UM–SJTU Joint Institute

Topic 2: The Random Oracle Model

VP475 Project 2

Project Group 12

Group Members:

Fang Han	518370910009
Shen Heyin	518370910049
Dong Yier	518370910189
Li Anna	518370910048

Date: July 30, 2021

Abstract

Bitcoin is a digital currency. After a decade of development, it has had a major impact on the world economy. In this report, we illustrated the composition of a Bitcoin system, including the blockchain and the wallet. We also described in detail about the transaction and mining procedure involved in the system which is based on advanced cryptography and mathematics. For the wallet part, we introduce a specific kind of wallet-HD wallet. Then we are able to analyze its complexity and security based on the Elliptic Curve Digital Signature Algorithm(ECDSA) itself and the construction of Bitcoin model. This report serves as a literature review and gives a comprehensive and clear introduction of the Bitcoin system.

Keywords: Bitcoin, Blockchain, Mining, Cryptocurrency Wallet

Contents

1 Introduction

Bitcoin is the first decentralized network and digital cryptocurrency in the world. It uses a peer-to-peer (p2p) system to verify its legitimacy based on cryptography, and processes and spreads transactions all around the network. In other words, the online payments are sent from one party to another directly without going through a third authority institution. This setup not only improves the efficiency of transactions but can also increases the security, since without the participation of the third party, we don't need to worry about the information disclosure, and attackers can no longer destroy the system by just attacking the central bank. [4] In this system, bitcoins are circulated through a software or a service provider.

1.1 Components

A Bitcoin network is mainly composed of three parts: the blockchain, the mining, and the wallet.

Blockchain As a p2p system, Bitcoin must ensure its security and authenticity. One famous concern is the double-spending problem. For example, in the context of computers, when Alice sends $10 to Bob, she only sends a digital money file, which can be done quickly through e-mail. However, we know that the file sent is just a duplicate, and Alice still keeps the original one. Therefore, it's plausible to send again the money file to another one. [9] This problem was solved by the blockchain, a public ledger where all the transactions are constantly tracked and recorded, similar as an account book. Then, for every new transaction to happen, it must first pass the check via the blockchain to make sure the bitcoins to be sent are effective, thus solving the double-spending problem.

Mining "Mining" refers to the procedure where "miners" try to pack the records of all the transactions within a certain time period and integrate it to the blockchain by solving a complex cryptographic problem. As running the protocol requires a great amount of efforts and substantial computer power to determine whether a posted transactions are valid, the miners will be paid as a reward for their effort and the employment of their resources. However, only the first successful participant will get the reward of the newly emitted bitcoins. Therefore, the mining process can be seen as a zero sum game.

Wallet The wallet can generate an unique Bitcoin address composed of a sequence of letters and numbers for the user. The address functions similar as a bank account number with which the user can receive payment. The bitcoins can be bought at a Bitcoin exchange, a vending machine or through purchases of things. [2]

1.2 Procedure

In a bitcoin network, the basic operating mode is as follows:

1. New transactions created, and be broadcast to the entire network for each user to know.
2. Miners try to pack the transactions by solving a complex problem.
3. Someone figures out a solution.
4. All users verify the result, since the blockchain is publicly accessible, all the participants can verify by themselves the validity. If the result is correct, a new record was created and stored, and new round begins; while if the result is wrong, the solving procedure continues.

In this report, we will discuss in detail about the procedure of the Bitcoin system and how it works based on advanced cryptography and mathematics.

2 Blockchain

The blockchain is a public decentralized ledger of all the transactions across a peer-to-peer network and can be considered as a chain of blocks linked together through cryptography. All the nodes collectively follow a protocol to communicate and validate new blocks. [14]

Bitcoin is the first system that introduced this technology of blockchain, making it the first digital currency to solve the double-spending problem without an intermediary, and reversely, blockchain becomes the foundation of the existence of bitcoin, and anyone can establish a server and enter the blockchain network to become one node in it. [13] This blockchain is maintained by a network of communicating nodes who run the bitcoin software. As mentioned previously, all the transactions such as ”A sends B bitcoins to C” will be recorded and broadcast all around this network using readily available software applications. Therefore, for each network node in the system, they store their own copy of the blockchain, thus all the users know all the transactions. In this way, the traditional information asymmetries in systems involving a third party can be eliminated and Bitcoin achieves independent verification of the chain of ownership.

2.1 Structure[3]

Each block is composed of two main parts what we call the head and the body, shown in 1. The head contains the version of the block (4 bytes), the hash of the previous block (32 bytes) up to the genesis block, the hash of the Merkle Root (32 bytes), the timestamp (4 bytes), the difficulty bits (4 bytes in compressed format, 256-bit in practice), or the so-called target, and the nonce (4 bytes) for ”mining”.

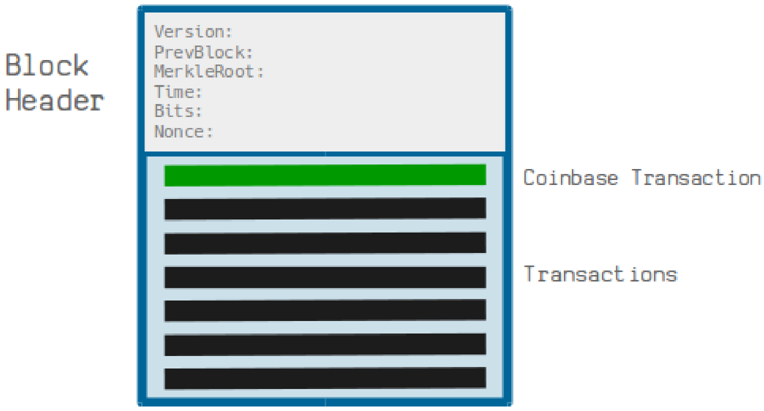


Figure 1. Structure of a block

Version Sometimes separate blocks can be produced concurrently, creating a temporary fork. In addition to a secure hash-based history, any blockchain has a specified algorithm for scoring different versions of the history so that one with a higher score can be selected over others.

PrevBlock The length of the head is 80 bytes and the hash of a block is obtained by taking twice the SHA256 algorithm on the head data.

Merkle tree The Merkle tree is a hash binary tree used to summarize all the transactions in a block, generate the digital signature of the entire transaction set and provide an efficient way to verify whether a transaction exists in the block.

Timestamp The timestamp is responsible for showing the existence of a certain transaction data when the block was published to get into its hash. Since each block contains the information of the previous one, they form a chain with each block reinforcing the ones before it. In this way, blockchains can reject the illegal tampering of the data for once the data is recorded, they cannot be changed retroactively without altering all subsequent blocks.

Target & Nonce The target is the difficulty level for mining, a measure of the amount of computation to find a feasible value to solve the cryptography, and the resulting value will be the nonce. The value of the target will be adjusted so that at varying intervals of time with an average time of 10 minutes, a block containing a new group of accepted transactions is created and added to the blockchain.

2.2 Numerical Settings

Now we can take a look on the basic setups of the bitcoin protocols: [12]

1. Miners who find new blocks will be rewarded, in 2008 with 50 bitcoins, then halved every 4 years, in 2018 with 12.5 bitcoins.
2. 10 minutes to generate a new block on average.

3. The total amount of bitcoin is 21 million, calculated based on the previous two settings:

$$50 \times 6 \times 24 \times 365 \times 4 \times (1 + \frac{1}{2} + \frac{1}{2^2} + \dots) = 21,000,000$$

4. Transactions written into the body of a block are prioritized according to their transaction fee.
5. The block size is only 1MB and a transaction is around 500 bytes, so a block can only contain more than 2,000 transactions at most.

2.3 Features

Bitcoins combine a high degree of identity protection with a decentralized system of verifying transactions. Most of the bitcoins employ a double-key cryptography, which is based on a hashing algorithm that protects the privacy of the payers and the payees. [9] This will be further discussed in the following sections.

Decentralized Since the network of Bitcoin does not have a centralized physical node or a management agency, Bitcoin distributes this responsibility across the network and delegates the verification of transactions to every network node. The maintenance of the network functions relies on all the nodes that with maintenance functions, and the status of each node is equal. The damage of one node or even several nodes will not affect the operation of the whole system, so the network has strong robustness. [14]

Disintermediation The data transmission between nodes is anonymous and the nodes do not need to trust each other. The whole system is run by open and transparent mathematical algorithms. The data are open among all the nodes, so there is no way to deceive others.

Security Each node in the system has a complete copy of the entire ledger. Therefore, unless one can control over 51% nodes, the modification of the ledger based on a single node is noneffective and can not affect the data on other nodes. [2] What's more, though bitcoin transactions are publicly accessible, the participants are recorded by their alias in the blockchain, thus protecting their privacy.

In all, Bitcoin is developed upon the invention of the blockchain and is, ideally, equitable, independent, and transparent using the distributed network of users who will contribute computer power in exchange of new bitcoins. The blockchain is considered a type of payment rail and are widely used by cryptocurrencies, but it also has a lot more to offer to beyond digital payments, and has already inspired other applications. [13]

3 Transactions and Mining

In this section, we will introduce the transaction and mining of the bitcoin. Basically, mining is based on the process of transaction. Thus, we would first introduce the process of bitcoin transaction.

3.1 Transaction

The bitcoin transaction is the process that the custody of the bitcoin is transferred from one place to another. The transaction process consists of a version number, a locktime value, an input list and an output list.[5] First constructed by Nakamoto[8], the model of the transaction part is shown in Figure 2.

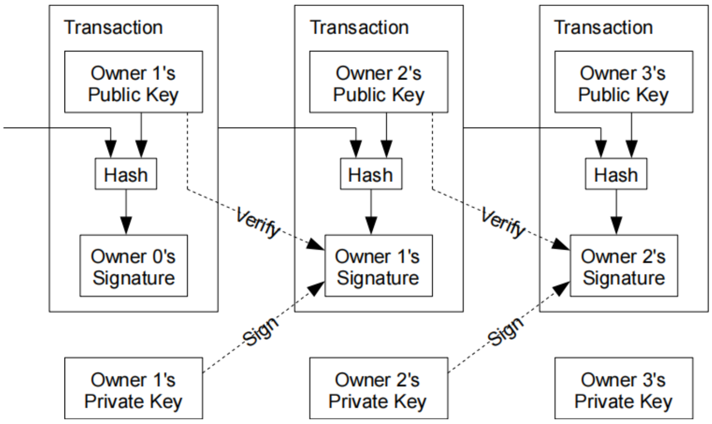


Figure 2. Model of the bitcoin transaction

3.1.1 Transaction Inputs

Namely speaking, inputs are the funds that will be spent in the transaction process. Similar to the trade in the real world, we need to pay the money and the input plays the role of the money in the transaction process. However, different from the cash, the bitcoin transaction input is composed by the following items:[4]

- Previous tx
- Index
- ScriptSig

The previous tx records the information of a previous transaction. The index is the specific output of the transaction. As for the scriptSig, it is mainly composed by two parts, a signature and a public key of the owner, who is the provider of the inputs. The public key is a verification for the signature provided by the owner.

3.1.2 Transaction Outputs

The transaction outputs are composed by value and scriptPubKey. The value represents the number of bitcoins in the transaction process. The scriptPubKey can be interpreted as the money transfer in the transaction process. However, due to the property of the bitcoin, the outputs can be not unique. For example, if Alice want to pay 50 bitcoins to Bob, but the last amount of bitcoins she received in the last transaction is 100, then at this moment, the 100 bitcoins were packaged into one scriptPubKey and will be sent totally to the system. The system will break the package into two part, while 50 bitcoin package was sent to Bob as scriptPubKey and the rest 50 package was sent back to Alice. This makes the outputs in one transaction process not unique.[4]

3.1.3 Key Problem of the Transactions

In [8], as Nakamoto mentions, the key problem of the bitcoin is how to prevent a coin from one owner being double-spent. As shown in Figure 2, there is no procedure to cancel the original scriptPubKey in the transaction process. In real trade, we can rely on an authoritative third party, which checks every transaction to prevent double-spent. However, this means that the fate of bitcoin, or more exactly, negotiability of bitcoin will be controlled by the third party. Thus, in [8], Nakamoto proposes another method to prevent double-spent situations.

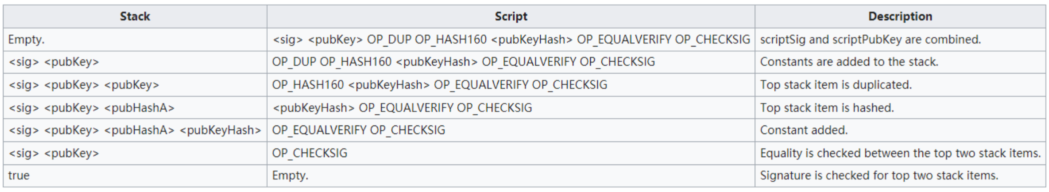
In [8], Nakamoto first proposes the concept of the timestamp. The timestamp server is a server that "take a block of items to be timestamped and widely publish the hash"[8]. Generally speaking, the timestamp proves that the data, bitcoin, must have existed in the address, for which we received the result of its hash, during the time shown in the timestamp. Each timestamp includes the information of its previous timestamp, which forms a chain.

In [8], Nakamoto also mentions the concept of proof-of-work. The proof-of-work is a necessary part in the verification part. While a request of transaction enters the system, the data that transaction involves will be scanned and verified by a single hash. If the work that needs to be done is equal to the value we sent in the transaction part, then the system will return true for the transaction.

3.1.4 Type of Transactions

As shown in Figure 2, the verification process is the necessary part of the bitcoin transaction. Basically, bitcoin has two methods of verification, Pay-to-Pubkey Hash(P2PKH) and Pay-to-Script Hash(P2SH).[4]

The scriptPubKey the user gets is not a full key. As mentioned in the inputs part, the user needs to send both the signature and public key. Then the script verifies the provided public key and verifies the signature by the public key. The checking process of the P2PKH is shown in Figure 3:[4]

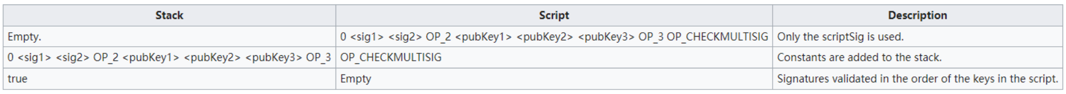


Stack	Script	Description
Empty.	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig and scriptPubKey are combined.
<sig> <pubKey>	OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Constants are added to the stack.
<sig> <pubKey> <pubKey>	OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Top stack item is duplicated.
<sig> <pubKey> <pubHashA>	<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Top stack item is hashed.
<sig> <pubKey> <pubHashA> <pubKeyHash>	OP_EQUALVERIFY OP_CHECKSIG	Constant added.
<sig> <pubKey>	OP_CHECKSIG	Equality is checked between the top two stack items.
true	Empty.	Signature is checked for top two stack items.

Figure 3. Checking process of P2PKH

where scriptPubKey contains OP_DUP, OP_DUP, pubKeyHash, OP_EQUALVERIFY, OP_CHECKSIG and scriptSig contains sig and pubKey.

Different from the P2PKH, the P2SH needs to implement fewer operations and can be applied to complex situations. The process of the P2SH is shown in Figure 4:[4]



Stack	Script	Description
Empty.	0 <sig1> <sig2> OP_2 <pubKey1> <pubKey2> <pubKey3> OP_3 OP_CHECKMULTISIG	Only the scriptSig is used.
0 <sig1> <sig2> OP_2 <pubKey1> <pubKey2> <pubKey3> OP_3	OP_CHECKMULTISIG	Constants are added to the stack.
true	Empty	Signatures validated in the order of the keys in the script.

Figure 4. Checking process of P2SH

where scriptPubKey contains OP_HASH160, OP_EQUAL, scriptHash and scriptSig contains signature and serialized script.

3.1.5 Commission charge

There is a little detail in the transaction process. In the real trade, every currency transfer in the electronic payment system will require commission charge. In the bitcoin world, since the transaction process needs miner to work, it also needs commission charge.

Not same as the commission charge in the real world, the commission charge in the bitcoin world does not have such mechanism that regulates the commission charge for every transaction. This is to say that, in every transaction you want to make, if you want your transaction to be prior than the others, you can pay more commission charge fee to the miner.

By the way, the commission charge won't appear in the output part.

3.2 Mining

Mining is a kind of specific process of transaction. When the nodes in the bitcoin network assemble new broadcast transactions into a block, there will be a reward for the user who execute the work and this is the concept of mining.[5]

When a node finds a valid proof-of-work for a block, it will broadcasts the block to all other nodes. Other nodes will accept the block if and only if all the transactions of the block are valid and the transactions have not been included by the previous transaction.[5] Then with the new block being accepted, it will form a block chain, which is mentioned in section 2.

It seems that if we have some strong methods to calculate, we can mine new blocks more and more. However, the total number of the bitcoins is bounded. The number is 21 million. With more blocks to be discovered, the reward of discovering a single block will be deducted half and half. This makes the bitcoin never suffer from the inflation problem, which is a serious problem for all the currency published by a center.

3.3 Nonce

As previously mentioned in section 2, the hash value of each block is calculated by applying twice SHA256 hash algorithm on the data string stored in head. Denote as

$$Hash = SHA256(SHA256(string))$$

Recall that the string is composed of several determined substrings (such as the hash of the previous block and the target), and the random number nonce. Once the input string is determined, the hash value can be quickly calculated and we obtained a 256-bit number. However, this number $Hash$ must satisfy the condition that

$$Hash < Target$$

. As a 256-bit number, the target usually have the first n significant bits equal to zero. Therefore, to find a hash value smaller than the target value is the same as to find a hash value with the first n significant bits equal to zero. While all the other parts of the string is fixed, the only thing that can be changed is the nonce. Since there is no efficient method to calculate the inverse of the hash function, determining the correct nonce requires a great amount of trial-and-error. [10]

In practice, a miner

1. guesses an initial nonce
2. appends it to the hash of the current header
3. rehashes the value
4. compares this to the target hash.

If the resulting hash value meets the requirements, we say the miner finds a solution and create a new block. Then the miner can get the reward.

3.3.1 Difficulty

The target value can determine the difficulty level of the hash problem. When the target has n zeros at the beginning:

$$Target = \underbrace{0\dots0}_{n}\underbrace{1\dots}_{256-n}$$

Since the probability for zero to appear at each position is $\frac{1}{2}$, then the probability that we find a $Hash$ is

$$\frac{1}{2} \times \cdots \times \frac{1}{2} = \frac{1}{2^n}$$

Therefore, the number of computation needs to perform is 2^n theoretically.

Though the transaction for each participant is different, resulting in a different complexity, we can still notice that it is almost impossible for a miner to find the right nonce on first guess. In other words, we can roughly ensure that every miner should try massive nonce options. The smaller the target value is, the harder it is to find a solution.

The block difficulty is the same among the entire network so that all miners may find a correct value. [10] Since the bitcoin network wants to keep an average time period of the creation of new blocks to be 10 minutes, the target number will be adjusted to control the difficulty level and thus control the average time to meet the requirement. If the number of blocks processed is less than required, the difficulty will be reduced, and the reverse situation is the similar.

4 Cryptocurrency Wallets

4.1 General Overview

In the previous part, we discussed about have to trade the bitcoin and how to mine bitcoin by proof-of-work. And we discussed about the usage of public cryptography system in bitcoin. However, there are still some essential problems of bitcoin system, which are where to store the private key, where to store the digital Bitcoin and how to validate the transactions. Therefore, we need a ”wallet” to satisfy our needs, which is called cryptocurrency wallet. By using the cryptocurrency wallet, we can prevent any other attackers from stealing our bitcoin or altering the transaction.

When a transaction take place, namely, someone exchange the value of his bitcoin for something else with another bitcoin wallet. After the transaction, every individual wallet will receive the information and use its secret key to sign and validate transactions. By doing this, people can prove that the payer and the seller are the owner of their wallet.

Bitcoin wallets contain only keys, not coins. Each user has a wallet that contains multiple keys. Wallets are key chains that contain only private/public key pairs. Users use the keys to sign transactions, thus proving that they own the bitcoins. In the Bitcoin network, ownership of bitcoins is determined by a digital key, a bitcoin address and a digital signature.

4.2 Classification

Cryptocurrency Wallets are generally classified into two kinds: non-deterministic wallet and deterministic wallet. The main difference between them is whether the multiple keys they contain are related to each other.

4.2.1 Non-deterministic wallet

For non-deterministic wallet, every private key is generated randomly. This wallet is also called "Just a bunch of keys", which is often simplified as "JBOK".

Since every private key is generated randomly, it costs a lot of space and time for you to store and backup the keys. Also, it repeated use a same bitcoin address, which lowers its security.

4.2.2 Deterministic wallet

All keys are derived from a master key, which is the seed. All keys in this type of wallet are related to each other, and if the original seed is available, all keys can be generated again. A number of different methods of key derivation are used in deterministic wallets. The most common derivation method is the use of a tree structure, called a hierarchical deterministic wallet or HD wallet.

4.3 Hierarchical Deterministic wallet[15]

In this part, we will introduce Hierarchical Deterministic wallet. For deterministic wallets, frequent backups are not required, but use elliptic curve mathematics to ensure that one can calculate the public keys without revealing the private keys. Then, to share public keys and recover information, we introduce hierarchical deterministic wallets such that selective sharing by supporting multiple key pair chains is allowed.

4.3.1 Convention

In the construction of HD wallets, elliptic curve cryptography uses the field and curve parameters defined by secp256k1[11].

The operations between two elements are: addition(as is defined by elliptic curve addition) and concatenation, which is appending one byte sequence onto another.

We also need to define some functions to regulate the arrangement of bytes:

- $point(p)$: returns the coordinate pair $[p]O$, where O is the base point of secp256k1
- $ser_{32}(i)$:serialize a 32-bit unsigned integer i to a sequence with 4byte
- $ser_{256}(i)$:serialize integer$p$ as to a sequence with 32byte
- $ser_p(P)$:serialize the coordinate pair $P = (x, y)$ to a sequence by SEC1's compressed form:$B_0||ser_{256}(x)$, where B_0 can be 02 or 03
- $parse_{256}(p)$: Turn a sequence with 32byte to a 256-bit number

4.3.2 Extended Keys

To realize the sharing of public keys, we define a function to derive some child keys from a parent key. We extend both private key and public key with an extra

256bits of entropy, which is called the chain code.

The private key is written as (k, c), where k is the normal private key and c is the chain code. The public key is written as (K, c), where $K = point(k)$ and c is the chain code.

For each extended keys, it can generate 2^{31} normal child keys and 2^{31} hardened child keys. We define the index for all child keys. For normal child keys, the index is defined from 0 to $2^{31} - 1$. For hardened child keys, the index is defined from 2^{31} to $2^{32} - 1$, which we use i_H to represent $i + 2^{31}$.

4.3.3 Child Key Derivation Functions

In this part, we will explain how to compute the child extended key from its parent extended key and its index i. We need to consider whether it is public key or private key, and whether we can use i_H to represent i.

Private parent key to private child key

The function CKDpriv$((k_{par}, c_{par}), i) \rightarrow (k_i, c_i)$ is defined as:

- - If $i \geq 2^{31}$:

    let I=HMAC-SHA512$(Key = c_{par}, Data = 0x00||ser_{256}(k_{par})||ser_{32}(i))$

  - Else:

    let I=HMAC-SHA512$(Key = c_{par}, Data = ser_p(point(k_{par}))||ser_{32}(i))$
- Split I into two 32-byte sequences, I_L and I_R.
- Return child key k_i: $parse_{256}(I_L)$+$k_{par}(mod\ n)$
- Return extended key c_i: I_R
- If $parse_{256}(I_L) \geq n$ or $k_i = 0$, the returned key is invalid.

Public parent key to public child key

The function CKDpub$((k_{par}, c_{par}), i) \rightarrow (k_i, c_i)$ only works for normal child keys which is $i \leq 2^{31}$:

- let I=HMAC-SHA512$(Key = c_{par}, Data = ser_p(point(k_{par}))||ser_{32}(i))$
- Split I into two 32-byte sequences, I_L and I_R.
- Return child key K_i: $point(parse_{256}(I_L)) + K_{par}$
- Return extended key c_i: I_R
- When K_i turns out to be the point of infinity or $parse_{256}(I_L) \geq n$, the returned key is invalid.

Private parent key to public child key

The function $N((k,c)) \rightarrow (K,c)$ is only defined for extended parent keys and child keys. The returned key K is point(k) and the returned extended key c is the passed extended key.

- N(CKDpriv$((k_{par}, c_{par}), i)$)(always works)
- CKDpub$((k_{par}, c_{par}), i)$) only for normal child keys

Public parent key to private child key

Undefined

4.3.4 The key tree

Then we build a tree to cascade several CKD constructions. Beginning with the master extended key m, we calculate CKD$_{priv}(m,i)$ for several values of i and have some level-1 derived nodes. Using the resulting extended key, CKD$_{priv}$ can be applied. Then we can bulid a tree structure depends on that, just as figure5 shows

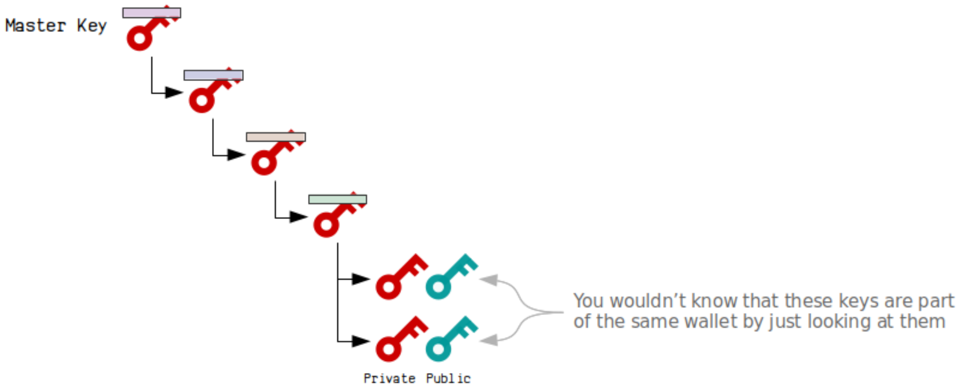

Master Key
You wouldn't know that these keys are part of the same wallet by just looking at them
Private
Public

Figure 5. The key tree

4.3.5 Wallet Structure

Now we arrive at the construction of HD wallet. Suppose we have several accounts and the default account numbered as 0, we give each account an internal keypair chain and an external one. Then, each account can use the external keychain to generate new public addresses. The whole structure is shown in figure6.

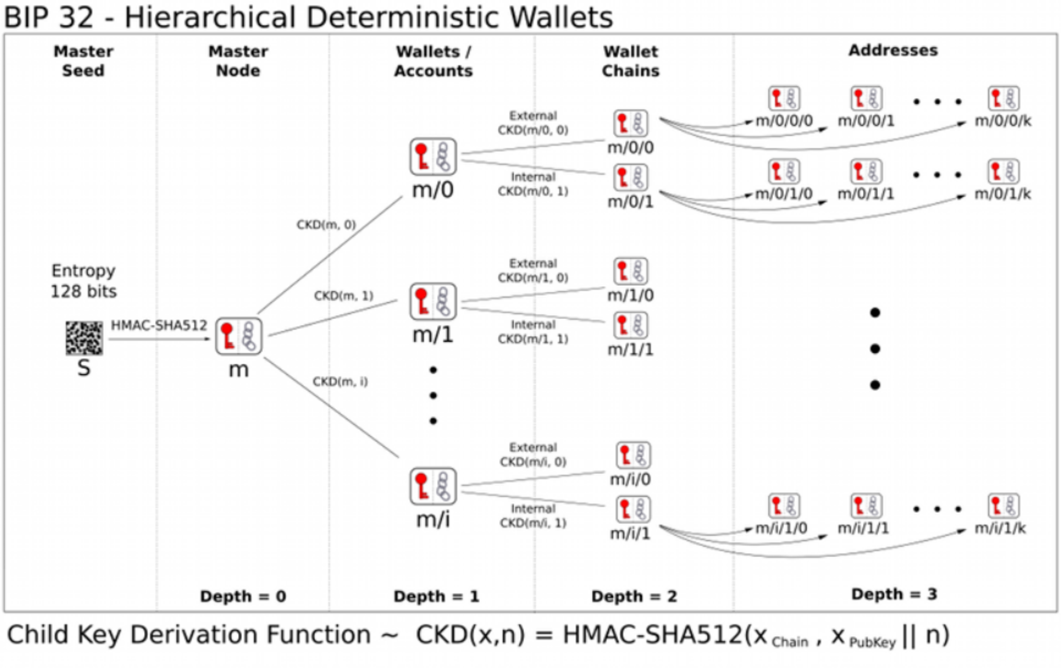


Figure 6. The wallet structure

Where $m/i_H/0/k$ is the kth keypair of external chain and $m/i_H/1/k$ stands for the kth keypair of the internal chain.

4.3.6 Transactions

Finally we will briefly state how to make transactions based on HD wallet. When two business partners want to often transfer money, one use the extended public key as the external chain of a specific account as "super address". Then for frequent transactions, the keys can just be taken from the key tree without requesting a new address for each payment. Lots of space and computation costs are saved.

5 Privacy and Security

5.1 Introduction

Overall, BTC is a safe transaction system based on cryptography. What's more, the rule of reward makes the profit of mining greater than the profit of hacking with the same GPU resources.
In this part, we will discuss the privacy and security of bitcoin in terms of each steps in bitcoin. However, there are still some issues in BTCs, so we also discuss about the possible attack and their soultions.

5.2 Privacy of Transaction

Since the process of transaction has been explained in the previous part, we begin to talk about the security without introducing the transaction. For the transaction, the privacy of users will be guaranteed by using the scheme of signature, which could protect the personal information as well as verify the transaction. Here we analysis the security of each algorithm used in transaction, to prove that transaction of BTCs is safe enough to provide P2P service.

5.2.1 SHA256

SHA256 is a member of SHA-2 crytographic hash funtions.The main process of SHA-2 are similar, which is shown in figure 7

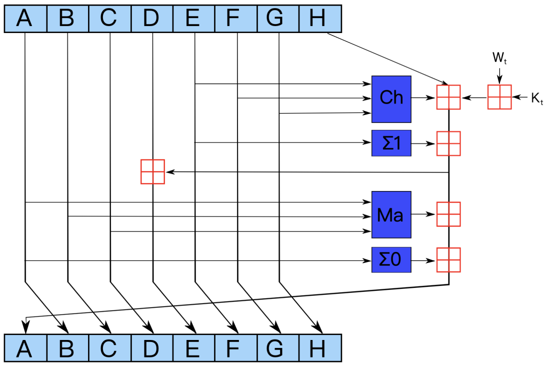


Figure 7. Process of SHA-2

In this figure, the blue components mean that :

$$
\begin{aligned}
\mathrm{Ch}(E,F,G) &= (E \wedge F) \oplus (\neg E \wedge G) \\
\mathrm{Ma}(A,B,C) &= (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C) \\
\sum_{0}(A) &= (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22) \\
\sum_{1}(E) &= (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)
\end{aligned}
$$

And the red squares is addition modulo 2^{32}.
For SHA-2, the security against collision attacks is from 112 bits to 128 bits, which is quite a safe algorithm in terms of current popular technology.This algorithm is used in the creation of bitcoin addresses to enhance the security of the privacy. What's more, during mining, SHA-256 also use it as the Proof of work algorithm.[4]

5.2.2 Merkle Trees

Merkle tree is a kind of binary tree based on SHA-256, which is composed of a root, some leaf nodes and some median nodes. For each parent node, it will hash the concatenate of value of its children nodes. During the validation, we use the value of one node and the value of all the other parent nodes to reconstruct the value of root node. If the value of reconstructed version is the same to the origin version, we could prove that this transaction exists[4].

Merkle Trees guarantee the safety and stability of BTCs, since it could find the loss and damage of data in a short time. This provide BTC a really low rate to send wrong information among users.

5.2.3 RIPEMD160

Based on Merkle-Damgard construction, BTC applys RIPEMD-160 algorithm, which is also a hash function.
Similar to MD5, the input is divided into data blocks with length of 512 bits.

5.2.3.1 Filling
We filling the input M with 1 in the first filling bit and 0 in all the other bits, until the filled text K satisfies: Length of K $\mod 412 = 448$.

5.2.3.2 Extra Length Value
Next, concat the length of M with 64 bits to the end of K. After this step, the length of K' could be divided by 512.

5.2.3.3 Initialize MD Cache
RIPEMD-160 use cache with 160 bits to store the midterm result and final hash value. Here is the value of initial value of the cache:

$$A_0 = 67452301$$
$$B_0 = EFCDAB89$$
$$C_0 = 98BADCFE$$
$$D_0 = 10325476$$
$$E_0 = C3D2E1F0$$

5.2.3.4 Process the Ciphertext The core of this processing algorithm is a compress function module with 10 loops, each of which is composed of 16 different steps. The whole algorithm could be divided into 2 parts, and each of them is processed with the same function with inverse order. In each loop, the system take the current K' and A,B,C,D,E as input, and to update the value of these cache. After the end of the final loop, the result is outputted with addition of result of calculation of 2 different parts.

5.2.3.5 Discussion of Security
Through the function module with 10 loops and each is composed of 16 steps with 2 processing lines, RIPEMD could guarantee its security because it is hard to generate the same hash value. According to Hans Dobbertin[7], it needs $O(2^{80})$ operations to find 2 text with the same hash value, therefore it needs $O(2^{160})$ operations to find the other text with the same hash value given text.

5.2.4 ECDSA

ECDSA(Elliptic Curve Digital Signature Algorithm) is used to validate the rightful owners, which is dependent on the curve order and hash function. In this algorithm, we define r and s as the signature, d_A and Q_A as the private key number and publich key number respectively, and define z as the hash of the message we want to sign.

5.2.4.1 Signing Algorithm

In this algorithm, we use z and d_A to compute signature pair r and s

1. According to the curve, we get the group order n.
2. Generate a random number k in [1,n-1] which is cryptographically secure
3. Get G as the generator point of the secp256k1 curve
4. $(x, y) \to k * G$
5. $r \to x \mod n$
6. If r==0, update K and start over
7. $s \to k^{-1}(z + r * d_A) \mod n$
8. If s==0, update K and start over

5.2.4.2 Verification Algorithm

1. Check whether r and s are with in [1,n-1]
2. $u_1 \to z * s^{-1} < \mod n$
3. $u_2 \to z * s^{-1} \mod n$
4. $(x, y) \to u_1 * G + u_2 * Q_A$
5. Check if x and y is not equal to the point at infinity.
6. Check if $r == x \mod n$
7. If all the previous steps hold, then the signature is valid.

5.2.4.3 Discussion of Security By using the Elliptic Cryptography, ESCDA could promise its high level of security.

5.3 Security of Wallet[15]

Speaking of the security of wallet, we could discuss its security under several conditions:

5.3.1 Known public key

Under this condition, it is impossible for an attacker to find the corresponding private key easier than solving EC DLP, which is assumed to require 2^{128} group operations.

5.3.2 Known Child extended private key and integer i

Under this condition, it is impossible for an attacker to find the parent key easier than solving a 2^{256} brute force of HMAC-SHA512

5.3.3 Known any number of tuples

Under this condition, if the first element of tuples are distinct, it is impossible for an attack to determine if they are generated from a common parent extended private key easier than solving a 2^{256} brute force of HMAC-SHA512.

5.3.4 Not existing properties

It is difficult to find i given both a parent extended public key and a child public key.
It is also really hard to find the first element of parent extended public key given the whole key and a non-hardened child private key.

5.3.5 Possible challenges

Since the information of a parent extended public key which is added by any non-hardened private key descending from it means that we have known the parent extended private key. Therefore, we should pay attention not to release the extended public keys. To solve this weakness, we introduce the hardened keys, which could prevent compromising the master or other accounts when leaking the account-specific private key.

5.4 Possible Attacks[6]

Although BTC is of high-level security, there still exists possibility of successful attack, here we list some popular attack and their feasible reaction.

5.4.1 Double spending

Explanation and examples
Actually this is not an attack, but is a common types of fraud for all versions of virtual currencies. Double spending happens when the same balance is spent twice. Here is an example, after I buy a set if GTA5 with 100BTC, I copy my blockchain and continue buying Stein-Gate0 with the same 100BTC, then the double spending will happen. However, BTC has its own verification, if this happens, only the first trade will be verified by all the nodes, which means that attack based on double spending is really hard to carry out.
For the double spending attack, it make use of 51%vulnerablility in the blockchain network. That is, if the attacker have "the computing power equivalent to 51% of the entire excavator network"[1], he/she could make fake blockchain with higher speed and compete with the real blockchain and fooled all the other miners.
Adverse effects
If the double spending succeeds, then the sellers will lose their products with no income, and the users will send a fake blockchain.
Feasible countermeasures
This could be solved by adding a third-platform to observe, or nearby nodes should be able to notify the merchant when double spending happens.

6 Conclusion

In this project, we discuss each parts of BTC(BitCoins). The whole system of BTC is composed of blockchain, transaction, mining and wallet. Blockchain is the core technology of BTC, which provides a way for people to transact without the verification of the third-way instruction. Blockchain works like a data chain. It is added a new tail with history of trade everytime and is sent to all nodes. After all nodes verifies the correctness of the trade, the blockchain is stored. Blockchain and its transaction system is a decentralized, disintermediate and safe process, which could be considered as a new-type currency. However, the trade of blockchain needs to be recorded and calculated, which leads to another important scheme-mining. While mining, all the nodes in this game solve proof-of-work, and the first successful node have the opportunity to create a new blockchain. Moreover, we still prove that the blockchain is a safe process for people to make transaction. However, there still exists some possibilities to be attacked, so we also introduce each countermeasure for those attacks. The safety of blockchain is not only related to the cryptography, but also related to the economics. For example, if the profit of attacks is lower than the profit of mining with the same amount of electronic resources, than the possibility of being attacked is lower.

Bibliography

[1] Rashmi Agrawal et al. *Blockchain Technology and the Internet of Things: Challenges and Applications in Bitcoin and Security*. CRC Press, 2020.

[2] Lear Bahack. “Theoretical Bitcoin Attacks with less than Half of the Computational Power (draft)”. In: *ArXiv* abs/1312.7013 (2013).

[3] *Bitcoin Block and Transaction Data Structure*. July 27, 2021.

[4] *bitcoin wiki*. https://en.bitcoin.it/. July 24, 2021.

[5] *bitcoinsv wiki*. https://wiki.bitcoinsv.io/. July 24, 2021.

[6] Mauro Conti et al. “A Survey on Security and Privacy Issues of Bitcoin”. In: *IEEE Communications Surveys Tutorials* 20.4 (2018), pp. 3416–3452. DOI: 10.1109/COMST.2018.2842460.

[7] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. “RIPEMD-160: A strengthened version of RIPEMD”. In: *Fast Software Encryption*. Ed. by Dieter Gollmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 71–82. ISBN: 978-3-540-49652-6.

[8] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. In: (2008).

[9] Lam Pak Nian and David Lee Kuo Chuen. “Introduction to Bitcoin”. In: 2015.

[10] *Nonce*. July 27, 2021.

[11] Certicom Research. *SEC 2: Recommended Elliptic Curve Domain Parameters*. January 27, 2010.

[12] *What Happens to Bitcoin After All 21 Million Are Mined?* July 26, 2021.

[13] Wikipedia contributors. *Bitcoin — Wikipedia, The Free Encyclopedia*. [Online; accessed 27-July-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Bitcoin&oldid=1035111521.

[14] Wikipedia contributors. *Blockchain — Wikipedia, The Free Encyclopedia*. [Online; accessed 27-July-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Blockchain&oldid=1033575964.

[15] Pieter Wuille. *BIP32: Hierarchical Deterministic Wallets*. 2012. URL: https://github.com/genjix/bips/blob/Master/bip-0032.