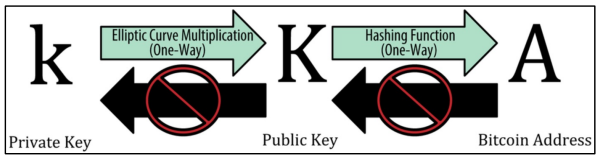
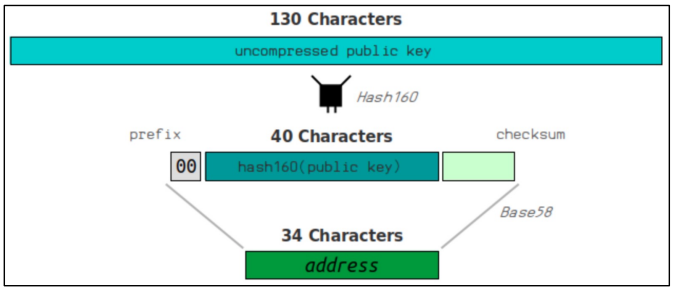
Table of contents

Bitcoin addresses

- Bitcoin address
  - A private key is a secret random number
  - A public key is derived from private key
    - Using elliptic curve multiplication, irreversible
  - An address is derived from public key
  - One way
    - Don’t lose private keys

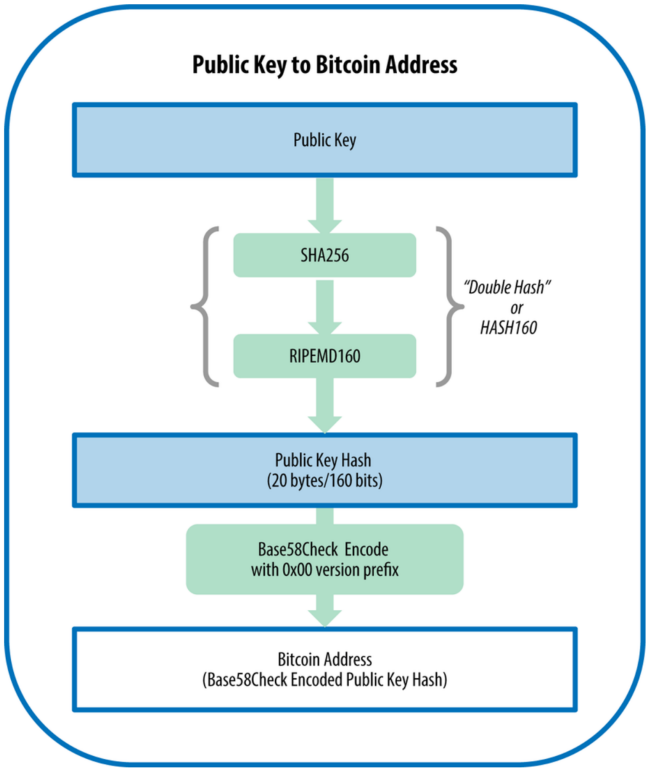




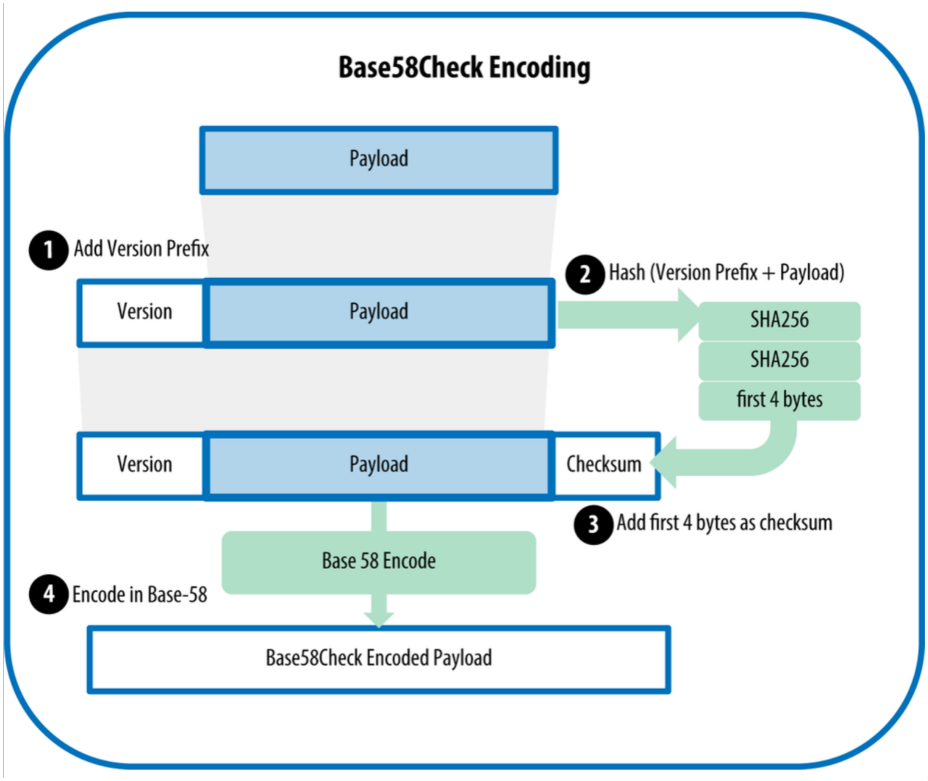
Bitcoin addresses

- Bitcoin address
  - 26-35 alphanumeric characters
  - Usually produced from public keys/scripts of users
    - *Base58CheckEncode(RIPEMD160(SHA256(PublicKey or Script)))*
  - Avoid sharing longer public keys | publicly shareable
  - Signify a private/public key pair
    - Payment script (in case of P2SH)

Bitcoin addresses



Bitcoin addresses



Bitcoin addresses

- Base58Check encoding
  - Uses 58 characters
    - Set of lowercase, uppercase English letters, and digit
      - Except 0 (number zero), O (capital o), l (lower L), I (capital i)

Bitcoin addresses

- Common address types and their prefixes

Type	Version prefix (hex)	Base-58 result's prefix
P2PKH address	0x00	1
P2SH address	0x05	3
P2PKH Testnet address	0x6F	m or n
P2SH Testnet address	0xC4	2
Private key WIF	0x80	5, K, or L
BIP32 extended public key	0x0488B21E	xpub
Bech32 (SegWit)	0x0303000203 (for checksum only)	bc1

Bitcoin addresses

- Vanity address
  - Address that starts with some human-meaningful text
    - 1BigHit9XN991TquuvZpDB9dSV52v9pY9w
    - 1234mNnAPb8YnCsbacaNhB4BqwxB4U4321
  - Repeatedly generate candidate private keys
    - 58 possibilities for every character
      - Testing billions of keys
    - k-characters string at start, generate 58^k addresses (on avg)

Bitcoin addresses

- Vanity address
  - Tools/technique
    - In Bitcoin, Private key x | public key is g^x | Address $H(g^x)$
      - Exponentiation shows scalar multiplication in elliptic curve group
      - Exponentiation slow
      - Generate x
        - Try $x+1$, ...
        - $g^{x+1} = x.g^x$
        - Multiplication is faster than exponentiation
      - 12 (1 is fixed) => 1 in 58 keys
      - 123 (1 is fixed) => 1 in 3,364 keys
  - Deceiving trust with resembling vanity addresses

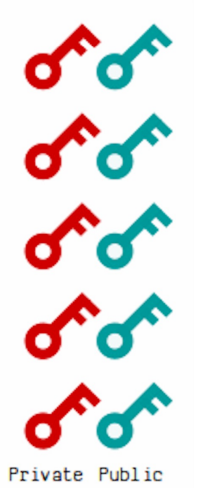
Wallets

- How to safely store private key?
  - Hot storage
    - Storing keys in an online computer
      - Risky
    - Small amount
  - Cold storage
    - Offline (e.g., paper?)
      - Not so convenient
    - Large/archival amount

Wallets

- Wallets
  - Containers for private keys
  - Implemented as structured files or simple databases
  - Contain keys, not coins

- Nondeterministic wallets
  - Random | Type-0 | JBOK
  - Collections of randomly generated private keys
    - Needs regular backup
  - "Use an address only once" is difficult here

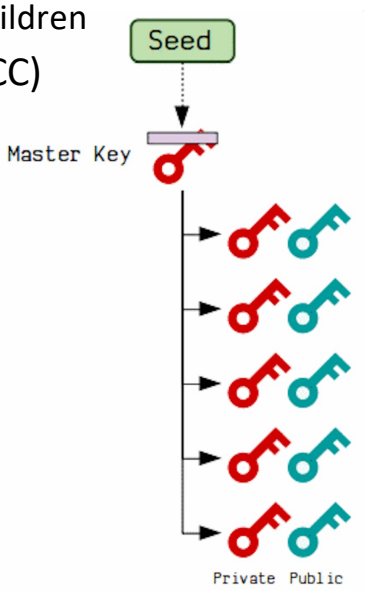


Wallets

- Deterministic wallets
  - Seeded | Type-1
  - Contains private keys that are derived from a seed
    - Through a one-way hash function
  - Seed can recover all keys, import/export wallet
    - Single backup
    - 12-24 Mnemonic words encode random number (seed)

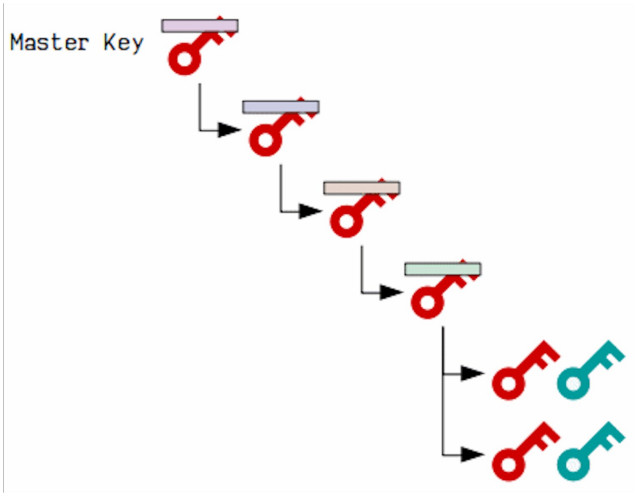
Wallets

- Hierarchical Deterministic (HD) wallets
  - Type-2
  - Keys derived in a tree fashion
    - Seed => Parent/Master => Children => Grandchildren
  - Possibility with different keys and ChainCode (CC)
    - Parent private key+CC => Child private key+CC
    - Parent public key+CC => Child public key+CC
    - Parent private key+CC => Child public key+CC



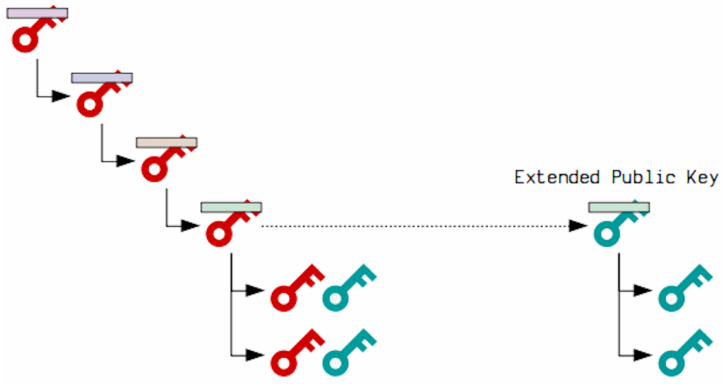
Wallets

- Hierarchical Deterministic (HD) wallets
  - Single backup
  - Organization
    - Each child key can generate its own key-pairs
      - Use different tree branches for different purpose/department



Wallets

- Hierarchical Deterministic (HD) wallets
  - Generate public keys independently (without private keys)
    - Parent public key+CC => Child public key+CC
    - Store parent public key on server, generate new receiving addresses



  - One can't spend funds associated with child address using parent private key

Wallets

- Hardware wallet
  - Secure hardware device
  - Storage in protected area of microcontroller
    - Can't transfer keys out of device in plaintext
    - Immune to viruses (unlike software wallets)
  - Open source software
  - KeepKey uses USB connection, authorization via it's display/button
  - Ledger Nano X uses Bluetooth to connect with mobile devices

Wallets

- Paper wallets
  - Cold storage
    - Offline BTC storage
  - Private keys printed on paper
    - May also include the address
      - Can be derived from private key

Wallets

- Paper wallets
  - Security against loss of key (e.g., computer mishap)
  - Paper wallet keys are generated offline (no internet)
    - Client-side JavaScripts, and then printed
    - Never stored on a computer (no hackers, keyloggers)
  - Same issues as a paper
    - Vulnerable to theft, copy

Wallets

- Paper wallets
  - Paper wallets come in many shapes, sizes, and designs

  - Sophisticated paper wallets use BIP0038 encrypted private keys
    - Printed keys are protected by a passphrase
  - Deposit funds into paper wallet several times
    - Withdraw all funds at once, spending everything
      - Spending exposes private key online

Wallets

- Online wallets
  - Like a local wallet that you manage yourself
    - Info is stored in cloud
    - Access via a web interface
  - Service provider
    - Delivers the code that runs on your browser/app
    - Stores your keys
  - Convenient, no s/w installation
  - Trust them, their code, their storage encryption

Wallets

- Bitcoin exchanges
  - Accept deposits in BTC, USD, EUR, etc.
  - Allow banking-like activities (send/receive BTC, convert funds)
  - Tx among customers of same exchange do not happen on blockchain
  - Connect Bitcoin economy with fiat currency economy
  - Risks
    - Non-regulated
    - Reserve fraction
      - Proof of reserve
        - Self Tx of an amount as a proof of reserve
      - Proof of liabilities
        - Merkle tree
          - Show customers partial tree from users' leaf up to root