Blockchain: Simple Explanation

Oleg Mazonka, *2016*

***Abstract*—This paper presents a step by step introduction to what blockchain is and how it works.**

***Index Terms*—Blockchain, Hashchain, Bitcoin, Cryptocurrency.**

I. Introduction

Blockchain is a buzzword. You have probably heard it so many times you feel like you should undersatand what it is. By the end of reading this article you just might.

The original problem comes from the idea to have something represented as digital entity that can be passed securely from one party to another. Imagine you have some money in your bank account, and you would like to securely transfer some of it to another person. This ususally goes through a digital transaction – no physical money is passed around. This is nice and simple. But, it requires trusting the bank. Is it possible for some digital entity stored on my computer, representing money, to be passed securely to someone's else computer, without a bank? Yes it is possible. However, regardless of representation any digital sequence can be copied. This creates the well-known problem of "double-spending": one person makes an electronic transaction more than once using the same money. This problem was not solved properly until the first blockchain cryptocurrency, Bitcoin, appeared in 2009.

Many other cryptocurrencies appeared in the following years and followed the same principle of blockchain technology, which solves the double-spending problem without a bank. The digital sequences, which are linked to a person, are passed around without possibility to copy and double-spend by placing one hashchain inside another hashchain. What is a hashchain? Let us first recall what is hash.

II. Hash Functions

Hash function is a method to convert data of arbitrary size into a digital string of predefined fixed length, called hash. One of the simplest hash functions is the modulo operation. Any digital string can be converted into a number (possibly very big); this number can be divided by a constant; and the remainder of that division is the result, hash. Obviously the result is less than the constant, because its size is no greater than the size of the constant. This hash function is simple but at the same time not used too often because people want to have another property – one way computation. It should be easy to compute the hash, but finding any input to the hash function must be difficult, or better virtually impossible.

Hash functions with such a property are sometimes called cryptographic hash functions, if necessary to distinguish from others. Such functions digest the bits of the input data in a very convoluted way to make the reversible computation impossible. The best known cryptographic hash functions are MD5, SHA1, SHA2. An example of MD5 is this

$$\mathrm{MD5}(\text{"}\texttt{abc}\text{"}) = \texttt{900150983cd24fb0d6963f7d28e17f72}$$

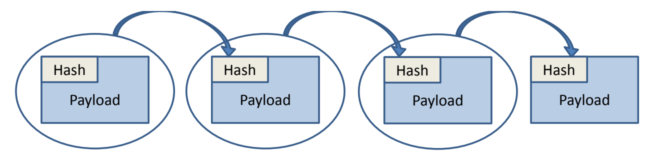
The result is a 128-bit string shown here in the hexadecimal format.

If the hash value, represented in some form, is fed again into the hash function, a new hash is obtained. If this process repeated and the results are combined into a sequence of hashes, one obtains a hashchain.

III. Hashchains

Hashchain is a sequence homogeneous data chunks, or simply blocks, linked together by a hash function. Figure 1 schematically shows a simple hashchain.

Figure 1



Each data block consists of the hash and the payload. The hash of each block is calculated out of the whole previous block. Payload in each block is arbitrary data. This hashchain has the important property: no data can be modified at any block without affecting the integrity of the subsequent blocks. For example, if the payload of the first block is changed, then the hash of the second block must be changed as well, and hence the hash of the third, and so on.

Next step is to authorise only one person to create a new block in the chain. One way to do it is Public Key Cryptography (PKC).

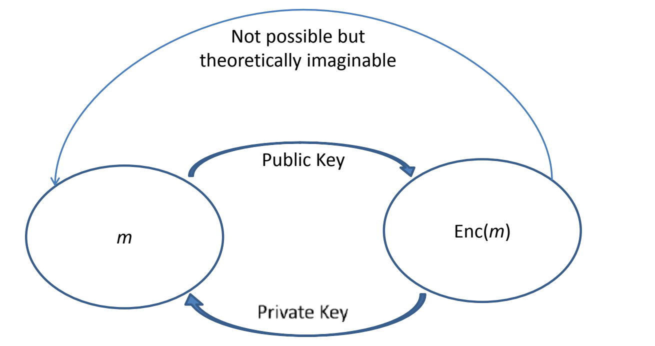
IV. Public Key Cryptography

The basic idea of PKC is similar to hash functions – one way computation. Given some data m (m for message) anyone can compute encrypted value Enc(m). But, only the one who knows a special key related to this encryption can compute in the opposite way, i.e. find m from Enc(m). The latter process is called decryption.

To achieve PKC, one must first create a so-called pair of keys: public and private. The public key is used to encrypt data and can let be known to anyone. The private key is used to decrypt and must be kept in secret.

Figure 2 shows schematically the arrows of computation. The top arrow represents the theoretical possibility to decrypt without knowing the private key upfront. For example, RSA encryption is based on mathematical property of big number factorization.

Figure 2



If someone finds a way to factorize a big number (which is the public key), then that person would be able to decrypt without knowing the private key, in fact breaking the cryptography.

There are two main scenarios where PKC works. The first is when one party wants to send a secure message to another. In this case the first party encrypts the message by the public key of the second party. The encrypted message can be made public because only the second party is able to read the message encrypted with their key. The only vulnerability of this method (apart from the theoretical one by cracking the math) is knowing and trusting the public key in the first place. To attack this problem a whole system of hierarchical public certificates exists, for example, in the Internet https protocol.

The second scenario is quite opposite to the first. Given some data, *Alice* encrypts it using its private key (or encrypts only hash from the original data) and then publishes both data: the original and encrypted. Anyone knowing the public key can verify that encrypted data is actually computed using the private key that is paired with the public key used for decryption. This verification works as a validation Alice actually did that. That process is called digital signing and the encrypted part is called *digital signature*.

Figure 2 shows the first scenario. In the second scenario the private and public keys are swapped. Let us follow this again. Alice takes a hash from a document m and encrypts this hash using his private key, then makes both the document and the encrypted hash public. Anyone else can take the hash of the document and decrypt the encrypted hash using the public key of Alice. If these two values are the same, then it means that Alice has indeed signed the document m. This scenario with digital signatures is the one which is useful in hashchains.

V. Securing Hashchain with Public Key Cryptography

Now, let's go back to the question of whether it is possible that only one person is enabled to add a block to the hashchain. If the last block (Figure 1) contains the public key of the current owner and the digital signature of the previous block owner, then creation of the next block will require the signature of the current block, i.e. of the person who has the private key. When creating a new block the current owner places another public key in the next block and signs that block. So the next block will be owned by whoever keeps the private key corresponding to the newly published block. And the signature proves that only the previous owner could have done that.

Figure 3 (from the original Bitcoin paper of Satoshi Nakamoto)

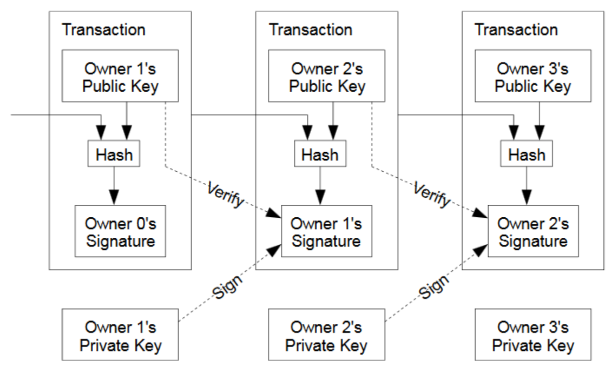


Figure 3 shows how PKC can be used in hashchains.

Hashchain with PKC authorisation can perform as a function for secure transfer of digital objects, called *tokens*. Suppose that one hashchain represents a token. It may be something to which the real world assigns a value. The owner of the last block is the owner of this value because only he is able to pass it to somebody else. However this kind of hashchain implemented as-is would require a central place where the blocks are actually created, stored, and can be verified. We come back to the concept of a bank – a place which is required to be trusted.

VI. Blockchains

Imagine that there is a place (possibly in virtual reality) where a continuous process exists that constantly creates new blocks for some global hashchain. And imagine that you can place any data you like into the payload of the blocks of that global hashchain. If such a place and process would exist, then you can insert the blocks of a hashchain representing values into the blocks of this global hashchain. In other words that would be one hashchain inside another hashchain. Essentially this mechanism is called *blockchain* technology.

Figure 4

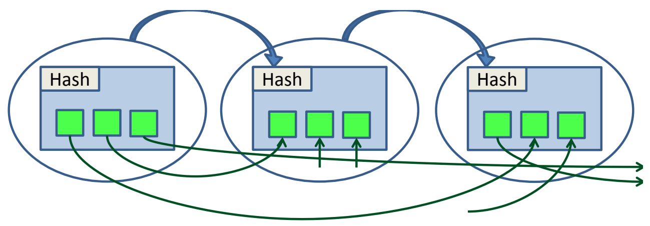


Figure 4 shows an external hashchain where block data consists of blocks of internal hashchains. The blocks of the internal hashchains can chaotically appear in the external hashchain. The obvious condition is that older internal blocks cannot appear in newer external blocks.

Different solutions exist on how to organise that global hashchain. And one particular solution proposed in the original Bitcoin paper made Bitcoin popular.

VII. Social Incentive

In Bitcoin network the global hashchain is a database distributed among many computers. This global hashchain consists of many internal hashchains representing some valuable tokens called *bitcoins*. New blocks created on those internal hashchains are called *transactions*, because they represent changing ownership of bitcoins. The owner of a particular bitcoin creates a new block on the corresponding internal hashchain and publishes this block as a new transaction to be included into the next block of the external hashchain. Publish means sending to all other known participants of P2P Bitcoin network. When a new block of the external hashchain is created it is published in the same way.

Any participant can create a new block for the external hashchain including published transactions into the block. But this is not easy. To avoid chaos all participants follow a particular set of rules – the protocol. And whoever does not play by these rules is ignored. The protocol is based on three principles:

1) creating a new block requires significant computational effort;
2) creating a new block is rewarding, so many would make en effort to successfully create a new block, and
3) when branching occurs, the longest branch wins.

The first one simply requires that the hash taken from the block plus some random data, starts with a number of zero bits. So this random data is modified and the hash is calculated until the condition of zero bits is achieved. That requirement makes block creation difficult.

On the other hand whoever creates a block receives a number of newly created internal hashchains representing some value, bitcoins. That also is part of the protocol. Moreover since the bitcoins represent a kind of currency and internal hashchains represent transactions with that currency, the transaction owners can give a small portion of the transaction value (fees) to the creator of the new block of the global hashchain. So the creator of the new block receives the newly created bitcoins plus all fees collected from the transactions placed into the block.

The third rule ensures that there is only one current valid copy of the database distributed among many computers. It is obvious that since many parties try to create blocks, some different blocks are created independently. Since the propagation is not instant, new blocks may be created on top of those independently created blocks. That makes the global hashchain branch. The third rule makes sure that only one branch which is the longest is considered to be valid. This rule works fine because block creation is difficult, hence the probability of two or more branches surviving die out very quickly.

We can see that the external hashchain plays the role of the central repository effectively replacing an authority entity. Since the external hashchain is global, it stores bits and pieces of internal hashchains uniquely and reliably binding the tokens with their owners, at the same time making double-spending impossible. The database of the external hashchain is distributed among many not trusting each other parties, so it makes it more difficult to enforce the whole community of those parties to follow some externally applied regulations such as political laws.

VIII. So What is Blockchain?

Blockchain is a particular organisation of hashchains inside another hashchain. The external hashchain has to be based on a set or rules that make balance between usability, simplicity, incentive, and trust and authority. Its purpose is to replace the central access point that can be controlled and possibly abused by a human. Internal hashchains are required to have authorisation mechanism so one party can own something of particular value. The authorisation mechanism may not necessarily be PKC. For example, Hasq hashchain is a much simpler hashchain using only hash functions as authorisation.