Mastering Blockchain

Ch.4 Keys, Addresses, Wallets

Kun-jung Wu

Department of Economics
National Taiwan University

R06323008@ntu.edu.tw

February 2, 2018

Overview

1. 1. Introduction

2. 2. Private and Public Keys

3. 3. Addresses

1. Introduction

Keys & Addresses

- Private Key: PIN number of bank account (singing transactions)
- Public Key: Bank account number (receive bitcoin)
- Address: The title of the beneficiary
- Wallet: the file to store keys

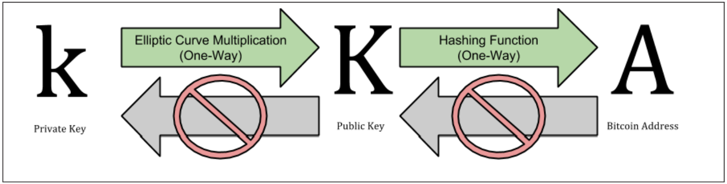


Figure 4-1. Private Key, Public Key and Bitcoin Address

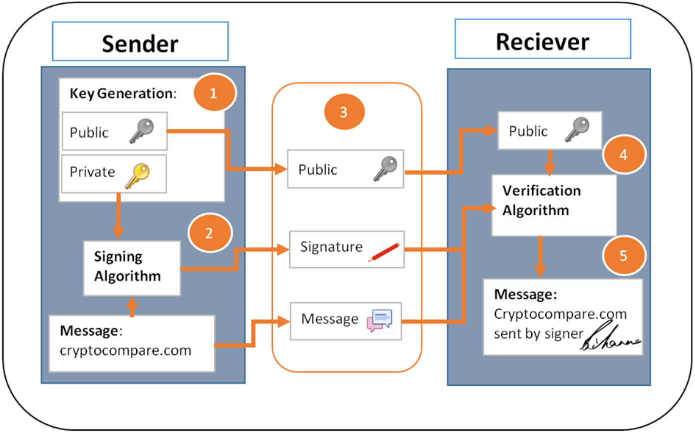
Not every address is linked to a specific public key. It can also link to a script. (See Ch.5)

2. Private and Public Keys

Public Key Cryptography

- The private key to be used to generate signatures on transactions.
- This signature can be validated against the public key without revealing the private key.
- When spending bitcoins, the current owner presents their public key and a signature (different each time)
- Everyone in the bitcoin network can verify and accept the transaction as valid,

2. Private and Public Keys



2. Private and Public Keys

Private Key Generation

- Pick a 256-bit number randomly.
- From 1 to n-1 (n $= 1.158 \times 10^{77}$, slightly less than 2^{256})
- Ex: (in hexadecimal form)
  1E99423A4ED27608A15A2616A2B0E9E5
  2CED330AC530EDCC32C8FFC6A526AEDD

2. Private and Public Keys

Public Key Generated by Elliptic Curve Cryptography

- $y^2 \mod p = \left(x^3 + 7\right) \mod p$
- $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- $\mathrm{K} = \mathrm{k} * \mathrm{G}$
- `https://en.wikipedia.org/wiki/Elliptic_curve`

secp256k1 Standard

- More details on: `https://eng.paxos.com/blockchain-101-elliptic-curve-cryptography`

2. Private and Public Keys

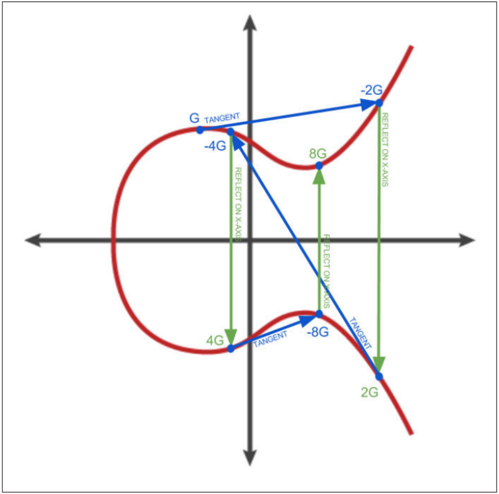


Figure 4-4. Elliptic Curve Cryptography: Visualizing the multiplication of a point G by an integer k on an elliptic curve

2. Private and Public Keys

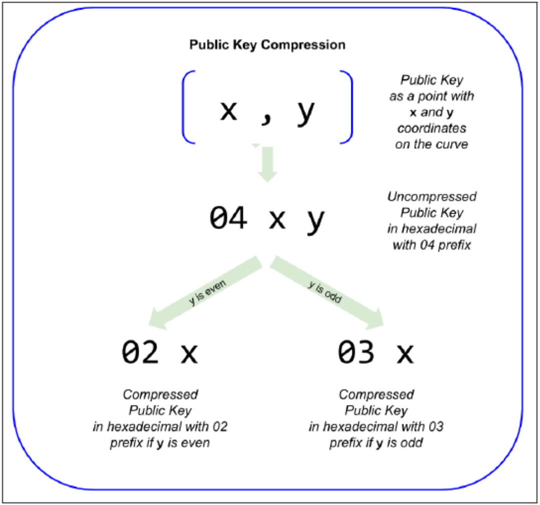

Public Key Compression
x , y
Public Key as a point with x and y coordinates on the curve
04 x y
Uncompressed Public Key in hexadecimal with 04 prefix
y is even
y is odd
02 x
03 x
Compressed Public Key in hexadecimal with 02 prefix if y is even
Compressed Public Key in hexadecimal with 03 prefix if y is odd

Figure 4-7. Public Key Compression

3. Addresses

Address is not public key

- Bitcoin Address allows a variety of transactions.
- Address can represent a public key
- Or, A script.

Types of Transactions

- Pay to Public Key Hash (P2PKH)
- Pay to Public Hash (P2PH) (obsolete)
- Pay to Script (P2SH)
- Multi-Sig

3. Addresses

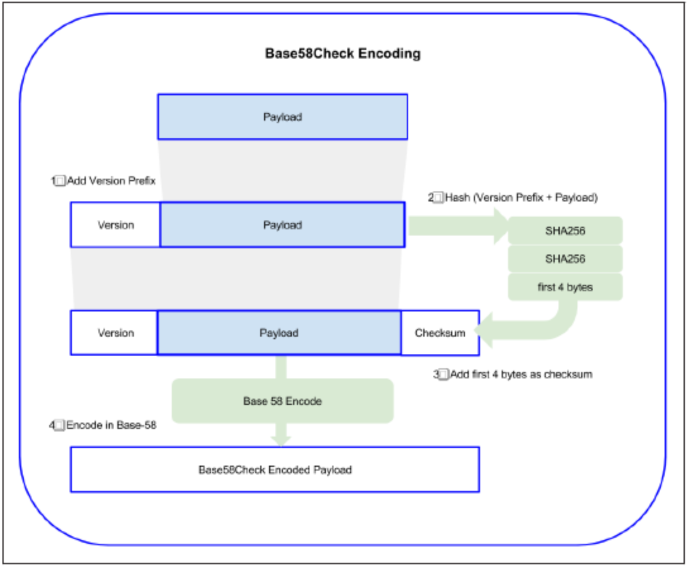
"Double Hash"

- Secure Hash Algorithm-256 (SHA256)
- RACE Integrity Primitives Evaluation Message Digest (RIPEMD)
- A = RIPEMD160(SHA256(K))
- Output is a 160-bit number

"Base58Check Encoding"

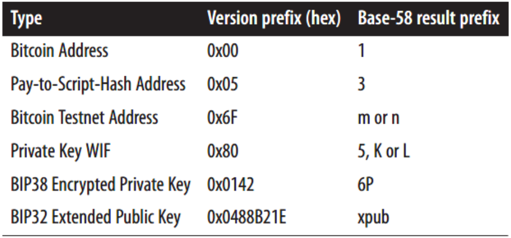
- 123456789ABCDEFGHJKLMNPQRSTUVWXYZ abcdefghijkmnopqrstuvwxyz
- base 64 without the 0 (number zero), O (capital o), l (lower L), I (capital i) and the symbols + and /
- Plus Version and Checksum

3. Addresses


Base58Check Encoding
Payload
1□Add Version Prefix
Version
Payload
2□Hash (Version Prefix + Payload)
SHA256
SHA256
first 4 bytes
Version
Payload
Checksum
3□Add first 4 bytes as checksum
Base 58 Encode
4□Encode in Base-58
Base58Check Encoded Payload

3. Addresses

Table 4-1. Base58Check Version Prefix and Encoded Result Examples



Type	Version prefix (hex)	Base-58 result prefix
Bitcoin Address	0x00	1
Pay-to-Script-Hash Address	0x05	3
Bitcoin Testnet Address	0x6F	m or n
Private Key WIF	0x80	5, K or L
BIP38 Encrypted Private Key	0x0142	6P
BIP32 Extended Public Key	0x0488B21E	xpub

3. Addresses

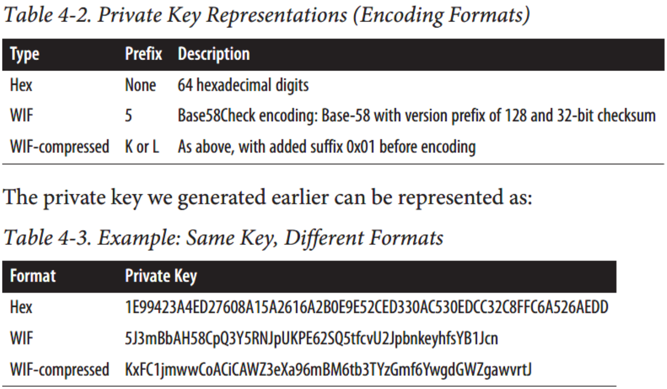


Table 4-2. Private Key Representations (Encoding Formats)

Type	Prefix	Description
Hex	None	64 hexadecimal digits
WIF	5	Base58Check encoding: Base-58 with version prefix of 128 and 32-bit checksum
WIF-compressed	K or L	As above, with added suffix 0x01 before encoding

The private key we generated earlier can be represented as:

Table 4-3. Example: Same Key, Different Formats

Format	Private Key
Hex	1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD
WIF	5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn
WIF-compressed	KxFC1jmwwCoACiCAWZ3eXa96mBM6tb3TYzGmf6YwgdGWZgawvrtJ

https://eng.paxos.com/blockchain-101-elliptic-curve-cryptography

The End