Hacash: A Crypto Currency System for Large-scale Payments and Real-time Settlement

Creator: Anonymity
Translator: Janet Wang (1Dv1A3tQfYNt4Aj3s3Z297GiZRDa7MfDXr)
Editor: Ken You (19T8NKm9cbzcpoaMxADdFco945aj6rQs32)

Abstract. This paper proposes an channel-chain-based orderly multi-signature real-time offsetting settlement method, which can expand the transaction per second without any upper limit, and can inflict predatory punishment for dishonest parties to ensure the security of funds and the financial system of encrypted electronic currency. This method has built-in composite signature addresses and hierarchical equity control addresses, multi-party signature transaction structures, full-category payment protocols and asset change protocols to fulfill most payment needs of modern finance, enterprises and individuals.The preemptive bookkeeping right reward distribution method that combines proof of work, proof of history and fork voting can effectively avoid double spending, prevent 51% attack and reduce the Matthew effect. Incentive mechanisms such as currency issuance, public account and channel fees, channel interest, and block diamonds that conform to economic laws are adopted, so that the entire system can maintain long-term effective operation without trusting any institution.

The basic principle of the channel chain settlement network is: each two accounts lock a number of funds to form a payment channel, and both parties can sign multiple payments in private. During the period, there is no need to broadcast to the entire network to confirm the transaction, and only the final submission of the balance distribution to the main network to retrieve the funds they have correctly owned, thus greatly expanding the number of transactions per second of the entire system. If one party ends the channel, its funds will be locked for a period of time, and if the other party presents evidence to the main network during this period to prove the other party to be fraudulent, the exposing party will seize all of the other party's funds, thereby forcing both parties to remain honest. Only need to connect multiple payment channels, and start from the receiver to have all the funds transfer parties sign in order until the payer signs, all relevant parties will receive and disburse money at the same time to ensure complete payment, instant transfer and fund security; the channel can charge a small amount of handling fee as an incentive to provide stable services.

1. Preface

1.1 Crisis

Throughout history, the development of civilization and economic progress has always been accompanied by a premise: cultural and financial freedom. Regardless of capitalism or socialism, whether developed or developing, the periodic economic crises have made people increasingly call for stricter control over the financial system. It results from the fact that different modern social and economic systems have one thing in common: mandatory sovereign credit currency and fractional-reserve banking system. This is essentially a kind of fraud, is the source of financial instability, and is the biggest cause of unfair distribution of

benefits other than violent plunder. The ensuing economic intervention policies either would exacerbate or overcorrect the economic environment, and repeatedly harm the economy and people's living standards that should have been stably growing, just like the dynasty change.

The unstable, unpredictable, and arbitrary intervention and manipulation of the financial environment will allow large capital to easily obtain excess value without entering the real economy. When the rate of return of capital, especially giant capital, exceeds or even far exceeds the average return of long-term overall economic development, the gap between the rich and the poor will inevitably widen at an accelerated pace until social conflicts, economic crises and turmoil begin. If a certain generation happens to come of age at the beginning of the Great Depression that lasted ten or even twenty years, then their destiny will be doomed to be difficult and unsuccessful. In order to smooth out the serious intergenerational injustice caused by this cyclical crisis, economic intervention and stimulus policies have been adopted, and the injection of stimulants has become a painful expedient (which perhaps results in more serious social unrest, revolution and even war). However, in the face of unfavorable population and debt trends, it is impossible to deceitfully deal with long-term structural crises through short-term stimulus.

The financial aid and monetary liquidity stimulus for banks and large enterprises is essentially to rob the poor and help the rich, to transfer the costs and losses to the whole society, and to make the whole people pay for the greed and wanton of a small group of people. We need to be able to solve the problem fundamentally, avoid or drastically reduce the financial crisis and economic collapse, so as to protect the interests of financially disadvantaged groups from being exploited.

In the Northern Song Dynasty of China, due to the inconvenience caused by the government's mandatory use of iron coins in Sichuan, the original paper money appeared among the people: Jiaozi. This seems to be a kind of "progress" in monetary finance. The government at the time found that it was a perfect tax tool, and then nationalized the right to issue Jiaozi. Simply by printing paper money, there is no need to spend a lot of money and resources to measure the land, count the household registration, collect money and make a ledger, and it becomes easier to secretly extract from the people a lot of wealth. However, the ensuing repeated hyperinflation and economic collapse broke the dynasty’s wishful thinking, and finally in the early Ming Dynasty, people completely lost faith in any pieces of paper money, and returned to the era of precious metal currency. They even divided silver into parts to trade by their weights. This seemingly primitive and downgraded currency system gave the people great financial and economic freedom, and achieved a rare period of financial stability in Chinese history. The living standards of ordinary people were improved. This situation continued until the late Qing Dynasty.

Currency comes from the market and will eventually serve the market. There is no high or low in its form, and good or bad in its essence. Only whether it is more adaptable and efficient matters. We need a currency market that embraces free and fair competition, and a transparent financial environment that resists fraud. The injustice in this world is essentially the injustice of information and the injustice of finance. The answer to this problem is to vigorously develop open finance and remove any access restrictions with open real-time auditing.

1.2 Future of Currency

Gold is the most suitable item on earth as currency, but people are not using it for large-scale transactions. The main reasons are: 1. It is bulky and inconvenient to carry; 2. It cannot be divided indefinitely for small payment; 3. Long-term circulation causes wear and damage; 4. The technology of counterfeiting and adulteration develops with time. In European history, gold-as-reserve paper money has been used for circulation for a long time. This method seemingly combines the advantages of gold and paper, but has its shortcomings as a means of daily payment: No individual or organization can withstand the temptation to create money out of air. No matter how firm the promise is, banknotes will always be overissued. It is no different from the historical situation where gold, silver, and copper coins are mixed with other metals to deliberately cause depreciation. And paper money even could cause severe inflation and lead to economic collapse which has appeared repeatedly due to the extreme reduction in issuance costs.

Two other controversial issues that prevent gold from being used as a universal payment method are: 1. The location of reserves is unevenly distributed, and countries with underground gold mines are almost collecting seigniorage from all over the world; 2. It is impossible to quickly adjust the supply of gold in the event of drastic changes in economic scale and its growth, which is likely to cause money shortages and deflation. In particular, the debate on the second point has been widely seen in the writings of various schools of economics, such as Austria, modern currency, and Keynes. Some of them believe that credit expansion is the source of the economic crisis and a scourge, while others firmly believe that currency without elasticity will lock up the development of the entire economy.

In history, currencies have roughly gone through the following stages:

a. General value items (grains, cloth, cattle, sheep, cigarettes, etc.)

b. Rare and stable objects (gold, silver, copper, gems, shells, etc.)

c. Trust accounting vouchers (certificates of deposit, debt certificates, anonymous checks, commodity certificates, paper gold, etc.)

d. Sovereign credit symbol (fiat currency)

Among them, the first and second stages are gradually eliminated due to the inability to meet the needs of modern commercial payment.The third stage perhaps gives rise to economic fragility and collapse caused by credit expansion and banker fraud. People immediately expect the central bank system established by the government to solve the problem, so here came the fourth stage, but the effect is obvious.

We believe that the future of currency, that is, the fifth stage, will be an open electronic network system with "recognized rules" as the main body and backbone, and "individual credit" as a branch and supplement, it will continue to improve and grow into an optimal system in the process of free competition. This "recognized rule" does not depend on anyone's will, and is not controlled by powerful interest groups, just like no one can create gold out of air. Fair rules allow the world to participate in the system effectively. Currency is far more than storing value or an accounting unit. It is a signal and an information system for economic operations. We can tolerate and repair partial credit failures, but absolutely cannot bear the consequences of the overall credit collapse.

In 2008, the emergence of Bitcoin and the blockchain technology behind it and its effective operation over the past ten years pointed out the direction for us.

1.3 What We Need

The core value of the blockchain is not decentralization, nor permanent preservation of data and evading supervision, but de-trust.

The emergence of Bitcoin is not to improve the efficiency of partial temporary payment or to reduce some instantaneous transaction costs, but to avoid or reduce the consequences of evildoing by companies and institutions that have failed our trust. Its vision is complete anonymity and decentralized peer-to-peer trust, while the future of blockchain is open and transparent and distributed self-auditing trust, which consumes cheap computing resources in exchange for the reduction of overall social transaction costs and the improvement of negotiation efficiency.

Just as the Internet satisfies the basic needs of human information and communication, the blockchain is to solve the big problem that has plagued mankind for hundreds of thousands of years: fraud. We have paid huge costs in history, consumed countless resources, and blocked too many deals. Therefore, we cannot add any possibility of fraud into the future monetary system.

We need a currency and a financial system to form a currency system with minimal trust. No one can arbitrarily reduce its quality. And without spending too much trust costs, it can better maintain the stable development of the economic system and fully meet the needs of modern commercial payment, corporate accounting and financial settlement.

2.Basic Principle

2.1 Technical Theory

Currency needs relative rarity and exclusivity to become a signal of value. No one will use water, soil or leaves as a medium of exchange. However, in the world of electronic information, everything can be copied almost at no cost. Then the currency with binary data as the carrier has a fatal essential flaw: the quantity is unlimited and it is difficult to distinguish between true and false. If we still rely on an online issuing agency or an electronic mint to ensure the real validity and upper limit of the number of coins, then from historical experience, no one can bear the temptation to create money out of air. In the end, what everyone has in hand will be worthless.

An intuitive solution is to code each electronic currency with an integer number, and announce an upper limit of the number (or the upper limit automatically increases by a fixed amount every year), and then mark each coin with the owner's electronic signature (ignore the question of who owns the currency first). Each currency will be paid by the current owner with the private key to sign the recipient's public key, indicating that the coin has been paid to the other party, and all historical payment records will be kept. This solves the problem of counterfeit and unlimited electronic currency.

However, the scheme proposed above has several obvious flaws:

a. One coin cannot be divided for small payment or change;

b. The same coin can be paid to two different people or even more people at the same time (double spending);

The first problem can be temporarily alleviated with a small enough denomination (the upper limit of the number is large enough to code each coin of different value). The double-spending problem is more troublesome. One way is to allow a public central database

to save and prove the record of each payment, but it still relies on the integrity of the person in charge of the agency, and there is a possibility of fraud. Another way is to let everyone have their own account book listing the record of every payment made by everyone through broadcasting, and check the owner of the coin on their own account book when receiving the coin, so as to avoid double spending.

The solution for everyone to have a ledger seems to solve the double-spending problem, but it still has big flaws:

a. Maintaining a set of ledger to continuously record all payments is costly, especially for people who don't often receive money. In the end, everyone will still rely on the accounts issued by some large institutions like banks.

b. When some payment notification is not delivered to all the account holders due to technical issues of broadcasting (for example, the submarine optical cable is cut), everyone's account records will be inconsistent. A large number of different versions will be formed over a long period of time, and everyone will not agree with each other, and eventually the payment system will collapse.

c. If everyone spends a lot of cost to maintain the ledger just to check the validity when receiving money, then this currency system is too inefficient, and will eventually lead to unmanned maintenance of the ledger because of free riders.

Of course, there is the most critical question: who should own these electronic currencies first?

2.2 The principle of self-interest

In the long run, any cooperation that unilaterally depends on the other party’s integrity and commitment without other incentives and checks and balances will ultimately be unsustainable.

Any social cooperation system that continuously effectively operates recognizes human greed and self-interest, and uses these weaknesses to maintain the self-operation of the system. Bitcoin is on the right track and has grown stronger during its ten-year development. It creatively combines account book records and currency issuance to provide sufficient motivation for everyone to maintain the account book, and adopts the method of proof of work to compete for bookkeeping rights so that everyone's account book is always unified, thereby cleverly solving the problem mentioned above. Of course, it does not use a system encoding each coin, but applies an unspent balance (UTXO) method to support small payments and change. The detailed theory of Bitcoin is not introduced here. Please refer to Satoshi Nakamoto's paper.

2.3 Problem

There is no free lunch in the world. Just like our derivation of "propose a problem-solve a problem-encounter a new problem" above, the monetary system of open competition and accounting still has two shortcomings that everyone has criticized:

1.Through the repeated calculation of the Hash algorithm to compete for the account record right of a certain period of time, the method of obtaining additional currency rewards consumes a lot of hardware and energy.

2. As every payment needs to be broadcast to all account holders and confirmed, the

overall payment efficiency is very low (approximately 7 transactions per second), and it also leads to high fees, which cannot meet the needs of modern business.

Strictly speaking, we do not think that the first point is a real shortcoming. In addition to short-term speculation and irrational prosperity, everyone will always find a balance between costs and benefits in a field of free competition. We use a lot of energy and resources because it is profitable in general, and the cost will eventually be compensated in other ways. Just as we did not exhaust all the oil and gold on the earth, because it is not economical to do so.

The idea that a currency system should consume as little energy as possible is very primitive. The ancients also had a similar question and prejudice: merchants just moved objects from one place to another and did not create anything, so why did they make so much money?

There is a trade-off between consuming energy (or any other form of resource) and avoiding harm:

a. Everyone could enjoy the benefits of free and convenient without paying any cost. The price is that the central organization that provides services may be compromised or used to harm everyone's interests in the dark.

b. Or pay a certain cost (money or energy) to avoid the risk of someone or organization cheating ourselves.

If someone claims to achieve the above two points at the same time (both completely free and absolutely safe), either they fail to recognize the essence of the problem, or they are unkind.

The most intuitive way to temporarily alleviate the second problem is to expand the capacity, that is, expand the upper limit of the block size or reduce the block generation interval, but this cannot fundamentally solve the problem. The block size has a theoretical upper limit, and too large a number will prolong the download and transmission synchronization time, which further limits the upper limit of the block interval time. Also, new transactions will always fill up the expansion space, and the amount of transaction payments in the world is far greater than the current theoretical upper limit of expansion restricted by the hardware system. On the other hand, the expansion of the amount of data to a certain extent will also result in the failure of personal computers to save a complete account book, and eventually eliminate most of the participants. Consequently the complete data will be grasped by some large institutions with strong financial resources.

Another solution is to extremely reduce the total number of recorders (such as 21 or 101), set higher requirements and standards on the performance of their work, and give them new coins as incentives. This way, the entire system can carry a large transaction volume. This seems encouraging, but as mentioned above, this solution has its inevitable shortcomings, and its nature is not much different from the historical mint branch system.

There are also some people who are willing to tolerate temporary data inconsistencies in exchange for a great increase in overall system throughput (a generalized DAG structure). This can only be applied to scenarios where data authenticity is not strictly required or does not require immediate verification, but not a small instant payment system.

Not thinking it as a problem, or not considering it clearly, Satoshi Nakamoto did not specify in the paper the solution to carrying huge transaction volume.

3.Channel Chain Settlement Network

3.1 Basic assumptions

According to the operation of the world, strong adult men can easily snatch the goods of women and children on the street for their own possession, but most people do not do this because everyone knows that the police and legal system behind them will make them pay more. Facts have proved that even if there are risks, as long as there is a strict punishment mechanism, everyone's rational profit-driven behavior will maintain the overall effective operation of the system.

We propose that the public ledger should not contain all transaction records, but should be used as a strict, accurate and effective arbitration system and final clearing system that cannot be manipulated to ensure that payments made in private cannot be fraudulent, otherwise severe punishment will be imposed. In an unlimited number of repeated peer-to-peer transactions, everyone will tend to remain honest for long-term cooperation. This system is highly elastic, can dynamically increase transaction volume on demand, and theoretically has no upper limit, so that it can fully meet the needs of modern business development.

3.2 Main principle

We need an instant, high-frequency, secure small payment system, which can be used for purchases in online malls or physical stores. If there is an intermediary who provides financial services, then everyone should also ensure that the income and expenditure are correct. The main point is that the settlement of the capital chain should be real-time synchronous rather than asynchronous, otherwise it will bring serious centralization and capital security problems.

First, we need to create a series of two-way settlement channels. They are customers and capital channel service providers (referred to as nodes), merchants and nodes. When necessary, customers and merchants can directly establish settlement channels. When a payment is initiated, the merchant or the node serving the merchant queries the route of the capital channel chain, establishes the TCP connection of the entire chain, and then obtains all the channel ID, the last transaction hash, transaction serial number, balance confirmation and other information from all nodes. After that, a complete transaction is established and sent to all participants. Then, starting from the merchant, everyone sends their signatures of the payment transaction to all other parties in order until the customer’s signature is received by everyone, at which point the merchant signs the transaction confirmation message and closes sequentially all TCP connections from the end of the chain. In the end, all relevant parties received (and disbursed) the funds at the same time, and the payment transaction was completed. Each channel can charge a small service fee to incentivize nodes to provide stable services.

3.3 Technical realization

The following describes all the technical details and the data status after each step is completed.

1) Create a joint settlement channel

First create a joint-signature transaction:

```
{
  // Open a settlement channel, fund1 and fund2 are the investors
      // Once the channel is opened, the balance of both parties will be deducted until the channel is closed
  // The lock-up period (number of blocks) at the end of the settlement channel
  lock: 2016, // approximately equal to one week
  // Custom ID of settlement channel, randomly generated
  channelId: 232353253456,
  fund1: {
    address: '1313Rta8Ce99H7N5iKbGq7xp13BbAdQHmD', // Investment address
    amount: 1234, // Investment amount
  },
  fund2: {
    address: '19aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
    amount: 1234,
  },
}
```

Both parties sign the above transaction and broadcast it to the main network to be confirmed, and then two-way payments can be made privately for unlimited times and frequencies. All transactions generated after this do not need to be broadcast to the main network.

2) Private settlement

For each payment, both parties sign the settlement information and exchange signature results at the same time. The transaction structure is similar to:

```
{
  // [Off-chain settlement] Confirmation of phased balance distribution
    // If a phased balance distribution is submitted on the chain, the address will be locked
for the agreed time,
  // settlement channel id
  channelId: 232353253456,
      // The hash of the last confirmed channel transaction (this field is the hash of the transaction where the transaction channel was opened at the time of the first transaction)
  prevTrsHash: Buffer.alloc(32),
  // Channel transaction serial number, automatic increment
  autoincrement: 123135,
   // Confirmation of fund distribution after write-off (the part that one party has more than the other party)
  diffConfirm: {
    address: '1313Rta8Ce99H7N5iKbGq7xp13BbAdQHmD',
    amount: 1234,
```

```
    }
}
```

If every customer and every store signs a settlement channel, it will be too much trouble and will lock up too much money. We envisage some nodes that can provide channel connection services and form a channel chain settlement network with each other. Merchants and customers only need to sign a settlement channel with a small number of two or three nodes, and then they can easily make transactions and payments with everyone else through this network. It is similar to when you access the Internet, you only need to access a broadband operator, and you don't need to pull a separate network cable from everyone else.

3) Channel route

Suppose that customer A signs a channel with node X, merchant D signs a channel with node Y, and node X and node Y also create a settlement channel. At this time, if the customer A wants to pay to the merchant D, the funds will flow to D's address through a chain composed of A, X, Y, and D, through query and search between nodes (or routing query by a third party, similar to a domain name DNS server) to find the shortest possible (or lowest handling fee) path, and form the two-way path of the TCP connection in order:

A <=> X <=> Y <=> D

4) Building a chain payment transaction

It can be seen from the above that the payment funds of customer A pass through two nodes X and Y until merchant D has to pass through three settlement channels. At this time, by the merchant's service node Y inquiring, or by means of active two-way broadcasting, all participants have obtained the ID, transaction serial number, handling fee, balance confirmation and other information of each settlement channel and confirmed it. Then, merchant D constructs a channel chain payment transaction, similar to the following structure:

```
{
    // Channel transfer transaction
    amount: 1234, // payment amount
    // way of passage
    channels: [
        {
            /**** Channel 1: (A => X )****/
            // Settlement channel custom id
            channelId: 232353253456,
            // The last transaction hash confirmed by both parties
            prevTransactionHash: Buffer.alloc(32),
            // Channel transaction serial number, automatic increment
            autoincrement: 123135,
            // Channel handling fee, which can be zero or negative
            fee: 12,
            // Confirmation of channel difference when this transaction is completed
            diffConfirm: {
                address: '19aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
```

```
            amount: 1234, // amount
          },
        },
        {
          /**** Channel 2: (X => Y)****/
          /* ...omitted... */
        },
        {
          /**** Channel 3: (Y => D) ****/
          /* ...omitted... */
        }
      ],
    }
    // The above data is only an example, the number of array elements in the `channels` field will be three, the last two have the same format and have been omitted
```

The above transaction data will be broadcast to all participants, and each channel will only take a single element in the channels array as a settlement certificate. Because nodes X and Y have two channels at the same time, they can receive and spend funds at the same time, maintaining a balance of payment and receipt, and no one will lose money because of the breakdown or offline of other nodes.

5) Orderly signature

The four parties have received the transaction and confirmed it. If there is an information, financial or technical error, any party can disconnect the TCP connection, thereby closing the entire channel chain and terminating the payment.

Enter the signing phase at this time:

a. Merchant D first signs the transaction with the private key, and then sends the signature to node Y

b. Node Y receives D's signature, verifies it and forwards it to Node X, and then signs the transaction itself and sends the result to X and D

c. Next is node X repeating the actions of Y, forwarding the signatures of D and Y, signing the transaction and sending it

d. At this time, all parties including customer A have received the signatures of D, Y and X, and the entire channel is waiting for customer A's signature

The reason why merchant D must sign in reverse order in the direction of capital flow is that the receipt signature depends on the payment signature to take effect. The node must first confirm that the other party has signed the receipt of the funds before signing the payment. The entire process is chained. The customer and the merchant must confirm that all nodes in the channel chain have signed the transaction, so that the funds can be fully credited to the account in real time after payment, otherwise it is possible that the customer has signed the payment, but the merchant may not receive the money.

6) Payment receipt

The status of the entire transaction and channel at this time depends on the signature of customer A. Once A signs, it will all take effect at the same time:

a. Customer A receives the transaction data and all the signatures of merchants D, Y, and X, verifies and confirms that they are correct

b. A signs the transaction and sends the signature result to X

c. X receives the signature and forwards it to Y, and Y forwards it to merchant D, and the entire settlement channel is completed

d. D signs a message with the private key to confirmconfirme that the payment has been received successfully, and sends the result to node Y, and then disconnects the TCP connection

e. Y forwards the message to X, then disconnects; X forwards it to customer A, and disconnects

f. Customer A receives the receipt from merchant D, and then all connections are disconnected; the entire channel chain settlement is completed, and the payment is successful

7) Each channel's settlement and handling fee

For each individual channel in the channel chain, both parties have received a complete transaction, including all the transfer records of funds in the entire chain. Each only needs to go to the channel that matches with them in the channels array to make settlement, and the settlement amount is customer A's payment amount minus the handling fees of all previous channels.

Each channel can charge a small amount of handling fee (fee field) in order to repay the cost of fund lease, similar to the income of loan interest. Usually the ratio of the handling fee depends on two points: 1. The amount of funds paid; 2. The cost of hardware network services. Note that the handling fee can be zero or even negative. The channel that charges negative fees will receive less funds after signing a transaction than the expenditure at the same time, which can be regarded as a subsidy method to attract customers to open up the market.

8) Error handling

We must ensure that funds arrive in the account simultaneously and instantly, and no one may suffer losses due to this. Any error or problem occurring in any party during the payment process will terminate the payment of the entire channel chain and disconnect all connections:

a. If a technical failure causes any party to disconnect before receiving the final payment receipt, the channel chain is terminated

b. The signature verification fails and the channel chain is terminated

c. If the payment amount or handling fee is incorrect, the channel chain is terminated

d. The transaction serial number, last settlement hash or balance confirmation is incorrect and the channel chain is terminated

e. The signature expires and the channel chain is terminated

The node or merchant can set a timeout period (for example, 3 seconds). If no subsequent signature or receipt is received after the expiration time, the TCP will be disconnected and the entire channel chain will be terminated. The purpose is to avoid incomplete transactions, which can lead to the loss of one party’s funds.

Through the above data exchange process, we have completed the entire payment behavior, and all parties are securely credited to the account in real time. The small amount of fees charged by intermediate nodes can motivate them to provide stable services.

3.4 Channel closed

A channel will sign a large number of payments within a period of time, and repeat two-way payments for multiple times. If the two parties have no dispute over the final balance allocation, they can sign a transaction indicating the final termination of the channel and withdraw their respective balances, and broadcast to the main network for confirmation:

```
{
  // Both parties confirm the balance and close the settlement channel
  // Balance allocation takes effect immediately without lock-up period
  // settlement channel id
  channelId: 232353253456,
   // Confirm the balance allocation and diffConfirm is the difference between the two balances
  diffConfirm: {
    address: '19aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
    amount: 1234, // amount
  },
}
```

Once the above transaction is confirmed on the mainnet, the funds locked in the settlement channel are immediately returned to both parties.

3.5 Arbitration protection

Since the fund is locked in the settlement channel by both parties jointly, if one party loses the private key, the other party will not be able to unlock and withdraw its own funds in the channel. Considering that one party maliciously refuses to sign to close the channel, or for other reasons, temporarily unable to cooperate with the other party, we need the ability to unilaterally terminate the channel.

The solution is to broadcast the most recent channel chain transaction (or reconciliation transaction that includes multi-party signatures) to the main network for confirmation by reference. By the clear balance in the transaction, the other party (the terminated party) will receive the funds immediately. The party who proposed to end the channel (the terminating party) also will unlock the funds. But the price is that the account will be locked for a pre-arranged time (lock field, one week for example), and the balance cannot be transferred during this time period, in order to avoid arbitrarily (maliciously) ending the channel by any side.

If one party submits to the mainnet a transaction that is not reflective of the latest balance distribution but is beneficial to this party, and wants to unilaterally terminate the channel and seize the other party’s funds, as shown above, the submitting party’s account will be locked and the balance cannot be transferred. At this time, the other party can submit the latest balance confirmation to the main network (judging by the autoincrement of the transaction serial number). Once confirmed, the latter will immediately seize all the funds of the former, including all the in-channel and locked balance. Through severe punishment for

evildoing and rewards for proof (late-mover advantage and peer-to-peer game) to keep both parties honest.

Considering that the contribution of one party in the consumption or salary settlement channel may be 0, or the party in the end has spent all the funds, then the cost of doing evil is just the fee for submitting the channel transaction on the main network. But once the seizure of funds is successful, this party will get a lot of benefits. There might be accounts choosing to do evil at low cost in this situation. If so, a certain amount of funds can be locked on the main network as compensation for evil actions in multiple settlement channels.

If the honest party decides to terminate the channel on its side at the cost of the account being locked for the agreed time, neither party will lose funds since the other party cannot give (non-existent) updated balance allocation. By repeating the peer-to-peer game infinitely, everyone will tend to choose honesty and cooperation.

3.6 Balance payment

Imagine the following situation: a merchant first signs a channel with a channel service node to receive payment. At this time, the merchant's investment in this channel should be zero, and the customer pays the merchant through the settlement network. After a period of time, the merchant signs another channel with the node to purchase or pay wages. At this time, the node's channel investment should be zero.

For the convenience of accounting statements, these two channels are both one-way payment channels. One is only used for receiving payments, and the other is only for spending. However, there is an upper limit on the funds locked in the channel, and in order to increase the efficiency of funds utilization, the amount will not be too large. After a period of time, all the funds in the channel will be transferred to the other party’s account and the transaction cannot continue. At this time, the other party can only continue to add channel funds on the mainnet, or simply close and reopen a settlement channel of a larger amount. Because it will be very inefficient to wait for the confirmation of the main network and pay a lot of fees; it will lock up a large amount of more and more funds, and the entire system will not be able to bear it in the long run.

This problem also arises when the personal consumption channel is separated from the wage channel.

We use channel hedging and balancing to solve this problem. The underlying technology principle is the same as the channel chain transaction. Except that in this hedging transaction, only a receipt and a payment channels are included. At this time, the merchant and the node (or the individual versus the node, the node versus the node) are both the payer and the receiver at the same time. The transaction structure is similar to:

```
{
  amount: 1234, // payment amount
  // way of passage
  channels: [
    {
      // Settlement channel custom id
      channelId: 1111,
```

```
            /**
             * Merchant's payment channel, transfer funds to the node
             */
        },
        {
            // Settlement channel custom id
            channelId: 2222,
            /**
             * The merchant's payment channel, the node transfers the funds transferred from the collection channel back
             */
        },
    ],
}
```

The result of this transaction is that the balance in the merchant's receiving channel is transferred to the payment channel, and it is an atomic operation, and neither party has the risk of losing funds.

All parties involved in the settlement network can hedge and balance accounts regularly. There is no need to frequently interact with the main network and lock too many funds, so that the entire settlement network is always in a state of high utilization, and only a small amount of funds are used to support extremely large transactions.

3.7 Decentralization

Economies of scale and opacity of information make it impossible for us to completely avoid the existence of financial and data intermediaries. However, the restricted access and monopoly can make a coachman (intermediary) become a robber. Excessive centralization will lead to serious collapse of a single point, tax collection effects and a crisis of trust.

Take Bitcoin's Lightning Network as an example, if most transactions are concentrated in a few intermediary points for capital flow, then they become de facto banks. Once a node fails, it causes a large number of transactions to terminate in an instant. The funds stranded in these channels will be explosively submitted to the main network for unlocking, causing serious congestion and soaring fees, which even exceeds the handling fee of some channels with little in-channel funds.

We should try our best to avoid centralization. The channel chain settlement network has two features to avoid this problem:

1) The overall instant receipt of in-channel funds settlement

Once the payer signs the transaction, all participants' funds will be credited and spent at the same time, and will not stay and pile up in the channels of any intermediate nodes. Even if a technical failure or other force majeure causes the node to go offline, it will not affect the already existing transactions.

2) Channel signing lock-up period

A channel only supports the completion of a single transaction at one time, but cannot achieve concurrent payment and receipt, which reduce the transaction scale to a certain extent.

But there are several advantages: 1. Guarantee fund safety; 2. Simple and clear reconciliation; 3. Transactions will not be blocked; 4. Prevent centralized nodes;

The principle of the fourth point is: the relevant channel has been locked during the period before the payer signs the transaction and after connecting to TCP (other transactions cannot be supported at the same time, and the lock time may be tens of milliseconds). The transaction volume bottleneck makes it extremely uneconomical to lock a huge amount of funds to meet most of the payment needs of the centralized channels. And eventually a large number of small channels will provide a completely decentralized service, thus avoiding single-point breakdown and centralization crisis.

The decentralization of channels and the reduction of the funds in a single channel also brings another benefit: it makes the funds embezzled too small to be attractive.

3.8 Express channel

Channel locking guarantees security, but at the cost of reducing transaction throughput. Considering that the same node may set up different business divisions, and different nodes may also establish long-term trustworthy cooperative relationships. For some small and micro payments (such as buying a cup of coffee), there is no need for completely instant reconciliation between nodes.

We can use delayed reconciliation (such as once an hour) to greatly increase the transaction throughput between designated nodes. Technically speaking, the serial-locked transaction verification is changed to a concurrent mode, that is, both parties do not confirm the final distribution of the balance for each payment, but first allow the funds to pass, and then check the bill later. This mode allows the transaction volume per second of the specified channel to rise from 10 to more than 2000 (depending on the device performance).

The data structure is roughly:

```
{
   // Type of reconciliation (1. Instant reconciliation 2. Delayed reconciliation)
   type: 2,
   // Capital flows
   side: 2, // foud1 => foud2
   // 8 type settlement channel id
   channelId: 232353253456,
   // 8 type channel transaction serial number, automatic increment
   autoincrement: 123135,
   // Channel fee
   fee: {
      unit: 8,
      amount: 1234,
   },
}
```

Between the nodes supporting the express channel mode, we only need to compare the channel serial number list and the corresponding customer payment signature list after a period of time to determine whether the relevant transaction is successful, calculate the

correct balance of each, and sign the reconciliation.

For the business divisions of the same node, the express channel mode does not have any security risks. No matter if it is a node, a customer or a merchant, there will be no loss of funds, because the delay in account reconciliation is confined to the node, while outside of the channel, receipt and payment meet.

For different nodes in close cooperation, security depends on the business reputation and the result of unlimited repeated cooperation games. Due to concurrent payments, the actual amount owned by one party of the channel may be negative for a period of time, and it is impossible to withdraw from the mainnet before signing the reconciliation. The risk can be reduced to an acceptable range by limiting the amount of micro payments and increasing the frequency of reconciliation.

The risks caused by the large-scale adoption of the express channel model among merchants, customers and nodes will be discussed in Chapter 8.

3.9 Fund calculation

Assuming that on average a payment has to go through three channels connected by two intermediate nodes, we can calculate:

*T: total lock time = N: number of channels * S: data step * (t: TCP transmission time + c: verification calculation time)*

Substitute data 3 * 3 * (20ms + 15ms) to get 315ms, which means that a channel chain can support three transactions per second on average (worst scenario). If 100 units of funds are transferred to the channel chain network, in the case that all transactions use one-to-one payment, then the daily transaction volume is as follow.

*(100: total number of channels / (3: number of channels * 2: bilateral peer-to-peer funding)) * 3: transaction volume per second * 60 * 60 * 24 = 4320000 units*

If it is a one-way fund flow balancing receipts and payments, the fund flow will double in the best case: 8,640000 units. The upper limit of fund utilization rate reaches 86,400 times per day, 2.6 million times per month, and more than 30 million times per year. This means that we only need to lock in 0.0000116% of the funds to support the payment volume and the total issuance volume within a day.

Assuming that a transfer fee with ratio of 1/10 million is used, the total daily fee is 0.864 units, and the annual net rate of return is about 315% excluding compound interest.

4.Transactions

4.1 Basic data structure

To ensure the efficiency of system operation, the data structure of transaction should be as simple and compact as possible (stupid simple but efficient work), and be easy to understand for both humans and machines. (The importance of readability of financial rules for humans will be discussed in the chapter on design principles)

In general, it can be divided into the following three levels:

Blocks >> Transactions >> Functions, assets (Actions)

Example in json format:

```
{
 version: 0, // Block version number
 height: 0, // block height
 timestamp: 0, // block timestamp
 prevHash: Buffer.alloc(32), // hash of the previous block
 mrklRoot: Buffer.alloc(32), // Merkle root of all transactions

 /* other extend field ... */
 transactions: [// All transactions contained in the block
  {
   type: 1, // // transaction type
   timestamp: 12313423, // transaction timestamp
    address: "xxxxxxxxxxxxxxxxx" // The default main address of the transaction (the address of the fee payment)
   fee: {// Transaction confirmation fee
    unit: 248, // handling fee unit
    amount: 1234, // handling fee
   },
   actions: [// The specific asset object of the transaction or the actions performed
    {
     kind: 1, // Asset or action type (1 means transfer)
     bill: {// Total amount of transfer
      dist: 2, // precision space
      amount: new Buffer(), // amount
      unit: 248, // unit
     },
     address: "ooooooooooooooooo", // transfer address
    }
   ],
   signs: [// signature
    {
     publicKey: Buffer.alloc(32), // public key
     signature: Buffer.alloc(64), // signature value
    }
   ],
   multisigns: [// Composite signature
    {
     publicKeyScript: Buffer.alloc(32, 96), // public key script
     signatureScript: Buffer.alloc(64, 192), // signature result script
    }
   ],
  },
```

```
  ],
}
```

As you can see, a transaction is roughly divided into three parts: actions, signs and multisigns. As to why not use smart contracts and other more flexible and "advanced" transaction structures, we will discuss in Chapter 9.

4.2 Composite signature address

A single-signature address has the risk of losing or the secret key being stolen, and cannot meet the needs of co-escrowed funds. We need a function that allows two or three private keys to manage funds with different permission configurations, such as:

1) A and B make a joint deposit, and both people need to provide signatures to withdraw funds

2) Joint account of husband and wife, either of them can pay from the joint account

3) Exchanges, online wallets, and offline private keys--at least two of them are required to transfer funds in case of theft and loss

The multisigns compound name in the transaction structure is a compound address that supports two or three private keys, so the optional administration modes are 1/2, 2/2, 1/3, 2/3, 3/3 and so on. Therefore, a composite signature address can support up to 200 private keys to manage a composite address.

There is no secret private key in the compound address, but a piece of data that is made up of multiple public keys. And a public private key is derived from this piece of data. Each transaction needs to provide the derived public key as the basic script, and then provide a list of signature data for verification.

4.3 Hierarchical equity control account

The composite signature address can solve the problems of losing secret keys and keys being stolen, and it also can meet the need of a simple co-escrow. However, in the face of complex business structures (mainly beneficiary rights and voting rights), we need an account system that can support modern corporate equity structures.

Such accounts must meet the following characteristics:

a. Can be managed by several private keys to avoid security problems

b. Can replace (add, delete, modify) the administration private key, but the address is fixed

c. A changeable voting ratio

d. Support Weighted Voting Rights (WVR)

e. Can protect funds from lost in extreme cases

1) Construct

Due to the above characteristics, we need to save and manage accounts on the mainnet. The transaction to open an equity account is similar to:

```
{
// 1~10000 satisfies the effective ratio of votes (ten-thousand points ratio) (must be equal to or greater than this ten-thousand points ratio to operate the account)
    validRightsRatio: 6666,
// compose list
```

```
forms: [// Quantity within 200
// Voting rights and beneficial rights can be unequal (that is, Weighted Voting     Rights)
    {
          address: '19aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
          //    4 byte, 0~4294967295, number of equity
          rights: 1,
          //     4 byte, 0~4294967295, voting rights
          votes: 3,
    },
    {
          // member can be a compound address
          address: '29aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
          rights: 3,
          votes: 5,
    },
    {
          // Members can also be other equity accounts
          address: '39aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
          rights: 2,
          votes: 3,
    },
  ]
}
```

After the mainnet confirms the transaction, it will add the transaction timestamp to the member's address in the database to form a piece of data and generate a private key and a public key as a newly created equity account.

The members of the equity account can be ordinary addresses, compound addresses, or other equity accounts, and they indicate the corresponding voting rights and beneficial rights. The main network will store all address control structures in the database, and each address has a control system.

2) Verification

To check whether a transaction initiated (or participated in) by an equity account is valid, it is necessary to read the control tree from the database and check whether there are enough member signatures in the signature list. In the case of sufficient votes, the remaining signatures do not need to be checked.

An equity account may be controlled by multiple other equity accounts, and the upper-level equity account contains members of the higher-level equity account, similar to the multi-level investment relationship between companies. As a result, a transaction with a large address even requires hundreds of signatures to be verified, which will consume a lot of data space and slow down the transaction confirmation speed. We adopt the method of calculating transaction fees according to the size of transaction data space, which can prevent frequent transactions of super large equity accounts to a certain extent. In fact, special financial accounts can be authorized to take charge of daily ordinary payments, since transactions that

require equity account signatures are only for very low-frequency transactions such as major industry investments.

3) Management

After the equity account is registered on the main network, it can add or delete administrators, and reset authorization. Members who need to meet the number of votes can sign a change of transaction to complete. Similarly, the ratio of effective votes can also be changed.

The membership and authorization can be changed arbitrarily, and the address of the equity account remains unchanged.

4) Minimum account deposit

The equity control address is a very precious resource, which will take up a lot of data space and verification time. To avoid waste, when registering and adding (except for changing and deleting) administration members, in addition to allowing the transaction verifier to charge the ordinary transaction confirmation fee, it is also necessary to lock in the account the same amount of funds as the fee for account maintenance fee.

Before address registration, it is required to send a certain amount of funds to the equity account to ensure that the minimum amount of the account is sufficient to complete the registration. (Before the account is officially registered with the main network, the target address is generated locally, and funds are transferred to the local address on the main network.)

5) Logout

To avoid the inflation and waste of the state database space, the equity account allows account cancellation (by deleting the corresponding member list and control tree). The minimum account deposit will be returned to other designated accounts when the address is cancelled.

6) Security of funds

It is likely to happen that the private keys of a group of members may be lost due to force majeure, causing insufficient signatures to reach the effective number of votes. In this case, no transaction operations can be performed, and the organization's funds are actually lost forever. Besides, members with greater beneficial rights but less voting rights will suffer more serious losses. We need to be able to withdraw funds safely under such extreme circumstances.

It is designed in a way that any upper-level member address in the equity account control tree can initiate the fund protection mode.The condition for initiation is that the account needs to lock funds equal to 1% of the amount in the equity account, and the lock-up period is half a year. If within half a year, another member address raises the cancellation request, or the private key of the equity account has been retrieved and a transaction is initiated after the retrieval, the fund protection mode will be automatically exited.

After half a year, the account that raised the protection mode can withdraw the funds in the equity account by initiating another transaction to withdraw it to its own account, thereby avoiding permanent loss of funds in extreme situations.

4.4 Multi-party signing

Imagine an equity investment scenario: a company (referred to as Party A) accepts

10,000 units of capital investment from the investor (referred to as Party B) and transfers 20% of the equity. In this case, if either party A or B operates first, there is a risk of fraud: A transfers shares first and B does not transfer capital, and B transfers capital first and A may refuse to transfer shares.

We need a transaction that can complete both capital injection and share transfer operations at the same time. If any side fails, the other side automatically fails. At this time, both parties need to jointly sign a transaction:

```
{
  type: 1, // transaction type
  timestamp: 23423442, // transaction initiation time
  // Trading assets, actions
  actions: [
    {
      kind: 6, // from transfer to to
      from:'xxxxxxxxxxx', // address
      to:'ooooooooooo', // address
      bill: {// transfer amount
        dict: 1,
        amount: 10000, // amount
        unit: 248, // unit
      }
    },
    {
     kind: 7, // Added equity management members
     forms: [
       {
         address: '19aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM',
         rights: 3, // Earning rights
         votes: 3, // number of votes
       },
     ]
    }
  ]
}
```

This transaction contains two actions. After both parties sign, the transfer of funds and the distribution of equity will be completed at the same time.

However, there is still a bug in the above transaction: if the company preemptively broadcasts a transaction within the period when the signature of the above transaction is completed but not confirmed to take effect, and the original equity is issued one hundred times to the original shareholder, after the transaction is confirmed successfully, the new investor’s shares have been diluted to a negligible level. Hence, we also need to include a conditional action:

```
{
kind: 9, // Indicates that the voting rights and beneficiary rights of the
      corresponding address must be above a certain ratio before the
transaction can take effect
   address:'xxxxxxxxxxxx',
   targetAddress:'ooooooooooooo',
   rightPercent: 20, // % of income right
   votePercent: 18, // % of voting rights
}
```

This conditional action is ranked third, which means that after the above capital injection and share transfer operations are completed, the final equity ratio is required to be no less than 20%.

4.5 Payment Type

In order to meet the needs of modern financial payment, we should support multiple payment methods based on UTXO and balance, and fully consider the emergence of some payment service providers in the future. The examples of the payment types that will be provided:

In order to meet the needs of modern financial payment, we should support multiple payment methods based on UTXO and balance, and fully consider the emergence of some payment service providers in the future. The examples of the payment types that will be provided:

A. Pay by yourself

b. Ask the other party to pay to yourself (the other party needs to sign this transaction)

c. Let A pay to B, and you only have to pay the transaction fee (A's signature is required)

d. Pay the outputs using the the designated funds in inputs (the signature of the owner of the inputs is required)

e. Pay the outputs using all funds contained in inputs (the signature of the owner of the inputs is required)

For equity accounts, there will be some special asset change operations:

a. Assign the designated amount to all members in proportion to the income rights (share dividends)

b. Assign the designated amount to the vote holders according to voting rights (administration incentives)

c. More details of payment methods and data structure will be given in the appendix.

4.6 Signature Stripping

Due to the complexity of payment methods and equity control systems, a transaction will contain a large number of signatures. Half of a block may even be used to contain signature data. In order to save space and speed up synchronizing data of other transaction verifiers, the signature data must be strippable.

Technically speaking, the data and order in the signature list (including multi-signature and composite signature) are not calculated into the hash value of the final transaction. This way, the core data of the transaction can be stored or transmitted separately from the signature data.

After stripping, multiple participants in a transaction can sign independently of each other, which is conducive to independent decision-making in business activities.

4.7 Handling fee

Taking into account the needs of large-scale commercial payments, some service providers will pay for customers in advance and waive transaction confirmation fees. Some packaged or mixed payment services may also appear. The signature of the fee payer and the ordinary transaction participant should be separated.

Technically, a transaction contains only one payment method for fee, and the original transaction data signed by the fee payer contains the fee field. The signatures of other people do not contain the information related to the fee, and they only have to sign and confirm their own transactions.

Since the transaction fee of the main network is always fluctuating, the fee payer can adjust the fee and re-sign the transaction at any time, so as to obtain a more economical and suitable transaction confirmation time, without bothering each participant to repeat signing the transaction.

For the transaction confirmer (miner), the only hash of the transaction does not include the fee field, which can eliminate repeated transactions and achieve dynamic bidding for fee.

4.8 Field Format

A fully developed transaction payment system should adapt to the long-term needs in the future. The method expressing the amount of funds can both retain almost unlimited precision and save space as much as possible:

```
{
  bill: {
    dict: 1, // Indicates the positive and negative funds, and the amount of space
    amount: Buffer.alloc(), // amount amount
    unit: 248, // unit, representing a few zeros after the amount
  }
}
For data such as handling fees, there is not such a high precision requirement:
{
  fee: {
    amount: 1234, // amount
    unit: 240, // unit, decimal
  },
}
```

Other detailed field formats and explanations will be given in [Appendix].

5.Incentive

The reasons why an economic system can function well for a long time are simple: 1. Let people who devote creativity and improve efficiency make more money; 2. No one can get free lunch. All economic advantages are rule advantages, a result of system advantages.

Some traditional monetary theories believe that money is neutral and a constant that can be eliminated and offset. Based on this absurd assumption, one of the most important questions remains unanswered: who should own money first?

Currency is either itself a commodity or a representation of commodities. Just as no money can not be exchangeable for commodities, there are no commodities without property rights. The property right system and the market economy are two sides of the same thing. Without clear property rights, there will be no real market. Likewise, justice and efficiency are also two sides of the same thing. Without justice, there is no efficiency.

The fairness of property rights is the prerequisite for all economic efficiency. Crypto currency is not absolutely fair, but it can greatly promote such fairness.

5.1 Rewards of competition for bookkeeping rights

Maintaining the correctness and consistency of public ledgers is the most important task in the crypto currency system, and we should give sufficient rewards to achieve such goals. The creative combination of competition for accounting and currency issuance in Bitcoin fuels up the efficiency of the system's operation.

We set the ledger to be updated every 5 minutes (a block containing a list of new transactions is generated, and a specific hash algorithm keeps trying until it finds the data that meets the difficulty requirements, called Proof of Work, which is broadcast to all others immediately after generation.), the first transaction in the new block generates a certain amount of new currency, which is awarded to the ledger verifier (the miner) who first calculated the target data. Other verifiers receive and check the transfer amount and signatures in the newly generated block, and restart the calculation after this block, in an attempt to find the next block hash data that meets the difficulty requirement and get rewards. Everyone will automatically adjust the target difficulty value by calculating the number of blocks generated in a period of time to ensure that the ledger can always be updated every 5 minutes on average when the computing power changes.

The amount of new currency produced is adjusted once a year in the first stage, rising from 1 to 8; in the second stage, it is adjusted once every ten years, falling from 8 to 1; the third stage, the output permanently remains at 1. After 66 years, the number of currencies will be 22 million. The detailed currency issuance algorithm will be given in Chapter 6.

5.2 Public account fees

While encouraging competition for accounting to generate new blocks, it is also necessary to encourage blocks to contain as many effective transactions as possible, otherwise the ledger system will waste resources on idle. A transaction needs a certain amount of handling fee, which is obtained by the miner who records the transaction and generates the block. The amount of fees is determined by a dynamic bidding mechanism.

Moreover, since all transactions must be verified by all miners, the recording capacity and space size of the public ledger are always scarce. Except for the price we are willing to pay, we cannot accurately evaluate the level of urgency and value of the transaction. Paying

higher fees will get priority to processing; this method is a relatively more efficient way of identification.

5.3 Channel service fee

We elaborated the channel chain settlement network in Chapter 3, and envisaged the emergence of nodes dedicated to fund circulation and payment services. The amount of service fees charged by nodes depends on bidding, hardware network fees, and channel costs.

5.4 Channel interest

Similar to telephone lines and the Internet, a sufficient number of channels form an all-covering network to make the most of the system. We need to encourage everyone to put the extra currency that is temporarily unused into the channel to provide settlement services.

Since opening the channel also needs to pay the confirmation fee, just like a normal transaction, we proportionately set a very small amount of new currency to reward both parties of the channel that locked the funds to offset the fee. Through the limited-precision data format (omitting fractions), channels with large amounts of funds get less rewards than those with small amounts to stimulate the generation of more channels. Because of the confirmation of the fee and the limitation of real purchase payment amount, the channel funds will remain within an appropriate range, not too large, nor too small.

Starting from when the channel is locked successfully, based on the total funds locked by both parties, the interest of every 10,000 blocks (about 34 days, if insufficient, ignore it) will be settled once by compound interest, and the single settlement will be one thousandth, which means that the the annual interest rate is about 1.056%. When the channel is closed, new currency is generated, and is distributed in proportion based on the average value of the funds owned by both parties when the channel is opened and closed.

5.5 Block diamonds

An ideal currency that only exists in theory: no transaction costs, and the total amount changes in real time with the growth and consumption of total social wealth. Similar to a virtual gold with unlimited reserves, when the productivity increases, more coins are mined and enter currency circulation; when the productivity decreases, the output is automatically reduced due to the mining cost. In this way, inflation or deflation caused by drastic changes in currency supply and economic recession can be prevented. However, we have to face the cruel fact that it is impossible for reality to perfectly follow the theory.

The amount of output for bookkeeping rewards and channel interest is fixed, and its supply will not change with productivity or the market. We need a currency growth mechanism that can adapt to the fluctuations of population and technology cycles, and automatically adjust output. Based on market competition, more new coins will be generated when computing power increases, and the difficulty of mining will only increase and not decrease, so that the output of the new currency is immediately reduced or terminated when computing power decreases for market reasons.

We agree that a block diamond is a string of data that is compressed and calculated by a 32-bit hash value to meet a specific format. Each block can only contain at most one diamond (or not, depending on the computing power). The production algorithm is:

hash256((genesis_block_hash || prev_diamond_block_hash) + belong_user_public_key + nonce_number) ==> length_16_string

Specifically, the genesis block hash or the previous block hash containing diamonds plus the public key of the target owner, plus any random number is hashed to get a 64-bit string value, similar to:

35534631f31dfcf12200cdbad65c66ffb9d3fbd3ac985aa8a401bc4c3616bab3

Perform a special compression operation on the result derived in the previous step, and map every 4 bits to the character list 0WTYUIAHXVMEKBSZN to obtain the following 16-bit string result:

0NMSAK0ZYNSNBAZM, 00000000IXVKHNHZ or 0000000000UKNWTH

When the result satisfies the requirement that at least the first ten digits are 0 and the last few digits do not contain 0, a diamond is produced. Based on the above result, we call the literal value (identifier) of this diamond UKNWTH, and the literal value is unique and unrepeatable. At this time, the diamond is packaged into a block and broadcast. All diamond producers stop the previous calculations, and use the hash value of this new block to restart the calculation of the literal value of the next diamond. If multiple diamonds are generated in the same block time, the miner chooses one to pack into the block (maybe the one with the highest handling fee).

The total limit of diamonds is about 17 million. Every time one is excavated, the overall difficulty of excavation will increase correspondingly. As the number of excavations increases, it increases exponentially, and eventually approaches infinity.

Block diamond is a kind of high-dimensional heterogeneous currency, which can achieve the effect of dynamically regulating the money supply. Its value is determined by the mining cost and market.

5.6 Data Service

In addition to the internal reward mechanism described above, we also need some special data to calculate related services provided by merchants, and they will charge some service fees based on the calculation result, such as:

a. Channel routing
b. Trading mixed packaging
c. Transaction confirmation query
d. Monitoring malicious termination of channel
e. Encrypted private key escrow
f. Credit audit
g. Data security review

And it will be possible to produce special hardware devices for the channel chain settlement network such as transaction signature machines.

6.Currency

The most powerful technology is always dedicated to meeting the most urgent needs and solving the most serious problems. If the generalized distributed public ledger technology cannot reform the world monetary system to reduce the plunder, oppression and exploitation of financially disadvantaged groups, let alone others.

6.1 Total of growth

In the long run (static equilibrium rather than short-term speculation), ignoring the impact of use value, if an asset deserves hoarding, its appreciation rate must be greater than the average social production profit rate, otherwise everyone will sell it to invest in other projects. However, if the expected appreciation rate of assets with a constant total amount is exactly equal to the social productivity profit rate, a small amount of additional issuance expectations will not restrain investment and consumption. The destructive power of the deflation trap will manifest only when people are forced to use a single currency and borrow huge amounts of debt due to inflation. Inflation will also force the poor to invest. Since they do not have the information advantage and risk diversification strategies, they are more plundered.

Unlike paper money or gold, crypto currency is impossible to retrieve once lost. If the total amount remains constant, it will lead to excessive hoarding and speculative bubbles in the long run, and will lead the economic system to "stock competition" instead of creating value in new areas, damaging its core function as a currency. We need to introduce an inflation expectation to avoid the above problems, so that even if the actual amount of currency issuance does not exceed the actual economic growth rate, the currency will not really depreciate.

Since it is theoretically impossible to precisely adjust currency supply in circulation in response to the changes in the scale of transactions (the issuance of an additional unit of currency for every additional unit of commodity produced by the society is just a beautiful illusion), the only thing we can do is minimize the interference of changes in currency levels to economic activities. A relatively more feasible approach is to give an observable expectation of additional issuance, so that everyone can estimate the amount of currency growth within a certain period of time, and rationally arrange consumption, production and sales activities based on the market price index and purchasing power index after accounting adjustments. For example, when the purchasing power of money continues to rise, as long as the appreciation rate is stable and predictable, the amount of wages paid by the company will decrease proportionally over time rather than remain unchanged. Workers can also accept this approach, because the choice of contract is subject to the constraints of specific asset price changes. Though the long-term inflation has made people accustomed to the rise in wages, they still know that their wages have the same purchasing power becuase of appreciation.

1) Bookkeeping rewards

We use the Fibonacci sequence to determine the monetary rewards of the block. In the first stage, the amount of every 100,000 blocks is adjusted once every 0.95 years, and the reward gradually increases; in the second stage, the amount of every 1 million blocks

is adjusted once every 9.5 years, and the reward gradually decreases; in the third stage,

the reward finally remains constant at 1 unit per block:

1, 1, 2, 3, 5, 8, 8 (ten years), 5 (ten years), 3 (ten years), 2 (ten years), 1 (ten years), 1 (ten years), 1, 1, 1, 1, 1 ....

Then the total supply in the first 66 years is 22 million units:

*(1 + 1 + 2 + 3 + 5 + 8 + 8*10 + 5*10 + 3*10 + 2*10 + 1*10 + 1*10) * 100000 = 22000000*

Since then, the annual additional issuance ratio (1*100000)/0.95/22000000 is about 0.4785%, and thereafter it decreases year by year (about 0.3289% after 100 years, about 0.2506% after 200 years, and about 0.1462% after 500 years), and approaches zero indefinitely.

2) Channel interest

To acheive the large-scale application of the channel chain settlement network, we set a settlement cycle with 10,000 blocks of about 34 days. The funds locked in the channel are rewarded according to compound interest, and the single reward rate is 0.1%. Assuming that channel funds account for 1/2 of the total, the annual issuance rate is approximately:

((1+(0.001)) ^ (365/(5000/288)) - 1) / 2 ≈ 0.0053 (0.53%)

Adding bookkeeping rewards, the total annual increase in issuance during the steady period is roughly estimated to be 1% to 1.5%, and it will converge to about 0.53% to 1% indefinitely. Just for reference, the world's total average GDP growth rate from 1960 to 2012 was about 2% to 3%.

3) Diamond mining

The literal value of the diamond is composed of 16 characters of WTYUIAHXVMEKBSZN, and the last 6 digits of the hash value calculated are the legal literal value. The total quantity is:

16^6 = 16777216

We design that it takes about 25 minutes to mine a diamond for every 5 blocks. Regardless of the surge in difficulty, to mine all diamonds takes at least:

16777216*5*5 / (60*24*365) ≈ 800 years

Up to 60*24 / (5*5) (about 58) diamonds can be dug every day, and a maximum of about 21,000 can be produced every year.

The calculation difficulty is adjusted every 3277 diamonds (3277 = 16^6 / 256 / 20). When the first 20 bits of the 32-bit hash value are all 0, the mining difficulty reaches the maximum and all diamonds will be dug out at this time. However, due to the characteristics of hash calculation, the difficulty of mining will increase exponentially. In fact, it is

impossible for all diamonds to be dug out. Depending on the level of computing power, after a certain balance point (for example, several million), mining a new diamond needs to consume a large amount of computing power of the entire network, which will make the marginal output of diamond mining smaller and smaller. But the marginal cost will only rise, thus ensuring the scarcity of diamonds in the market.

Block diamonds are the accumulation of production surplus and the coffer of the economic system.

6.2 One-way transfer compatible with Bitcoin

As "everything can be copied" in the electronic world, compared to its predecessors B-money, Hashcash and many others, Bitcoin solves two seemingly contradictory problems that none of them can solve: 1) Double spend ; 2) Decentralized issuance administration.The crypto currencies before the emergence of Bitcoin already possessed important features such as peer-to-peer networks, asymmetric encryption, proof of work, and network-wide broadcast ledgers. They could only introduce a "booking center" or "minting authority" to prevent double spending. Or they could adopt decentralized issuance and management, but the existence of double spending must be tolerated. These two contradictory and seemingly irreconcilable problems are the core reasons why the "predecessors" of Bitcoin have not received a lot of attention. Bitcoin creatively combines maintenance of the ledger and issuance of new currency using the "blockchain", achieving scarcity in the electronic world, and making it the first widely recognized electronic product with "inherent value".

We believe that although Bitcoin is revolutionary, it is not perfect, especially its monetary nature. For example, its total issuance of 21 million and a 4-year halving of output make it only a kind of electronic commodity with actual value, and it cannot assume the function of daily payment and settlement currency. Our mission is not to overturn and replace Bitcoin technically, but to learn from Bitcoin's "blockchain" and "computing power issued currency" technologies. Base on the "commodity currency theory", we focus on expansion, improvement and completeness of the "monetary" indicators in the system, bring in Bitcoin to form a complete hierarchical currency system, and combine with the channel chain real-time settlement network to promote the large-scale use of cryptocurrency in personal payment and commercial settlement.

Taking "Bitcoin bifurcation chain" or "Bitcoin two-layer network" as an example, although the system's transaction volume per second can expand finitely, it cannot fundamentally improve its "monetary" defects. In fact, we believe that "currency" is inherently invariable and cannot be "improved" by nature, because currency essentially represents a "value exchange contract" that can be fulfilled in the future (based on the expectation of scarcity). This kind of contract is reciprocal and balanced at the beginning, but if this kind of contract can be easily changed in the future, it means that at least one party will have to bear unexpected losses. This kind of irreversible loss is expected to greatly prevent such coin from exercising its monetary function, which means that non-market-oriented monetary system other than the enforcemente of power will eventually fail. What prevents Bitcoin from becoming a currency for daily payment and settlement is not its technical limitations such as "block size", but a serious lack of "monetary characteristic".

We recognize the great revolution and value led by Bitcoin, and try to improve it based

on its merits. In reality, after Bitcoin is mined, whether it is stored in an exchange account or is mortgaged on Ethereum to issue a Bitcoin anchor, everyone will recognize that these "Bitcoin certificates" have the same value as their denomination. In fact, for cryptocurrencies such as Bitcoin, the recognition of value has nothing to do with its location and form of expression, but only with its "proof of scarcity." We adopt the "irreversible one-way transfer" method as a systematic improvement plan.

The basic technical principle is: in the new system, use the same “private key--address” account generation algorithm as Bitcoin, and send Bitcoin on the main network in integers from a certain address to a "black hole address" which is generated by a certain technology and whose private key is unknown to anyone. Then generate a corresponding "transferred Bitcoin" in the new system and send it to the original payment account, thus transferring the Bitcoin one by one, and proving the scarcity (total amount) of "transferred Bitcoin" in terms of technology. This process is irreversible, and because the Bitcoin are not sent to an account that others can embezzle, there is no trust risk in the whole process.

However, because the new system could not get the same attention as the original Bitcoin main chain in the early days, placing the value of the first few Bitcoin sent to the "black hole address” at risks. If the new system ultimately fails to gain widespread acceptance, these Bitcoin that were first transferred to the new system can actually be considered lost or destroyed. However, assuming that more and more Bitcoin get transferred and more people recognize the value of the new system, this risk will become smaller and smaller, and ultimately insignificant. More importantly, more and more "transferred Bitcoin" will also bring higher added value to the new system, and greatly enhance the application of both in payment settlement and open finance.

Based on the above two reasons, we design to issue different amounts of new currency as a reward for offsetting risks and adding system value while transferring Bitcoin, and send them to the account that transfers Bitcoin to the "black hole address". The first few Bitcoin will receive more new currencies, and then the reward gradually decreases, until one new currency is rewarded for each tranfered Bitcoin. In response to market volatility and short-term speculative behavior, the initial number of new currencies derived from transferred Bitcoin will be locked and released linearly every week (while the transferred Bitcoin will not be locked). The lock-up period of the first new currency derived from transferred Bitcoin is approximately 20 years for linear release; the second is 5 years; the fourth, fifth, sixth, and seventh are about 2.5 years respectively, until the transferred Bitcoin are enough and the amount of new currency issued each time is small enough, and the subsequent additional issuance will be unlocked when the market fluctuation is not significant. See the appendix for the specific amount of additional issuance and lock-up period.

At this point, there are three different energy levels and heterogeneous crypto currencies in the new system: 1) The total amount is absolutely limited, indivisible, and uniquely identified block diamonds whose mining difficulty will only increase; 2) The total amount is limited and divisible, transfer Bitcoin; 3) A new currency with unlimited total volume, divisible indefinitely. There are three sources of new currency: 1) bookkeeping and mining; 2) transfer Bitcoin to get additional issuance; 3) channel interest.

6.3 Units and symbols

If crypto currency wants to be truly used in the field of commercial payment, rather than being collectible like gold, it must be able to achieve large-scale secure real-time transaction settlement, stable incremental production, and unlimited division.

Infinite division guarantees that no matter how large the economy develops, small payments can always be made. Digital encrypted currency should completely avoid the transaction fee problem caused by the physical form of traditional currency.

We use a special data structure similar to scientific notation to store the amount of funds:

```
bill: {
   // 1 byte, 0~255, unit (a few zeros followed)
   unit: 248,
    // 1 byte represents the space occupied and the sign, 1~127 is positive, 128~255 is negative
   dist: 1,
   // 1~127 byte transfer amount
   amount: Buffer.alloc(),
}
```

Where a unit represents a decimal unit, such as {amount: 1, unit: 4} represents 1000, {amount: 137, unit: 8} represents 13700000000.

We set unit=248 as 1 currency, and use 100 million as the base, 1 = 100 million baht, and so on to set up five units:

unit: 248 is 1 mei = 10^8 zhu
unit: 240 is 1 zhu = 10^8 shuo
unit: 232 is 1 shuo = 10^8 ai
unit: 224 is 1 ai = 10^8 miao
unit: 216 is 1 miao

In daily accounting, 273.58 zhu can be recorded as ど 273.58:240, and 1 mei is recorded as ど 1:248.

The introduction of the unit unit means that we can divide a currency into (1/(10^248)), and the order of magnitude of atoms in the observable universe is about 10^80.

6.4 Prohibition of artificial monetary policy

Currency should not be used to regulate the economy. This is laziness and too dangerous.

The modern commercial economic ecosystem has become more and more complex, and has developed from sparse grasslands to the Amazon rainforest. The use of monetary policy to regulate the economy is similar to using only precipitation to regulate the growth of rainforests. Grassland can prosper as long as there is water, and the formation of rainforest requires more conditions and time. Things are not as simple as it seems. A healthy economic ecology can only grow and evolve under suitable conditions, and cannot be precisely planned. Today's "monetary policy" has become a set of institutionalized exploitation and wealth

redistribution systems.

People are always overconfident in themselves, but they can't understand that the laws of emergence of some complex systems cannot be completely deconstructed and modeled. The emergence of crypto currency is not to replace legal currency, but to create new financial rules and business models in a brand new area. So applying the traditional monetary and financial system on the crypto currency system is fruitless. Financial and economic rules have always been coerced by powerful interest groups for a long time. We need to fight for the financially disadvantaged groups to protect the fruits of hard work from endlessly exploitation and fraud. It is worth noting that the ultimate beneficiaries of large-scale and long-term monetary policy regulation must be those closest to money and power, which will inevitably widen the gap between rich and poor. As a result the poor will sink so deep into the quagmire of poverty that it is completely impossible for them to jump out of the trap with their own efforts. People at that time would expect a more powerful government to enforce redistribution. Social, cultural and economic production will drop to the bottom or even go backwards for more than a decade, which will eventually lead to a catastrophic human tragedy.

It is very stupid to allow certain people or organizations with more power to decide on the core value parameters of the currency system, such as the algorithm, quantity or speed of currency issuance. This is not an act of arrogance similar to religious fundamentalism. But the key to the future monetary system is to provide expected and stable development that cannot be manipulated. If some certain core value parameters are extremely unreasonable or unsuitable, let better ones replace them.

7 Privacy

It seems that some people do not understand that being honest does not mean being public. The biggest problem is that without privacy, the "fungibility" of currencies will be greatly affected, so that the same unit of crypto currency will be forced to have different market prices due to its historical records, thereby reducing the efficiency of the entire monetary system; and Merchants who steal cancellation fee data will analyze historical purchase behaviors and give everyone a price that is just acceptable. This will greatly harm our interests. If the products provided are monopolized, the consequences can be imagined.

7.1 Anonymous

In public ledgers, anonymity is actually a pseudonym and cannot always remain hidden. As we always have to voluntarily or be forced to reveal our identity in certain scenarios, the entire transfer chain could be traced, exposing our privacy. Anonymous address is just the base. We need other measures to cut off the direct connection between the payment and the receiving account to avoid being tracked.

7.2 Mixed payment

Normally, payment and receipt correspond one-to-one to each other in a transaction. One person initiates a payment to another and broadcasts it publicly. It is easy to infer the connection between the two.

A feasible solution is to have a group of people jointly initiate a transfer of the same amount to another group, so that the recipient and the payer cannot be accurately matched. The greater the number of people involved in the transfer group, the better the privacy protection. It is called Mixed Fixed Payment. The transaction structure is close to:

```
{
  // Mixed fixed payment
  kind: 6,
   fee: {// The service fee charged separately for each address; it can be zero or even negative
     amount: 1234,
     unit: 248,
  },
  bill: {// Identical transfer amount
     dist: 1,
     amount: Buffer.alloc(),
     unit: 248,
  },
  addressCount: 100, // Number of addresses participating in mixing payment
    inputAddresses: ["1313Rta8Ce99H7N5iKbGq7xp13BbAdQHmD", "..."], // multiple payment addresses
    outputAddresses: ["19aqbMhiK6F2s53gNp2ghoT4EezFFPpXuM", "..."], // multiple receiving addresses
}
```

This method of mixing does not need to rely on complex technologies such as ring signatures. It is simple and easy to implement, and has the following advantages:

a. Can compress the size of transaction data, increase the main network throughput, and save handling fees

b. Some people with strong privacy needs may attract enough payment-mixing participants by zero handling fees or even subsidies, which is a win-win situation for all parties

c. No need for the signature of the payee (offline to account)

Meanwhile, it also has some disadvantages:

a. The transfer amount is highly uniform and difficult to be used for normal commodity purchases

b. In order to adapt to transfers of different sizes, it is necessary to divide into multiple integer gradients for mixing, making it more difficult to gather a sufficient number of participants

c. There is still the possibility of being tracked, although this probability can be extremely low through multiple times of mixing

7.3 Pre-deferred payment

Payment mixing obscures the direct connection between the sender and the receiver of funds “spacially” (receiving and paying throughput, but the order is mixed). We need more security by blurring the time related to the whereabouts of funds. The principle is to use an

intermediary to immediately transfer out, but defer the receipt of funds, so that the ultimate recipient of the transaction cannot be known for a period of time.

Assuming that the payer is A, the intermediary is B, the final payee is C, and the transfer amount is 100 units, the basic steps are:

a. B creates a transaction trs1 that transfers 100 units to C and sends it to A

b. After receiving the transaction trs1, A creates a special transaction trs2 and sends 101 units (including 1 unit of handling fee) of funds to an encrypted temporary address addrx, which stipulates that after trs1 being effective for a period of time, B can receive 101 units of funds from the temporary address; by showing B trs2 to ensure B the safety of funds

c. A signs trs2 and broadcast it to the mainnet and confirm it takes effect

d. B signs trs1 and broadcasts it to the main network and confirms it takes effect. C receives 100 units of funds which are immediately available

e. After a period of time (for example, 6 months), B initiates a transaction trs3 withdrawing 101 units of funds from the temporary encrypted address addrx, and the overall transaction is completed

The above method has security risk due to the order of sending and receiving funds: A sends the funds to an encrypted address, but B cannot sign the trs1 transaction due to the loss of the private key, which actually causes A to lose 101 units of funds.

It should be noted that there is no interest motivating B to intentionally not sign, because B can obtain the corresponding handling fee for completing the entire transaction. Even if B's

account balance is insufficient, he can still borrow from a friend, and return it as soon as he gets handling fee and funds back. There are two situations that may cause the loss: 1. B's secret key is lost or cannot be signed due to force majeure; 2. B simply wants A to lose;

To avoid this security risk, A adds a condition when constructing trs2: if B still fails to receive funds after a timeout (for example, one year), 101 units of funds can be returned to account A. The risk at this time is transferred to B: the funds need to be received within the 6-month window time period, otherwise there will be the possibility of losing 100 units. Since B earns a fee, he can hedge this risk.

trs1 is an ordinary transfer transaction, and the transaction data structure of trs2 is:

```
{
  // Send funds to a one-time encrypted address
  kind: 9,
  bill: {// amount
    dist: 1,
    amount: Buffer.alloc(),
    unit: 248,
  },
  // 32-bit hash as the encrypted address = sha3-256 (pre-transaction hash + transaction confirmation block + designated receiving address)
  hashaddr: Buffer.alloc(32),
  // Time-out retrieval, which means that you can retrieve the funds by yourself if the transaction is not successful after one year
  overback: 105120, // Number of blocks, set to zero for permanent validity
```

```
}
```

The trs3 transaction data structure for withdrawing funds from the encrypted address is:

```
{
  // Withdraw funds from the one-time encrypted address
  kind: 10,
  // The encrypted address that was taken out
  hashaddr: Buffer.alloc(32),
  // Must be already exist transaction hash (precondition for withdrawal of funds)
  existTransaction: Buffer.alloc(32), // the hash of trs1 above
  // The number of confirmed blocks before the transaction
  confirm: 50000, // about half a year
        // Specify the receiving address as the address of the fee payment, and calculate with existTransaction and confirm that the hash must be equal to hashaddr, which proves the right to receive
}
```

Except for the risk of locking up funds, pre-deferred payment will not cause anyone to lose funds. Under conditions of sufficient trust (for example, trs1 and trs2 are signed in real time on site), trs2 can be set to be permanently valid, which allows the intermediary B to broadcast trs3 transactions only when actually needed. This time may be very long (for example 5 years), thereby ensuring A's privacy and security.

7.4 Encrypted settlement network

Although payment mixing can reduce the probability of being tracked, the transaction is still made public. If a company has a large amount of identity information corresponding to the account address for big data analysis, we are still at risk.

In addition to expanding transaction throughput on demand, the encrypted channel chain settlement network can also protect our privacy to a certain extent because almost all payment data is spread off-chain and not made public. Ordinary payment channel nodes may be required to provide the identity information of all connected customers, and may leak all data regarding consumption.

A feasible solution is to encrypt the channel chain transaction with the public key of the participants (including the payer, the recipient and the intermediate node), so as to avoid information leakage due to interception by unrelated parties. The disadvantage is that this scheme still relies on the security technology and security strength of the node.

7.5 Channel Offsetting

Strictly speaking, channel offsetting only hides the actual amount of funds owned by both parties of the node in the channel from ordinary consumers in each specific channel chain payment, and there is no technical guarantee. Those with bad intentions can still calculate their own funds by scanning the channels registered on the mainnet.

However, in the express channel, since there is no need to confirm the amount of funds in real time, it is impossible to know the amount of funds on both sides of the channel unless

all transactions of the channel in a settlement cycle are obtained.

We need to protect consumer privacy and important commercial secrets, but we do not provide absolute anti-censorship functions, because the latter will consume several times the data space and computing resources of the former, and can easily become shelter of smalicious and harmful behaviors such as ransomware, mining Trojans, etc. We will discuss it in Chapter 9 Design Principles.

8 Risks and Prevention

8.1 Channel chain signature deferral attack

The channel chain (especially after the express channel is used) can greatly increase the transaction throughput of the entire system. In a more developed stage, almost all commodity payments and transfers are made through the settlement network.

To ensure the security of strange instant payment and real-time receipt, the channel will generally keep being locked since the channel is established until the final receipt is signed. Depending on network conditions, it may be locked as long as 100 milliseconds to two or three seconds. During this period, because the channel is in an exclusive state, other transactions cannot be processed. In this way, the commercial competitors or malicious saboteurs of the service node can initiate massive payments of extremely small amounts for specific channels, and intentionally delay the payment signature. Each time the lock timeout expires, the attacked channel will be blocked by a large number of small payments for a long time (similar to a DDoS attack on the Internet) and will not be able to process other normal commercial payments.

The solution is that each node records a temporary cache data, which is used to record the address of the payer, the cumulative payment amount and the total lock time occupied. Divide the two to obtain a score, which represents the channel utilization score per unit time. If the score is particularly low or obviously abnormal, you can limit the payment frequency of the corresponding address or simply deny service to prevent such channel chain signature deferral attack.

8.2 Low-cost channel fraud

The normal operation of the channel chain settlement network relies on a severe threat of punishment: all account funds will be seized once evildoing is committed. Considering that sometimes the actual fund balance of a single side of the channel is extremely low or zero, then this party has a motive to broadcast the balance distribution beneficial to itself on the main network. If the other party does not closely monitor such activity, after a period of time, its funds will be lost, and the cost of the evildoer is only the confirmation fee for the public ledger.

To solve this problem of dishonesty, technical means alone are insufficient. Certain insurance and punishment mechanisms have to be adopted:

1) It is agreed that each account retains a certain amount of risk deposit, which can provide insurance for multiple channels at the same time. Once any one of the channels violates the rules, the risk deposit will be taken away. At this time, all other channels will be closed to prevent another sabotage.

2) Commercial service nodes can disclose their channel lists and identity information to each other. Once a party breaches the rules, the cooperation of all channels can be terminated and the evildoing can be publicized, resulting in severe punishment.

We can limit the risk and loss of channel default to a certain range, so that it will not constitute a systemic threat to the entire settlement network.

8.3 Creation and default of channel credit currency

There are essentially only two ways to increase transaction throughput to different magnitudes: 1) Data and service centralization; 2) Allow temporary local data inconsistency; in principle, the express channel in the channel chain belongs to the latter, that is, deferring reconciliation time and reducing its frequency.

This will cause the expenditure of a channel at a specific time to exceed the initial locked limit, and its available balance will actually be negative. The balance in the channel includes the actually locked funds plus one party's debt to the other party. At this time, credit currency emerges, which is similar to the credit expansion caused by the bank's partial reserve system. The entire system creates a lot of new currency out of nothing!

We should pay great attention to the systemic risks of the payment network at this time. When certain channel parties have huge liabilities and cannot continue to repay, it will lead to a chain default effect. When it happens, the traditional financial crisis and economic collapse replay in the crypto currency system.

However, we guess that in a crypto currency system without a central bank (Lender of Last Resort), people will not borrow recklessly (or will not lend too much money to the other party who is obviously unable to repay), because everyone has to be responsible for their decisions eventually. People cannot transfer losses to the public through so-called quantitative easing policies or financial aid, which would lead to consequences ultimately borne by a large number of innocent people, especially the poor.

Another thing that needs extra inspection and precaution is centralized crypto currency exchanges. If we entrust crypto currencies to them in a traditional way for long-term management, they will become banks. We need to establish a sufficiently broad channel chain settlement network to replace the role played by banks in traditional economies. The important thing is to ensure that no one can steal our money, whether directly or indirectly. If there emerge restricted access banks and fractional reserve system in the network encryption currency system, this will be the biggest irony to everyone.

8.4 Centralized computing power, 51% attacks and guerrilla mining

Bitcoin wanted to avoid over-centralization of rights so that it abandoned the voting algorithm based on IP addresses and chose CPU computing competition. But it did not expect that the great advantages of customized chips (ASIC) would eventually cause serious centralization of computing power. In a field of free competition with open access, the gradual concentration of resources and personnel seems inevitable, because it has economies of scale, resulting in lower cost and intense competitive. In fact, we are not worried about the centralization of computing power itself, but about the plunder, fraud, and destruction springing from it. Accordingly, we are not really afraid of monopoly, but of restricted access.

Crypto currency has become more developed and standard. Almost all mining

computing power can be rented temporarily. This kind of guerrilla mining behavior will make some currencies with smaller market value and computing power face great computing power fluctuates and be at a serious risk of 51% attack, limiting the growth of new higher-quality currencies due to the risk of plunder, and making less-satisfactory ancient currencies of huge volume maintain its dominance. In theory, 51% attacks' targets are mainly centralized exchanges rather than individuals, because they have large enough funds for attackers to take a risk. If everyone uses the channel chain network to form an exchange, then this attack will lose its target.

In essence, we cannot completely solve this problem, we can only try to improve it to avoid it. There are several ways to help:

a. Invent new mining algorithms to avoid or defer the trend of concentration of computing power caused by the emergence of hardware devices

b. To become a miner, one has to mortgage appropriate amount of funds so that he would become a member of the community of shared interests, thereby reducing the motivation for evil

c. Let "real users" vote, "encourage" all honest miners to choose the "correct" fork

1) X16RS hash algorithm

X16RS is an upgraded version of the X16R algorithm. The basic principle is to randomly combine 16 different hash algorithms to compete the advantages of ASIC. The original hash combination is determined once for each block, and the improvement of X16RS is that for each hash step the algorithm is randomly selected, thereby increasing the difficulty of FPGA design and operation.

2) Selection of historical witness path

The 51% attack is one of the biggest obstacles to the large-scale use and popularization of crypto currencies. Especially for some newly created crypto currencies with small computing power and more effective mechanism design, the potential 51% attack risk will kill them in the cradle, and the market will be dominated by old inhibitants. We have to endure the unsatisfactory old currency for a long time, limiting full competition and free choice in the cryptocurrency market.

The PoW mining algorithm solves the most central problem in a shared currency system: who owns the next batch of newly issued currencies. In other words, the PoW competition mechanism determines the future.

The principle of the 51% attack is: one (or a very small number of) miner(s) secretly use(s) a computing power higher than all other miners' combined computing power to calculate a longer chain without anyone knowing it. After a period of time (a few blocks later) a longer chain is suddenly broadcast to the entire network, forcing everyone to abandon the recognized chain and switch to the attacker's fork, so that the attacker can withdraw the transaction that has taken effect (Double spending).

In other words, 51% attacks are essentially distorting history.

The only difference between an attacker and an honest miner is whether the mined block is immediately broadcast to everyone on the network. If no one conceals the newly mined block, there is no attack. Then the key to the problem is how to force everyone to broadcast blocks in time using a reward and punishment system, or to design a mechanism preventing the concealed blocks from being recognized by everyone.

To achieve this goal, we invented Proof of Work. The principle is to allow several accounts with the most amount of currency in the network to voluntarily sign the broadcasted block hash (the account with the most amount of currency will have the most incentive to maintain the system from being attacked), add up all balances in the accounts participating the proof, calculate a "proof value", and write it into the chain of public broadcast. Honest workers will try their best to broadcast their blocks to these witnesses of proof and obtain their proof certificates, so as to avoid the blocks they dug up from being rolled back and consuming computing resources in vain. We use "proof value" to represent the extent to which a block (actually a mining path) is broadcast ("the higher the proof value, the more adequate the broadcast"). When the attacker conceals the secretly dug chain and broadcasts it to the entire network after a period of time, all miners will compare the total "proof value" of the two forks and choose the one with the larger "proof value" to continue mining . Since the essential feature of the 51% attack is to conceal the block from everyone's knowledge, its "proof value" cannot exceed the fully broadcasted fork chain, making the attack unsuccessful.

In a nutshell, we have introduced a "proof value" to force all mining participants to broadcast newly mined blocks in time to prevent secret mining chain, thereby avoiding 51% attack with minimal system cost. The essence is: the PoW algorithm determines the future, but there may be many situations (forks) in the future. In order to avoid "future being distorted", the fund proof mechanism (PoS without currency reward) is used to ensure that the fork cannot be rolled back. In other words, PoW determines the future, and PoS determines history.

At this time, for an attacker to successfully attack, he must have more than 50% of computing power and more than 50% of funds theoretically. And if there is such a miner, the most rational beneficiary behavior is to maintain the safe operation of the overall system, rather than attacking it and harming its own interests. In fact, for a crypto currency that is fully known and used by everyone, in the long term, no single node will have more than 50% of the computing power and funds at the same time, and there will be no actual attacks.

However, this scheme still has two defects:

1. Witnesses of proof have no monetary rewards other than the shared interests of the community; 2. Witnesses need to stay online in real time, which may cause security issues of attack and theft.

The first point is actually an intentional design, because the witness and the mining pool can sign a private reward distribution contract to solve the problem more efficiently, so there is no need to consume consensus resources in a public agreement.

To avoid the second point requires the improvement and popularization of commercial signature hardware equipment. Since it is a hash signature for a block, it is a fixed and unchanging structured data. The hardware device can handle this situation well, and protect the security of the private key in real time online from external attack.

We use a uint16 (two bytes) data in the block header to store the "proof value":

```
type Block_v1 struct {
// Version fields.VarInt1
Height fields.VarInt5
Timestamp fields.VarInt5
```

```
PrevHash fields.Bytes32
MrklRoot fields.Bytes32
TransactionCount fields.VarInt4
// meta
Nonce fields.VarInt4 // random value of mining
Difficulty fields.VarInt4 // difficulty level of the target
// quantity level of proofs [represents the amount of funds to the power of 2^X]
WitnessStage fields.VarInt2
// body
Transactions []typesblock.Transaction
}
```

WitnessStage represents an index of 2.

3) Fork voting

In theory, this is not a technical insurance but a deterrent. The result of the vote is not mandatory, and the power to choose which fork is still in the hands of all miners. There is still a hypothetical premise here: the majority of miners and users are still honest and willing to cooperate to maintain the normal operation of the system.

We design that users who lock funds into the channel for a long time are the users who are the most effective and most connected users of the system, and give them voting rights in proportion to the channel funds. When the system suffers a 51% double-spending attack, the channels established before 10,000 blocks (about 35 days) are eligible to vote, and they will broadcast a voting transaction to all honest miners:

Anti-51% attack voting:

```
{
   // This algorithm is used to vote for chain, calling on miners to switch to a recognized chain
   // The number of votes is based on the locked funds in the channel, and the conversion starts when a certain number of votes is reached (decided by the miner)
   // to correct the fork experiencing 51% attack. It will not start at ordinary times and will take effect at critical moments.
   // Only valid channels older than 10,000 blocks (35 days) are eligible to vote
   kind: 19,
   // The block hash that the chain must contain is generally genesis block of the fork
   // And must be a hash that already exists in the history of this chain
   targetHash: Buffer.alloc(32),
   // List of channels participating in voting id
   channelIds: [
      232353,
      3847658374,
      874568376455,
   ],
}
```

Honest miners receive multiple transactions and accumulate the total funds as the number of votes. When the number of votes reaches the negotiated threshold, all miners will switch to the honest chain to continue mining.

The attacker considers that even if 51% attack is successful at a cost of a large amount of computing power, there may be other honest miners and users who jointly vote to revoke the fork resulting from the attack. As a result, only the attacker actually recognizes his fork and cannot seize any interests, so that they will not easily launch attacks. Fork voting is a deterrence protecting rights and interests, and it is a nuclear weapon that will cause self-damage. We should always be prepared for risk monitoring and voting, so that potential saboteurs can't succeed. But don't use this double-edged sword easily.

The above methods can reduce the risk of centralization of computing power, guerrilla mining and 51% attacks to a certain extent.

8.5 Extreme price fluctuations

Crypto currency should be a commodity first, and it will inevitably face relative price fluctuations, whether relative to fiat money or a basket of commodities. However, the function of currency as a settlement unit and value storage needs to be ensured by a fairly stable relative price.

The reasons for the extreme volatility of crypto currency prices are roughly as follows:

1) Speculative fanaticism

2) Algorithms for additional currency issuance can easily lead to hoarding and hype

3) A large amount of money is concentrated in the hands of a few people, and the price is manipulated

4)Technical or mechanism errors lead to confidence breakdown

Among them, the second and third can be circumvented by a more reasonable new currency issuance algorithm. When designing the three-stage currency issuance mechanism, we fully considered two goals: 1. Disperse the currency in the hands of a large number of individual users as much as possible; 2. The growth of currency conforms to economic laws to avoid excessive hoarding;

We have already explained the infeasibility of artificially controlling currency issuance and allowing the heads of certain institutions large discretion to maintain currency price. Regardless of the these infeasible methods, when a currency’s market value is large enough, we expect it to digest and offset local conflicting expectations and price fluctuations, as well as take advantage of futures market’s hedging. Like ordinary commodities, the risk of price fluctuations cannot be completely eliminated, and can only rely on rich and developed financial markets for hedging, although this will bring additional accounting costs.

All currencies are imperfect, but some of them can work more efficiently.

9 Technical Design Principles

9.1 Simple and intuitive

Financial system software cannot afford the losses caused by software vulnerabilities. Especially in an open and shared system, no one is responsible for your losses. The risk of "smart contracts" with potential vulnerabilities is very high, making them unable to be used in

financial transactions on a large scale. Another more serious problem of "smart contracts" is that ordinary people cannot understand the conduct matters in the contract code, and they need to always seek help from professional coding technicians. This greatly limits its application scenarios and makes it unaccessible to ordinary people.

For an open shared financial software system, we need a set of standardized, human-readable instructions so that users without technical background can easily understand the details of the agreement and contract without any potential bugs. Particularly, the degree of intelligibility is the key.

9.2 Compact data and efficient execution

We must balance versatility and efficiency of the protocol, and even consider saving each byte of space and the time consumption of each instruction. The elegance of program module system design should give way to efficiency in the core part.

The Lisp machine with incomparably elegant language design and the SmallTalk operating system that turns everything into objects have both failed. History has chosen the C language and UNIX machine that are full of trade-offs, compromises and "filthy implementations". The reason behind it lies in the first principle of economics: cost and efficiency.

9.3 The scale of public ledger data is controllable

Increasing the size of the block space and the frequency of block generation to an extent that ordinary equipment cannot support will lead to the centralization of the power of bookkeeping, thereby endangering the security of the entire system.

We need a controllable data growth plan and a controllable scale of transaction processing computing resources to ensure that the processing and recording of the ledger is sufficiently decentralized. The technical indicators of its mainnet do not need to be permanent, but should probably be limited to the range that can be processed by a mid-level mainstream computer that can be afforded by ordinary households.

9.4 Signature stripping and data compression

After a long enough period of time (such as one year), a block cannot actually be returned, and historical transactions have become irrefutable. At this point, we should support the separation of signatures that occupy a large part of the block data, compress and store the transaction data, so as to support updating and querying the ledger on devices with lower hardware performance or storage space.

For longer-term historical data, all "status data" of general ledger nodes can be snapshotted at a certain time through the data consistency algorithm every month or every year, and be written into the mainnet for public recognition. The newly added general ledger node can start to synchronize a small part of the subsequent block data from a snapshot at a certain moment in the middle as needed, and abandon the traceability and verification of all historical transaction history, thereby greatly reducing the load and speeding up the available time.

10 Conclusion

We have proposed a crypto currency system that incorporates issuance, circulation and storage functions, and can be used for large-scale payment and real-time settlement. First, the basic principles and technical implementation details of the channel chain settlement network are discussed. We believe that a public ledger that includes interest incentives as the final arbitration and liquidation guarantee, with a strict punishment mechanism for dishonesty, can support massive payments, thereby greatly reducing transaction and trust costs. Such a system is characterized by a market-oriented new currency issuance mechanism that is very in accordance with economic laws, as well as strict fund security and real-time deposit guarantees, and it does not rely on any central institution. Taking division of labour and control of rights in a developed business environment into account, we design various transaction categories and simple technical agreements without potential bugs in this paper. In addition, it discusses currency issuance rules, solution of Bitcoin transfer, the importance of privacy and the protection of financially disadvantaged groups. It further discusses related potential risks and solutions to prevent them. So the framework introduced in this white paper contains the general rules and incentive measures required for a fair, efficient, and trust-free crypto currency system that supports issuance, circulation, value storage, large-scale payment and settlement.

Appendix

1. X16RS hash algorithm

```
function X16RS_HASH( prevhash_buf, stuff_buf ){
    function SHA3_256(a){ return crypto.randomBytes(32) } // suppose
    var hashfuncs = [ // suppose
        function Blake(a){ return crypto.randomBytes(32) },
        function BMW(a){ return crypto.randomBytes(32) },
        function Groestl(a){ return crypto.randomBytes(32) },
        function Jh(a){ return crypto.randomBytes(32) },
        function Keccak(a){ return crypto.randomBytes(32) },
        function Skein(a){ return crypto.randomBytes(32) },
        function Luffa(a){ return crypto.randomBytes(32) },
        function Cubehash(a){ return crypto.randomBytes(32) },
        function Shavite(a){ return crypto.randomBytes(32) },
        function Simd(a){ return crypto.randomBytes(32) },
        function Echo(a){ return crypto.randomBytes(32) },
        function Hamsi(a){ return crypto.randomBytes(32) },
        function Fugue(a){ return crypto.randomBytes(32) },
        function Shabal(a){ return crypto.randomBytes(32) },
        function Whirlpool(a){ return crypto.randomBytes(32) },
        function SHA512(a){ return crypto.randomBytes(32) },
    ]
    var hashloopnum = hashfuncs.length
    , stephashs = []
```

```
    for(var i=0; i<hashloopnum; i++){
        var funcidx = prevhash_buf.readUInt8(31) % hashloopnum
        // console.log(funcidx)
        prevhash_buf = stuff_buf = hashfuncs[funcidx](stuff_buf)
        stephashs.push(stuff_buf)
        // console.log(stuff_buf.toString('hex'))
        // console.log('----')
    }
    stuff_buf =    Buffer.concat(stephashs, hashloopnum*32)
    return SHA3_256(stuff_buf)}
```

2. Block reward

```
function calcBlockCoinBaseReward(block_height){
    var rwdns = [1,1,2,3,5,8,8,5,3,2,1,1] // length must uneven number
    , frix = parseInt(rwdns.length / 2)
    , pos = parseInt(block_height / (10000*10)) // almost 1 year
    // console.log(frix, pos)
    if(pos < frix){
        return rwdns[pos]
    }else if(pos < frix+((frix+1)*10)){
        return rwdns[frix + parseInt((pos-frix)/10)]
    }else{
        return rwdns[rwdns.length-1]
    }
}
```

3. One-way transfer of Bitcoin to issue new currency and lock-in cycle

LV:	1	BTC:	1,	1	HAC:	1048576,	1048576	LOCK:	1024w, 19.69y, 1024
LV:	2	BTC:	2,	3	HAC:	524288,	2097152	LOCK:	512w, 9.846y, 1024
LV:	3	BTC:	4,	7	HAC:	262144,	3145728	LOCK:	256w, 4.923y, 1024
LV:	4	BTC:	8,	15	HAC:	131072,	4194304	LOCK:	128w, 2.461y, 1024
LV:	5	BTC:	16,	31	HAC:	65536,	5242880	LOCK:	64w, 1.230y, 1024
LV:	6	BTC:	32,	63	HAC:	32768,	6291456	LOCK:	32w, 0.615y, 1024
LV:	7	BTC:	64,	127	HAC:	16384,	7340032	LOCK:	16w, 0.307y, 1024
LV:	8	BTC:	128,	255	HAC:	8192,	8388608	LOCK:	8w, 0.153y, 1024
LV:	9	BTC:	256,	511	HAC:	4096,	9437184	LOCK:	4w, 0.076y, 1024
LV:	10	BTC:	512,	1023	HAC:	2048,	10485760	LOCK:	2w, 0.038y, 1024
LV:	11	BTC:	1024,	2047	HAC:	1024,	11534336	LOCK:	1w, 0.019y, 1024
LV:	12	BTC:	2048,	4095	HAC:	512,	12582912	LOCK:	0w, 0y, 512
LV:	13	BTC:	4096,	8191	HAC:	256,	13631488	LOCK:	0w, 0y, 256
LV:	14	BTC:	8192,	16383	HAC:	128,	14680064	LOCK:	0w, 0y, 128
LV:	15	BTC:	16384,	32767	HAC:	64,	15728640	LOCK:	0w, 0y, 64

LV: 16	BTC:	32768,	65535	HAC:	32, 16777216	LOCK:	0w,	0y, 32
LV: 17	BTC:	65536,	131071	HAC:	16, 17825792	LOCK:	0w,	0y, 16
LV: 18	BTC:	131072,	262143	HAC:	8, 18874368	LOCK:	0w,	0y, 8
LV: 19	BTC:	262144,	524287	HAC:	4, 19922944	LOCK:	0w,	0y, 4
LV: 20	BTC:	524288,	1048575	HAC:	2, 20971520	LOCK:	0w,	0y, 2
LV: 21	BTC:	1048576,	2097151	HAC:	1, 22020096	LOCK:	0w,	0y, 1

4. Block diamond hash algorithm

```
function hash17diamond( buffer ){
    // console.log(str.length)
    if (buffer.length !== 32){
        throw new Error("buffer must be hash256")
    }
    let stuff = '0WTYUIAHXVMEKBSZN'
    , total = 16
    , hhlen = stuff.length
    let diamond = []
    , fv = 0
    for(let step=0;step<total;step++)
    {
        let i = step * 2
        , n1 = buffer[i]
        , n2 = buffer[i+1]
        fv = (fv + n1 + n2) % hhlen
        diamond.push( stuff.charAt(fv) )
    }
    return diamond.join('')}
function checkDiamond(stuff) {
    let chars = '0WTYUIAHXVMEKBSZN'
    if(stuff.length == 16 && stuff.startsWith('0000000000')){
        var sarys = stuff.substr(10).split('')
        , first = true
        // console.log(sarys)
        while(true){
            var l = sarys.shift()
            , idx = chars.indexOf(l)
            if(!l){
                return first ? false : true
            }
            if(idx==-1){
                return false
            }else if(idx==0){
                if(first){
```

```
                    continue
                }else{
                    return false
                }
            }else{
                first = false
            }
        }
    }else{
        return false
    }
}
```

Reference

[1] Adam Back,"Hashcash-A Denial of ServiceCounter-Measure",http://www.hashcash.org/papers/hashcash.pdf, 2002.

[2] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf, 2008.

[3] Ludwig von Mises, "Theory of Money and Credit", https://mises.org/sites/default/files/The%20Theory%20of%20Money%20and%20Credit_3.pdf, 1912.

[4] Friedrich August von Hayek, "Individualism and Economic Order", http://www.library.fa.ru/files/Hayek-Individualism.pdf, 1948.

[5] J.Huerta de Soto, "Money,Bank Credit and Economic Cycles", https://mises.org/sites/default/files/Money_Bank_Credit_and_Economic_Cycles_De%20Soto.pdf, 1997

[6] Joseph Poon, Thaddeus Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments", https://lightning.network/lightning-network-paper.pdf, 2016.