Fun Crypto-Crypto Facts – The Cryptography of Cryptocurrencies

Author	josh
Date	2024-11-17

Overview

Cryptocurrencies such as Bitcoin are named as such because these systems are build on *cryptography*. These fascinating systems of applied cryptography make use of several primitive functions, from hashes to elliptic curves. These systems are built to secure value, which makes the stakes even higher than the typical security and privacy use cases. There's a lot of deep and complex topics to unpack, but today we'll focus on some fun, high level tidbits of interest. Let's dive into several fun and fascinating crypto-crypto facts!

Crypto-Crypto Facts

Astronomical Keys

Bitcoin private keys are 256 bit ECDSA keys, specifically for the secp256k1 curve. These elliptic-curve values are used to derive public keys for receiving coins, and for signing transactions that send value to another party. Sometimes, people wonder if it is possible to brute-force or guess Bitcoin private keys to steal or recover lost coins. However, the math of why you cannot do so is pretty astounding.

The available 256 bit keys are slightly less than 2^{256}, which is 1.15×10^{77} possible keys. It turns out that this is around the same order as the number of atoms in the observable universe, which some estimates put at about 10^{82} atoms. That's so many keys, that even if we had a theoretically super-fast computer crunching the numbers, we don't have enough energy in the solar system to flip all those bits. If a key is sufficiently random, it is impossible to guess by brute-force. There's simply not enough resources to do so.

Hash Function Sources

Bitcoin uses two hash functions for generating addresses. The first is SHA-256, which gives a 256 bit output. This algorithm was designed by the United States National Security Agency, well known for its high level of cryptographic knowledge and controversial intelligence gathering tactics. The next algorithm, RIPEMD160, gives a 160 bit output resulting in a shorter overall address. This algorithm was designed by an academic research group called the Computer Security and Industrial Cryptography research group (COSIC).

There are some technical reasons for the choice of RIPEMD160. But some believe that the choice to combine SHA-256 and RIPEMD160 is due in part to their vastly different sources. If there's some hidden vulnerability or issue with SHA-256 due to its development by the intelligence community, it's less likely that RIPEMD160 has the same vulnerability. If

RIPEMD160 suffers from some issue not seen by the open academic community, it's possible the IC-focused developers at NSA accounted for this in the design of SHA-256.

Addresses Aren't Raw Pubkeys

When deriving a public key, we go from the private key data which much be kept secret to a public value that is shareable. In the case of Bitcoin, this public key is used to receive coins, and so that other on the network can validate our signed transactions. But a Bitcoin address isn't just a raw public key. In fact, it's a public key *hash*. The process of deriving a BTC address starts with the private key, used to derive the public key. That public key is then hashed using SHA-256, and again with RIPEMD160 before the final encoding.

This extra hash layer allows us to use a shorter address, and provides a slight additional layer of security. If you've only received with a Bitcoin address (public key hash), nobody knows the public key until you spend those coins. The public key must be revealed on spend so that others can validate your digital signatures, but it isn't revealed if you've only received at that particular address. In the event some serious vulnerability is found in the elliptic curve cryptography code for a particular wallet, this could theoretically save someone from theft if they never use addresses. An example of such a vulnerability is wallets that generate poor-entropy or reused k values for digital signatures, which can leak the private key.

Mining Probabilities

Bitcoin mining uses SHA-256, and a whole lot of computing power aimed at what is essentially a guessing game. In mining, the proposed *block* containing transactions batched together is combined with a random value called a *nonce.* Those values are hashed using SHA-256 to get a candidate *block hash*. The network has a specific way of deciding which miner wins the race with a special value called the *difficulty target.*

This is essentially another 256 bit number, and the winner of the race is the first miner to find a nonce such that the hash of the block + nonce is less than the difficulty target, numerically. Another way to think of it when visualizing these numbers in hexadecimal is to think of the number of leading zeros the block hash must have. It turns out that the smaller the difficulty target, the lower the probability that anyone nonce will result in the desired hash output. This difficulty/probability is what allows the network to adjust the mining of blocks to the overall computing power on the network. The more miners, the higher the difficulty – so that it's probable a block will be found every 10 minutes or so.

Unlimited Addresses from One Key

Ever wonder how a seed phrase of 12-24 words can be used to back up an entire wallet's contents? For example, one seed can be used to store essentially unlimited addresses for multiple currencies like Bitcoin, Ethereum, Litecoin, and more. How is it possible to store all this information within only 128 to 256 bits of seed phrase data? This is due to the magic of HMACs – hash based message authentication codes.

Using this singular chunk of seed data, we can derive a nearly unlimited amount of addresses just by adjusting the input to the HMAC. HMACs take a message and secret key, so the seed + index is used for the message and a specially computed value called a chain code is used as the secret

key. The internal details are complex; but the idea is simple – we can take one seed value and used a specific *deterministic* formula to get the same keys every time. So as long as you have the seed and follow that specified formula, you'll get the same keys every time. Even if your wallet device is destroyed, you can restore all of your keys and addresses with that seed. This makes backing up a Bitcoin wallet much easier than the old days, where you'd need to periodically update you backup with new randomly-generated keys.

Fun Facts, Brief and Stacked

Each one of these facts shared in this article could fill their own articles in depth, so there's a lot more to explore here for those interested. For this article, I wanted to share some of these interesting facts more briefly and in one place. In my opinion, cryptocurrencies are some of the most fascinating applications of applied cryptography – combining several neat technologies as a way to openly exchange value. These primitives also have many use cases outside of cryptocurrencies, from securing our data to keeping the web safe. Well worth studying!