemonic words = [quality, …, round]
  Mnemonic phrase = "**quality**couple…conduct**round**"

- Optionally for additional security you can allow users to enter a passphrase.
  For example: Passphrase = "MyBigSecret"

- The word "mnemonic" together with the passphrase is used as salt.
  If no passphrase is used the passphrase is an empty string "".
  For example Salt = "mnemonic" + passphrase
  Salt = "mnemonicMyBigSecret"

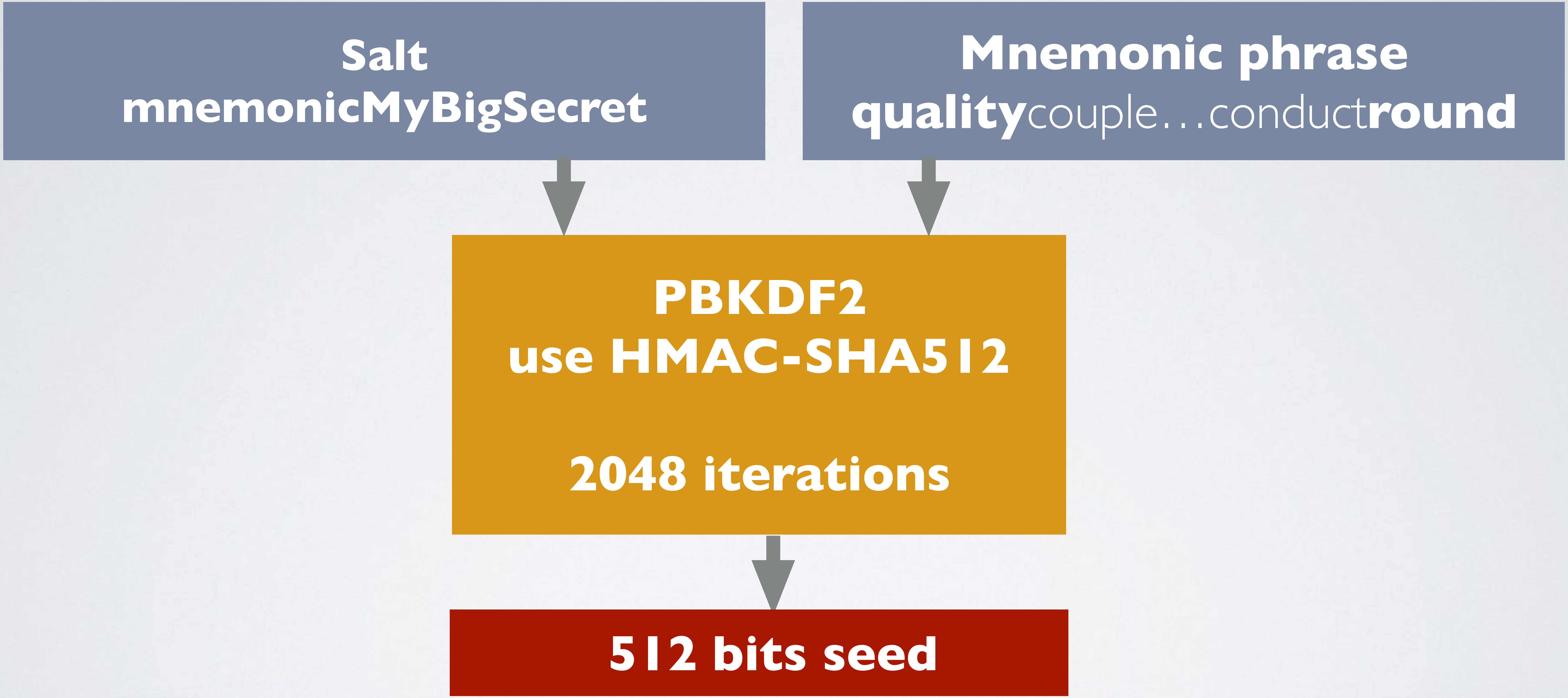
BIP-39

- Use the **P**assword-**B**ased **K**ey **D**erivation **F**unction **2** (PBKDF2) together with the mnemonic phrase and salt to produce a 512 bits seed. The iteration count is set to 2048 and HMAC-SHA512 is used as the pseudo-random function.

- If an attacker gets its hands on your mnemonic words the passphrase (it you have set it) will prevent the attacker to access the private keys.

- PBKDF2 is purposefully made slow to make brute force dictionary attack very difficult.

BIP-39



Salt

mnemonicMyBigSecret

Mnemonic phrase

qualitycouple…conduct**round**

PBKDF2

use HMAC-SHA512

2048 iterations

512 bits seed

BIP-39

- The 512 bit seed is used to generate deterministic wallets.

- How to generate deterministic wallets is explained in BIP-32 and BIP-44.

- It is important to know that each time you enter a different passphrase it will generate a valid 512 bit seed and thus a valid wallet with valid public and private key pairs.

- This feature can help you limit your loss after a 5$ wrench attack. You can setup a second deterministic wallet with some coins to satisfy the attacker. If you do not know what a 5$ wrench attack is watch this comic: https://xkcd.com/538/

- Storing your passphrase at the same location as your mnemonic words is not recommended and beats the purpose.

BIP-39

- But if you lose your passphrase, you have lost access to your coins.

- A JavaScript implementation of BIP-39 can be found at:
  https://github.com/bitcoinjs/bip39

- How this JavaScript library is used see:
  https://www.mobilefish.com/download/ethereum/bip39.html

- A Mnemonic Code Converter web application can be found at:
  https://iancoleman.github.io/bip39

WHAT NEXT

- In my next video about "Hierarchical Deterministic Wallet" I will explain how the 512 bit seed is used to create the private keys and corresponding public addresses.