Keys Summary

A bitcoin wallet contains a collection of key pairs, each consisting of a private key and a public key.

Private Key

The private key (k) is a number, chosen randomly. In order to create a bitcoin private key you simply need to pick a number between 1 and n-1, where , almost equal to . The number n is the “order” of the elliptic curve used in bitcoin.

A private key is usually represented in hexadecimal base (64 digit).

Example of a private key: 0C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D

Order of elliptic curve used in bitcoin:

FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364140

Public Key

Now that we have a private key we need to compute the corresponding public key.

A public key (K) is a point on the elliptic curve, a couple of numbers, obtained by the multiplication in the elliptic curve fields, between k and G, where G is called the generator point.

Where is the private key, is the public key and is the generator point.

This multiplication is irreversible, in the sense that having and is possible to obtain , but if I know only and is impossible to obtain .

A point in the elliptic curve must always satisfy this relation:

Where is the prime number of the elliptic curve.

Exemple of a public key:

x=50863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B2352 y=2CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6

Considering the equation of the elliptic curve, we can easily notice that for every there are two . For this reason, we can represent a public key in two principal ways: compressed and uncompressed.

Uncompressed public key

An uncompressed public key is obtained simply concatenating the coordinate with the coordinate and adding at the beginning, for a total of 130 hexadecimal digit.

Example of a public key uncompressed:

0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B23522CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6

Compressed public key

A compressed public key is obtained simply taking the coordinate and adding 02 at the begging if the coordinate is even, 03 otherwise.

Example of a public key compressed:

0250863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B2352

Private Key WIF (Wallet Import format)

In order to easily memorise a private key, a wallet import format was design.

Add first a version number (80 for bitcoin) in front of the private key, in order to recognize quickly for what cryptocurrency that private key was used.

Then add 01 at the end of the private key if you want a WIF compressed, none if you want a WIF uncompressed.

After that, add a checksum at the end.

In order to compute the checksum, you need to apply the SHA256 function twice to the result previously obtained, take the first 4 bytes (8 hexadecimal digits) and put them at the end.

Now you need to compute the Base58Encode, obtaining a 52 digit string.

Example of private key WIF: KwdMAjGmerYanjeui5SHS7JkmpZvVipYvB2LJGU1ZxJwYvP98617

Address

Now you need an address in order to have the possibility to receive money.

Compute the HASH160 on the (compressed) public key. This is an irreversible procedure, so you cannot obtain the public key from the public key hash.

In order to obtain the base58 format, you need to add a Version in front of the public key hash, add the usual checksum at the end and then encode everything with Base58Encode, obtaining a 34 digit string.

Example of an Address:

1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2