Common Address Encoding Formats

Author	josh
Date	2019-10-06 16:30:31

Overview

When sending money to someone else using Bitcoin, Bitcoin Cash, Ethereum, or another cryptocurrency, you send funds to the other user's *address*. This unique identifier for the other user's wallet may look like a "random" string of letters and numbers, like this: `13GuDW2Km8TR6iCYP8E5QGhNky2ne7T17r`. (Note: this is just a random address; don't use it!). But there's actually quite a bit more going on behind the scenes when it comes to address encoding.

Different cryptocurrencies use different schemes to turn raw address data into what you would actually copy and paste, type, or scan into a wallet for sending. In some cases, one cryptocurrency may support multiple encoding formats. Let's take a look at some of the most common formats used by major blockchains.

Common Encoding Formats

The OG Encoding format: base58check

The first major encoding format to appear is used most commonly in Bitcoin and Litecoin, the first major derivative of Bitcoin. This encoding format is known as *base58check*, and addresses look like this:

```
13GuDW2Km8TR6iCYP8E5QGhNky2ne7T17r
```

Base58check encoded addresses are generally derived using the same process (at least in the case of Bitcoin & Litecoin). First, the raw address is derived using a two-step cryptographic hash - first SHA-256, and then RIPEMD160. This gives us a 160 bit (20 byte) "pay to public key hash" address.

To encode the address, a *version byte* is added to the front of the raw hash. For Bitcoin, this version byte (in hexadecimal format) is `0x0`, and Litecoin uses `0x3`. Next, a *checksum* is generated, in order to help with error detection. The address including the version byte is hashed using SHA-256 twice. The first 32 bits (4 bytes) of the resulting hash is added to the end of the raw address.

Finally, the raw data is converted to base58. Base58 is a number system, just like what we're used to with base 10. But instead of digits 0-9, base58 uses this alphabet: `123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz` . It's overall quite similar to the popular base64 encoding, but omits certain characters that are difficult to distinguish when writing or reading. There's no 0, uppercase O, lowercase l, uppercase I, or non alphanumeric characters.

Base32-based: CashAddr and bech32

Over the last few years, base32 based encoding schemes have become more popular to deal with some of the issues base58check addresses are known to have. base58check addresses are shorter, but the mixing of uppercase and lowercase letters can make critical address data ambiguous and hard to read and write. Base32-based schemes solve this problem by only using non-ambiguous lowercase letters and numbers.

The first major currency to commonly use a base32-based system is Bitcoin Cash. BCH uses a special system called *CashAddr* for distinguishing BTC and BCH addresses. Cashaddr, like base58check, prepends a *version byte* to the public key hash. Version bytes for CashAddr can vary, as defined by the specification. The checksum uses a special algorithm to generate an error-detecting code called a *BCH (Bose–Chaudhuri–Hocquenghem)* code. The BCH code is the last 40 bits (5 bytes) of the final address. The address often includes a prefix to indicate the blockchain it's used on as well, such as `bitcoincash`. A CashAddr address looks like this:

```
bitcoincash:qqv0y9qvkwxcyjdwcm9f5zgelf8aapndhuc2u24x9n
```

Bitcoin (BTC) has introduced a similar system for segwit addresses only, called bech32. All of these addresses, when encoded, start with `bc` for mainnet addresses, and also end with a BCH code for error detection.

Hexadecimal

The final encoding type we'll discuss is the one used most commonly in Ethereum: hexadecimal. Hexadecimal encoding (often referred to as "hex") is used very frequently in computer science outside of cryptocurrency. The hex number system is *base16* and uses numbers and a few letters for its alphabet: `0123456789abcdef`.

Ethereum address derivation and encoding is quite simple compared to other common cryptocurrencies. There's no version byte, and no checksum (although a checksumming system has been introduced more recently, it's not covered here). The public key hash is simply encoded as hex, and `0x` is added to the front. This is a common indicator for hex format outside of cryptocurrency. An ethereum address looks like this:

```
0xc257274276a4e539741ca11b590b9447b26a8051
```

The advantage to base16 is simplicity. There's only 16 characters, all very easy to distinguish from one another! That makes it a more hardy format for reading and writing. However, it ends up creating rather lengthy addresses - for every 1 byte of data, you end up with two characters in hex.

Encoding - Because People Aren't Computers

These encoding systems all exist for one simple reason - us meatbags aren't particularly adept at reading and writing raw binary data. So instead of long chains of binary data, we send each other *encoded* addresses. Each of these common formats has its pros and cons when it comes to length,

ease of use, and error detection. But in general, all of them are designed to make it easier for us to transact with these currencies without having to deal with raw binary data - thank goodness!