PATH Protocol

Layer2 on BTC for web3

Layer2 Unlimited scaling scheme based entirely on Bitcoin consensus

1 Foreword

1.1 Legitimacy

In his March 2022 article, Ethereum founder Vitalik Buterin mentioned that Bitcoin and the Ethereum ecosystem spend far more on cybersecurity (i.e., PoW mining) than all other aspects combined. On average, tens of millions of dollars are paid to miners each day in block rewards and transaction fees. In contrast, much less is spent on research and development and other ecological improvements. But it is precisely because of this ability to focus on such large-scale capital investment and distribution through decentralized social organization that this power is so powerful, he calls it legitimacy.

In fact, the existence of this kind of legitimacy is not rare in the history of the development of human society up to now. In a social and political system like a country, from the imperial power and the government to the organs and departments at all levels under the leadership of the government, some kind of legitimacy has continued, and this legitimacy has influenced the behavior norms of every social individual, even playing a greater role than the law to a large extent.

The charm of the blockchain social legitimacy led by the Bitcoin network lies in that it is the first technology-driven belief force in human history.

But on the whole, we think Vitalik overemphasizes the legitimate influence brought by technology itself, while ignoring the power of belief accumulated by a technology in the process of human social dissemination. In fact, PoW miners can always find a choice with higher returns in the market if they are purely from an

economic perspective, and at the same time, from the perspective of investors in the financial market, The far-reaching impact of a brand can never be ignored, because it will continue to rank high in your mind as your default option.

On the other hand, what Vitalik deliberately ignores in his article is that even the Ethereum network is still far from the Bitcoin network in terms of legitimacy.

1.2 Status Quo

Discussing the legitimacy of a blockchain infrastructure (as opposed to a financial commodity) as a technical choice is, in fact, much like the biological significance of pedigree. Over the past eight years, Ethereum has played a huge role in shoulting the expectations and responsibilities of a decentralized society. Firstly, Vitalik was the first person to come up with smart contracts. Secondly, it is necessary to ignore that he himself was deeply involved in the early Bitcoin community and held high the banner of Satoshi Nakamoto's concept throughout the development of Ethereum.

It can be said that the past 8 years have been the 8 years of Ethereum, which has been leading the development of the industry, and proposed new directions at important nodes, seeking to solve new problems in technology. DeFi is the important motor in this bull run in 2021, and Ethereum is the foundation of DeFi, and it doesn't matter that Bitcoin itself didn't play much of a role in this bull run, it still gained a lot as digital gold. But when the profit-making funds are withdrawn from the market, leaving the industry with a pessimistic mood, especially when most practitioners are holding high the flag of Web3 Build, we find that the entire technical soil seems to deviate from the original belief, that is, decentralization.

There have been a lot of thinking and practice based on the expansion of Ethereum, of which the most in line with the decentralized concept of Bitcoin is

sharding and Plasma two technologies, but for whatever reason, both have not been well developed or applied, on the contrary, relatively more centralized two-layer scheme has become the "mainstream". Including the transformation of Ethereum to PoS, no matter how you look at it, capital has played a huge role in these important choices, and this is the biggest problem currently encountered in the blockchain soil, the brave eventually became the dragon. From a technical point of view, sharding and Plasma are both promising scaling solutions, and the reason to stop the development of these technologies is not the technology itself.

This situation has led people to return to the vision of the Bitcoin network for a new way out.

1.3 Rise of Bitcoin ecology

No one expected 2023 to be the year of the Bitcoin ecosystem, but it looks like it is already happening. The Ordinals protocol in the first half of the year, to the release of the Taproot asset protocol on October 19, and then the Bitcoin network expansion basic technology such as RGB protocol, Bitcoin smart contract Rootstock and Bitcoin Stacks, they began to be paid attention to, we found that the original based on the Bitcoin network can do more, And there is no ethereum ecological cancer.

These Bitcoin network ecological infrastructures are not new; they all benefited, to a greater or lesser extent, from the SegWit fork in November 2017. But it took until 2021 for Taproot Assets to light the fuse, which is pretty amazing, five years for the Bitcoin ecosystem to not compete with Ethereum, until 2021, when everything naturally happened. Just like the initial IPO in 2015, but the difference this time is that everything is decentralized, and we feel like we are back to the initial state of community-driven ecology.

A more pure technical soil, a more transparent governance society, and a decentralized network that theoretically can fully support the latest interactive experience of the Internet can undoubtedly represent the real Web3 revolution.

Of all the known solutions for scaling the Bitcoin network, PATHBTC is one of the most competitive, and it is technically open and scalable enough.

2. PATH Protocol solution

2.1 Cross-chain

In a decentralized world, various crypto assets in the blockchain industry form the cornerstone of the Web3 ecosystem, so asset cross-chain is a fundamental capability and one of the driving forces of PATH. According to the plan, this cross-chain protocol is named "PathBridge". In this chapter, we will choose the most reasonable technical solution to build this cross-chain network.

Mainstream cross-chain solutions

Cross-chain has always been understood in two different ways. One refers to the primary exchange of assets between two blockchain systems, and after the transaction takes place, the total circulation of related assets in both systems remains unchanged. The second is the transfer of assets between two different blockchain systems. After the transfer, the circulating amount of transferred assets corresponding to the two blockchain systems changes, but the sum of their circulating amount remains unchanged. PathBridge's cross-chain approach refers to the latter. Since Ripple proposed the InterLedger protocol in 2012, the blockchain industry has dozens of different technical solutions to achieve cross-chain asset transfer types, but in terms of the essence, these solutions can be divided into two categories according to the confirmation method, notarization and relay.

A. Notarization scheme

The proposal calls for a group of trusted organizations within the scope of decentralization to oversee two different blockchain systems

Control, and correctly complete predefined transfer operations when an asset lock-up is detected. From the implementation details, it has two kinds of technical implementation, multi-signature mechanism and distributed signature mechanism.

The multi-signature mechanism is that multiple notary nodes supervise the locked account at the same time, and only after providing a certain number of correct signatures can the account be operated to complete the cross-chain transaction. This way requires that both blockchain systems support the multi-signature mechanism or have the function of smart contract. This is the model used by ChainBridge and PalletOne, as well as Ren.

The distributed signature mechanism uses the linear feature of elliptic curve arithmetic to distribute the private key of the asset locked account to different notary public using threshold signature technology. A certain number of notary public needs to gather to complete the signature of cross-chain transactions. This method requires that both blockchain systems have some fixed mode of elliptic curve signature account system, such as ECDSA or EDDSA. Wanchain(Fusion) and tBTC use this model.

The advantage of this scheme is that the notary can access the blockchain system in the form of a client, and there is no intrusion to the original chain. This method is technically inclusive and flexible, and can adapt to most of the current block chain systems. The disadvantage is that it requires notarization by a third party other than the two chains, which adds one more possible risk point.

B. Relay scheme

The two blockchain system chains or smart contracts that use the relay scheme have lightweight block mappings of each other, and can verify the validity and authority of each other's blocks through these mappings. When the two parties

create blocks, any node can submit the block mapping information to the smart contract of the other party, and the smart contract of the opposite party simulates the block verification process. In this way, other contracts deployed on the two sides' blockchains can obtain the status information of the other side's blockchains through this mapping, which is used to verify whether the smart contract on the other side has truly received the locked funds, and then proceed with the next transfer operation. BTC-relay and Cosmos' Hubs and Polkadot's RelayChain are implemented in this way.

The advantage of this scheme is that no third party is needed and any individual can submit block mapping information to complete the relay. But the disadvantage is also obvious, that is, it is invasive, which either requires that the relay scheme is considered and the interface is set aside when the chain is designed, or it requires that there must be a smart contract. Moreover, each block needs to be submitted to the other party after the generation, so that in addition to consuming storage resources, for the chain that needs to pay fees, it will consume a lot of fees. At the same time, if a chain changes the consensus protocol, then the opposite relay needs to be rewritten. Another problem with the relay scheme is that if the original chain being relayed has a long fork, while the chain carrying the relay mapping has already been identified and released assets, it will be in a state of fragmentation, leading to unpredictable results.

Design ideas

Since PATHBTC is preemptive, the following is given before introducing the technical solution

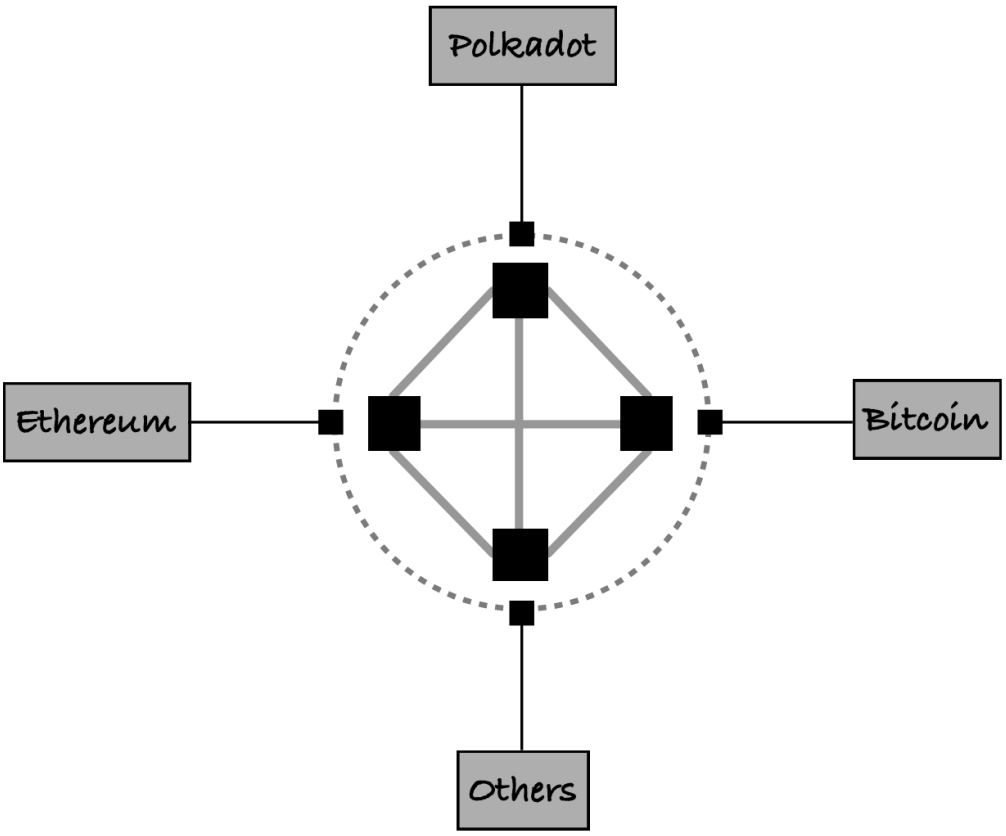
These design intentions:

1. A decentralized scheme that is not significantly less secure than the blockchain system being straddled.

2. It has enough feasibility and usability, not just theory and experiment.
3. Has broad applicability and can provide cross-chain functionality for most mainstream blockchain systems.
4. Cross-chain across homogeneous (FT) and non-homogeneous (NFT) assets can be accomplished.
5. Can eventually migrate to the cross-chain framework of the PATHBTC mainnet protocol network.
6. Provides privacy protection for the owners of homogeneous assets and non-homogeneous assets and related transactions

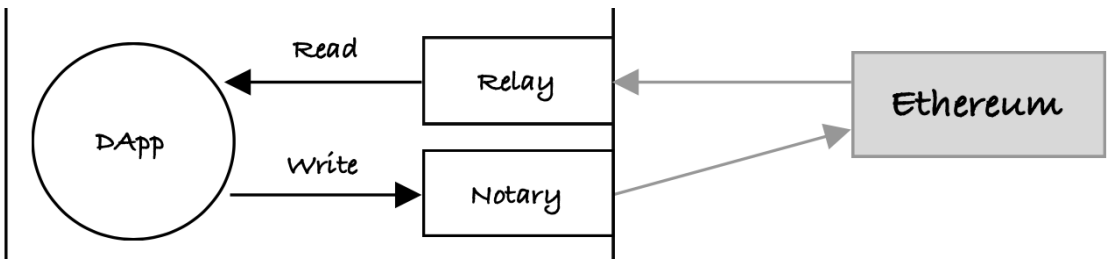
I can.

7. For the time being, the underlying system does not introduce another homogenized asset for consensus maintenance.



8. The design is simple and intuitive, easy to implement and maintain.

Since PathBridge will eventually migrate into the PATH Protocol Protocol network as the infrastructure for all virtual assets, and the PATH Protocol protocol network is independent of other blockchain systems, So PathBridge should interconnect with third-party blockchain systems independently of each other.



In the PATHBTC Protocol network, because the protocol is cross-chain from the start, it is a natural choice to put Repeaters from various other blockchain systems into the PATH Protocol's application account. Due to the diversity of other blockchains and the unique nature of the PATHBTC protocol, it is difficult to link the PATHBTC protocol to other blockchains in the form of Repeaters, so there must be a notary mechanism to issue instructions to other blockchain systems. The PATHBTC protocol must be a hybrid system with both relay and notary mechanism. The state of other blockchain systems is open to the PATHBTC protocol through the relay, and the application of the PATHBTC protocol network can query the state of other blockchain systems at will. And the cross-chain instructions of the PATHBTC protocol are issued to other blockchain systems in the form of notarization. As long as the notary node of the PATHBTC protocol is secure enough, it can be used in the least cost way to drive other blockchain systems.

Since the cross-chain function of PATHBTC ultimately adopts a hybrid mode, the cross-chain components that are used before the release of the PATHBTC network need to be notarized first in order to facilitate subsequent migration. Cross-chain notary nodes currently have no native state, and they can continue to exist as genesis nodes when the PATHBTC network is launched.

Implementation Method

1. Decentralized notary system

Notary nodes are some 24 hours running on the Internet network service program, all notary nodes constitute the notary committee. The committee was originally composed of several genesis nodes, and other nodes that want to enter the committee need to be approved by more than 2/3 of the members of the vote. A two-thirds majority vote of the committee allows a node to leave.

The execution of each cross-chain directive is a committee vote, and when the number of committee nodes is small, it needs more than 2/3 of the total number

of committee nodes to vote for it to be effective. When the number of committee nodes is too large, VRF algorithm will be used to generate a temporary committee for each cross-chain directive, and N rounds (N>=1) committee voting will be conducted in order to facilitate the final determination of the result.

Since PathBridge and PATHBTC do not initially issue homogeneous assets, for each cross-chain call will be charged for the corresponding asset as the execution of cross-chain transaction fees, and these fees will be transferred to the notary node account that initiated the vote when the cross-chain operation is completed.

2. Fairness and security

As a notary commission, there are two main challenges in security, selfish behavior and malicious behavior. Selfish Behavior

Behavior is an inactive act of providing notary services, which will cause the effective service resources of cross-chain services to fall. Malicious behavior is the behavior of making a false or invalid vote, such a node will be locked

The asset security factor is reduced. PathBridge takes the following steps to prevent this.

3. Genesis Node **KYC**

In order to reduce the probability of node collusion in the initial stage, PATH will carry out strict KYC verification for the initial limited founding nodes, which can ensure the independence of social relations of Genesis nodes to a certain extent, eliminate malicious behaviors targeting the initial system operation and ensure the benign development of PATH.

4. Pledge money

Each node that becomes a committee, if it wants to act as a cross-chain verifier of an asset, needs to pledge the asset involved as a deposit, and the node

must continue to operate for at least 1 year before it is allowed to exit. Normally, the margin is returned to the node when the node exits. However, in the case of the node's evil behavior, the committee member may decide to confiscate some assets by a 2/3 vote and may vote to expel the evil node. The pledged money of the expelled evil node will continue to be pledged within the previously set time until it is returned at maturity.

5. VRF

When the number of nodes in the committee is large, the temporary committee formed by the VRF algorithm will lock and release the assets every time the cross-chain instruction is operated.

Under normal circumstances, the voting weight of each node is the same, and PathBridge will take a fair cut of the issuing fee according to the notary's row, but in the following two cases, their voting weight may be reduced.

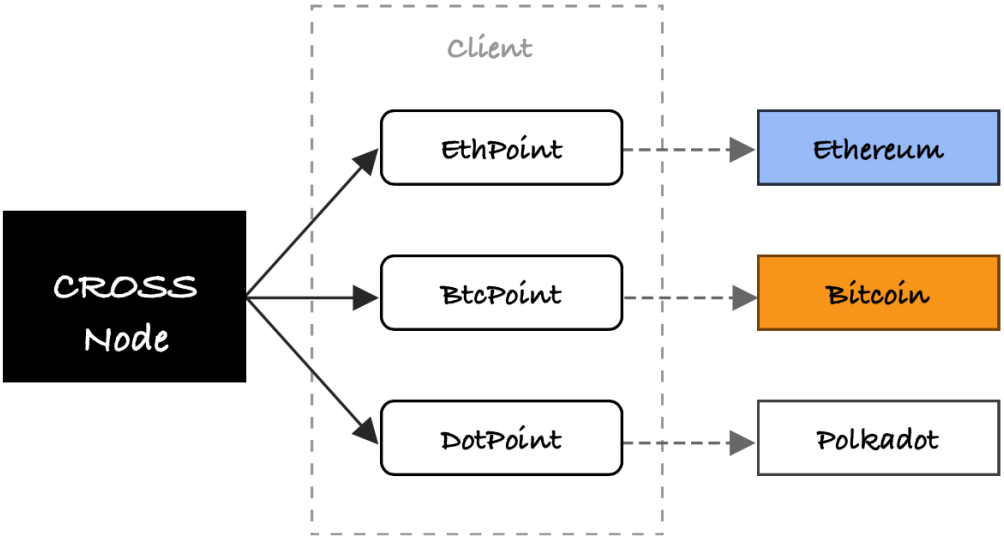
6. Committee oversight

The committee periodically reports evil nodes, and if a node is reported by more than 1/2 of the committee nodes in a cycle, it is considered evil, and the weight of the node's vote will be reduced.

7. **PathBridge Node**

PathBridge nodes connect to different blockchain systems via components with the suffix Point, such as Ethereum via EthPoint, Bitcoin via BtcPoint, and Polkadot via DotPoint. The Points all have standard interfaces to be called by the PathBridge Node. The Point component is the client of each blockchain system.

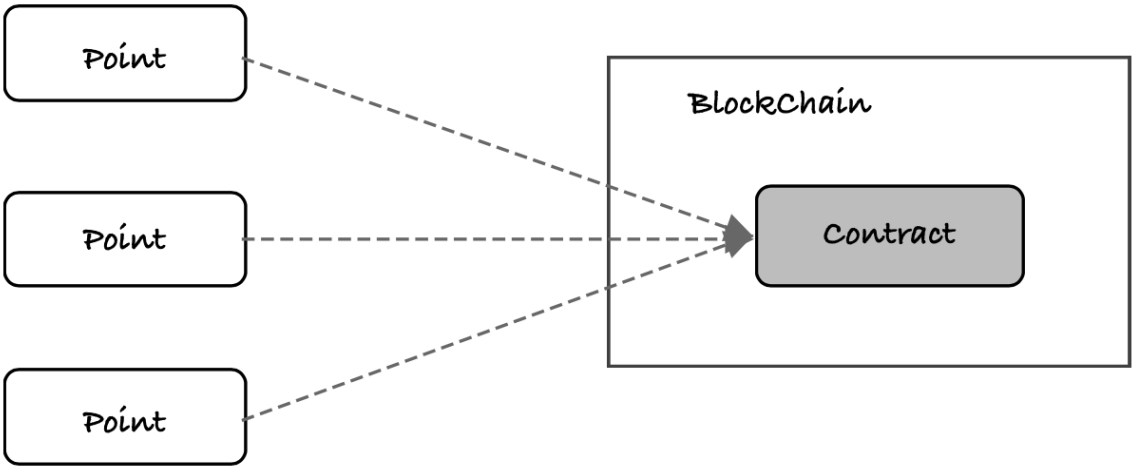
PathBridge reads the chain information through the Point and initiates cross-chain



instructions.

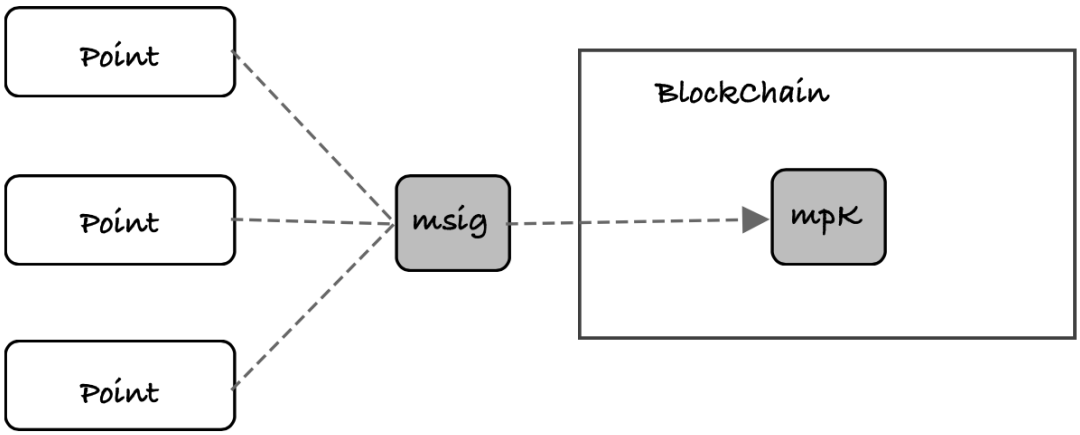
8. Multiple signatures

For blockchain systems with smart contracts, where funds are locked in smart contracts, the Point components of the different PathBridge nodes in the committee will register in a particular smart contract and make contract calls according to



decentralized rules, in this form implementing something like multiple signatures.

To solve the problem of multiple node voting consuming too much Gas, PathBridge introduces an aggregate signature mechanism. When the verification node detects a new cross-chain transaction, it immediately broadcasts its signature for that transaction. Any node can collect the signatures of the other nodes in the committee and then aggregate them into a call parameter to submit to the smart



contract.

For a blockchain system without smart contracts, if the account and signature system supports multiple signatures, the Point component of the different PathBridge nodes in the committee will access the fund lock account through the multiple signature function.

However, for the notarization mechanism implemented by the multi-signature method, generally when the number of signatures changes, the public key to lock the account will also change. At this time, it is necessary for the notary node of the committee to coordinate with consensus to produce another locked account to receive the assets of the original locked account before completing the membership or withdrawal of members.

2.2 Scalability

The PATHBTC protocol is a decentralized blockchain protocol that is the PATHBTC Decentralized Network

Nodes in PathBTC maintain ledger data and standards and rules for interacting with each other. These nodes operate independently in a network environment and exchange information, forming a decentralized infrastructure that is scalable, efficient, and immutable.

The PATHBTC protocol is the cornerstone of the web3 decentralized world after BTC expansion, so it needs to have the following features and challenges in addition to the basic public blockchain features:

A. Consensus of all kinds without native homogenized tokens and natural cross-chain support. In this way, various existing blockchain assets can be better utilized, and users will focus on the ecology and ecological assets on it, rather than on the assets defined by PATHBTC itself.

B. Support elastic expansion, support the sustainable development of the entire PATHBTC needs to have the scalability of computing, storage, bandwidth and other aspects, in order to cope with the future growing business needs.

C. Easy to develop complex decentralized applications, the future PATHBTC ecosystem of business complexity may be high, and in the traditional virtual machine form of the application framework, the architecture of such a complex decentralized application is very difficult, so PATHBTC needs to be able to support and simplify the development of complex decentralized applications.

D. Supporting complex assets. In addition to homogeneous assets, EPOCH ecology will carry out a variety of production activities around non-homogeneous assets, and asset types will become rich, so it is very important to support complex asset types in a native unified form. User privacy protection, privacy is

everyone's right, is a symbol of people's independence and freedom, PATHBTC needs to support privacy-oriented assets and transactions. Cross-chain has been discussed and studied in the previous chapter, and proposed the implementation of PATH cross-chain, the remaining several items will be discussed in the next.

3.3 Scheme Summary

A. Elastic expansion

Since the emergence of blockchain systems, the scalability of TPS has been a very critical and difficult challenge, and with the development of blockchain, storage and bandwidth will also face scalability challenges. All the time, various technical solutions have been proposed to solve the scalability problem, challenging the trilemma proposed by Vitalik (the inevitable tradeoff between decentralization, scalability, and security). These solutions fall into two main categories, Layer1 and Layer2 solutions. Layer1's solution mainly solves the scalability problem through the blockchain system's own consensus mechanism, while Layer2 achieves this by shifting transaction processing from a heavier on-chain system to a lightweight off-chain system. Our current focus is on Layer1's technical solution.

From the existing solutions, Layer1 solutions can be divided into sharding and directed acyclic graph (DAG) two categories. They all have their own characteristics.

B. Sharding

The main approach of the sharding scheme is to split the single chain structure of the blockchain into multiple chains, and each chain processes different transactions in parallel, so as to achieve the allocation of computing, storage and bandwidth resources. According to the granularity of the splitting, it can be divided into two ways: grouping and lattice.

The grouping method is to divide accounts into several groups, each group corresponds to a set of servers and an independently extended chain, and maintain several accounts. Sending transactions between accounts within a group is the same as on a separate blockchain, but sending transactions between accounts in different groups requires more work. NEAR and Etheruem 2.0 are doing this, and Polkadot and Cosmos are doing something similar, except they sharding by application rather than account. The problem with the grouping approach is that it often requires a separate backbone to handle transactions across shards and maintain consistency across shards. Therefore, this independent main chain becomes the bottleneck for cross-shard transactions.

The lattice structure is the ultimate form of shard splitting, in this way, each account has a separate chain, and transfers between different accounts are cross-chain transfers. The lattice approach simplifies complexity by separating the transaction into two transactions. The operation of each account is independent, so it can provide very high throughput. Nano and Vite do this. But Nano uses the ultimate consistency algorithm, it does not support smart contracts and decentralized applications, it is prone to fork and state rollback problems. While Vite introduces smart contracts and solves the rollback problem, it introduces a main chain called snapshot chain, just like the grouping scheme. In fact, Vite uses the snapshot chain to confirm, which weakens the advantages of the lattice fragment structure in a sense, and the throughput ultimately depends on the performance of the snapshot chain.

C. Directed Acyclic Graph **(DAG)**

A DAG is essentially not a chain but a graph. In general, each block of a blockchain has only one parent block to reference, which results in multiple blocks forming a chain structure. A DAG, on the other hand, extends the number of references to a parent block so that it can refer to multiple parent blocks. Because the DAG block structure is no longer a chain structure, it is possible to create and link

blocks concurrently, achieving huge throughput. Spectre and IOTA both take this approach. DAG is essentially a rather chaotic partial ordered structure, and the validation of blocks usually needs to be realized by some statistical rules that can cause the convergence of the structure, so although the throughput of DAG is huge, the validation speed is relatively slow, and it is ultimately consistent, and the compatibility of smart contracts is not that good

Ok?

D. Complex decentralized applications

Current decentralized applications come in two forms, one of which exists in the blockchain system in the form of smart contracts

The other is in the form of a child chain.

Smart contract, under normal circumstances, the blockchain consensus system must be modified after the completion of hard fork, that is, update all the network nodes of the protocol, this process has a certain complexity, and accompanied by a certain risk. The smart contract allows users to customize the consensus protocol without hard fork.

Smart contracts can be traced back to 1995 by cryptographer Nick Szabo. Smart contracts are computer programs that enforce the terms of a contract. In the blockchain world, a smart contract usually represents a special account that can take incoming transactions as input and run a pre-programmed program code to perform operations on the account, including modifying the account data and transferring assets. A smart contract is a special kind of decentralized application, which is driven by a virtual machine running on the blockchain system, and expresses a certain meaning by modifying account-related data.

The launch of Ethereum introduced the concept of smart contracts into the blockchain system for the first time, and this attempt turned the blockchain system into a decentralized operating system. Various decentralized applications and

ecosystems continue to emerge on Ethereum; For example, the vast majority of DeFi applications are built on Ethereum, which has led to the prosperity of decentralized transaction ecosystems. Later, many public chain systems have provided smart contract functions, and decentralized metaplasia has also been developed on these public chains, such as EOS, TRON, NEO, PATH and other public chain systems.

However, this form of decentralized application has some limitations, because the smart contract depends on the status of the account to express the meaning, so every modification of the status needs to be confirmed the consistency, and the method used to achieve the contract and the data structure have certain limitations, which limits the available means to optimize the system performance. In the use of a more tortuous way to achieve complex applications, it is also easy to produce a variety of vulnerabilities. And because the virtual machine running the smart contract is part of the system consensus, it can be cumbersome to maintain and upgrade it. In addition, smart contracts usually require users to make transactions to drive them, and cannot run a task on their own.

E. Subchain

With the development of cross-chain and elastic scaling technologies, the formula for using child chains as decentralized applications began to emerge. In this way, the decentralized application is realized in the form of an independent public chain, and connected to each other through the cross-chain mechanism, which can easily support more complex and efficient application forms. Polkadot aggregates public chains of different consensus mechanisms to form a large ecosystem through the Relay Chain, whose child Chain is called Parallel Chain. Cosmos, through Cosmos Hub, can link the subchains of Tendermint Consensus, and its subchain is called Zone.

There are two problems with decentralized applications in this way. First, though, these projects all provide SDKS

Designed to reduce the burden on developers, the development of a complete public chain still has a high threshold, unlike the development of smart contracts that only need to care about how to implement the business. In addition, it still has a main chain that acts as the communication between different subchains, which limits the performance of decentralized applications to some extent.

F. Privacy protection

Identity, transactions, and status are the primary privacy targets of decentralized systems. Although the accounts of the blockchain system achieve a certain degree of identity hiding through pseudonyms and broadcast means, since all data of the public chain system is unconditionally disclosed to the outside world, and most public chains cannot hide the details of transactions, attackers can always aggregate different addresses into the identities of a small number of real objects through some statistical means. In a sense, most blockchain public chain systems are not privacy-protected, and in many scenarios such blockchain systems are not applicable. Since the degree of privacy of the transaction represents the degree of privacy of identity and status in a certain sense, we classify the transaction according to the means of privacy. There are several blockchain systems that achieve some degree of privacy by obfuscating or hiding some or all of the information about the transaction.

Here are four privacy algorithms based on the comprehensiveness of the hidden information.

Mixing information on both sides of a transaction **--** mixing money

Mixing coins involves shuffling the inputs and outputs of multiple transactions in a centralized or decentralized manner without changing the result of the change in the final global state. Dash, for example, uses a decentralized coin mixing approach to achieve privacy protection. Mimblewimble takes a similar approach to identity concealment. This scheme has some problems, for example, it relies too much on special nodes to carry out coin mixing operation. The node responsible for coin mixing knows the transaction information of all parties, so there is the possibility of information leakage, and it needs to wait for other transactions, and the input and output after coin mixing may be related to a certain degree.

Obfuscating initiator information - ring signature

Ring signature is achieved by the transaction originator putting unrelated accounts into the same transaction as the transaction originator. It is difficult for transaction validators to find out who is the real originator among multiple fake originators. Ring signature is a cryptographic algorithm. The CryptoNote protocol used by Monero hides the originator in a transaction in such a way. This scheme is not resistant to dust attacks, and if the privacy of other accounts participating in the ring signature is made public, then the real originator will be exposed.

Hide all information of the transaction - zero knowledge proof

Zero-knowledge proof is a method that allows a third party to verify that the information character meets certain assumptions without unbiased disclosure of the original information. Zero-knowledge proof uses algorithms such as homomorphic encryption and secure multi-party computation. The blockchain system can use zero-knowledge proof to hide the details of the transaction, including the location of both the sender and the receiver and the transaction content, but it does not prevent the ledger generator from verifying the validity of the transaction. Currently, the Blockchain

ZeroCash uses zk-SNARKs to completely hide the transaction information, Monero uses Bulletproofs to check the range of input-output asset amounts of the transaction, PATH realized the hiding of the transaction amount and address of homogeneous (FT) and non-homogeneous (NFT) assets through SuperZK, and enabled smart contracts to hold and allocate encrypted assets. In a sense, the zero-knowledge proof algorithm is the most private of the currently available privacy schemes, it can achieve the entire transaction almost all information obfuscation, and lead to the account's own assets are also private. However, the biggest limitation of zero-knowledge proof is that its efficiency of generating proof is very low. When the more knowledge needs to be proved, the more computation needs to be done.

Implementation Method

By studying and summarizing various current blockchain protocols, through some assumptions and new decentralized algorithms, the PATHBTC project team finally proposed a decentralized public ledger protocol PATHBTC. This decentralized ledger protocol scales performance, throughput, and storage capacity while maintaining security.

A. The network environment assumes that

1. The ratio of the number of correct nodes to the total number of nodes at any one time must be greater than S.

2. When a correct node receives a message, within time T, all correct nodes will receive the message.

B. Dot-matrix shard ledger

PATHBTC adopts dot-matrix shard ledger, which has the following characteristics:

1. Each account is a separate blockchain.

2. Blocks for each chain can only be generated by the account private key holder.

3. A transfer between two accounts requires an outgoing transaction from the originating account and an incoming transaction from the receiving account to complete.

4. Each account has a Merkle tree to record the current status of the account.

5. When the incoming transaction corresponding to the outgoing transaction is recorded, the outgoing transaction will be marked as settled. At this time, the originating account can discard the settled blocks and save only the account status and the unsettled blocks.

6. In addition to transactions, the account can provide Key-Value storage space.

7. Assets are described by (Field, ID, Count). Field represents the application serial number, ID represents the asset identification number, and Count represents its quantity. Such a generic asset description can represent both homogeneous and non-homogeneous assets.

Unlike packet sharding, this ledger generalizes cross-chain behavior. To achieve high throughput and low latency, the ledger unloads the transaction, splitting it into originating and receiving parts, which are created by separate accounts. Since the creation of different account blocks does not affect each other, after the settlement state of the block is introduced,

This model can gain tremendous flexibility in storage and throughput. Unlike Nano's account model, PATHBTC supports diversified assets and supports Key-Value through a Merkle tree of accounts

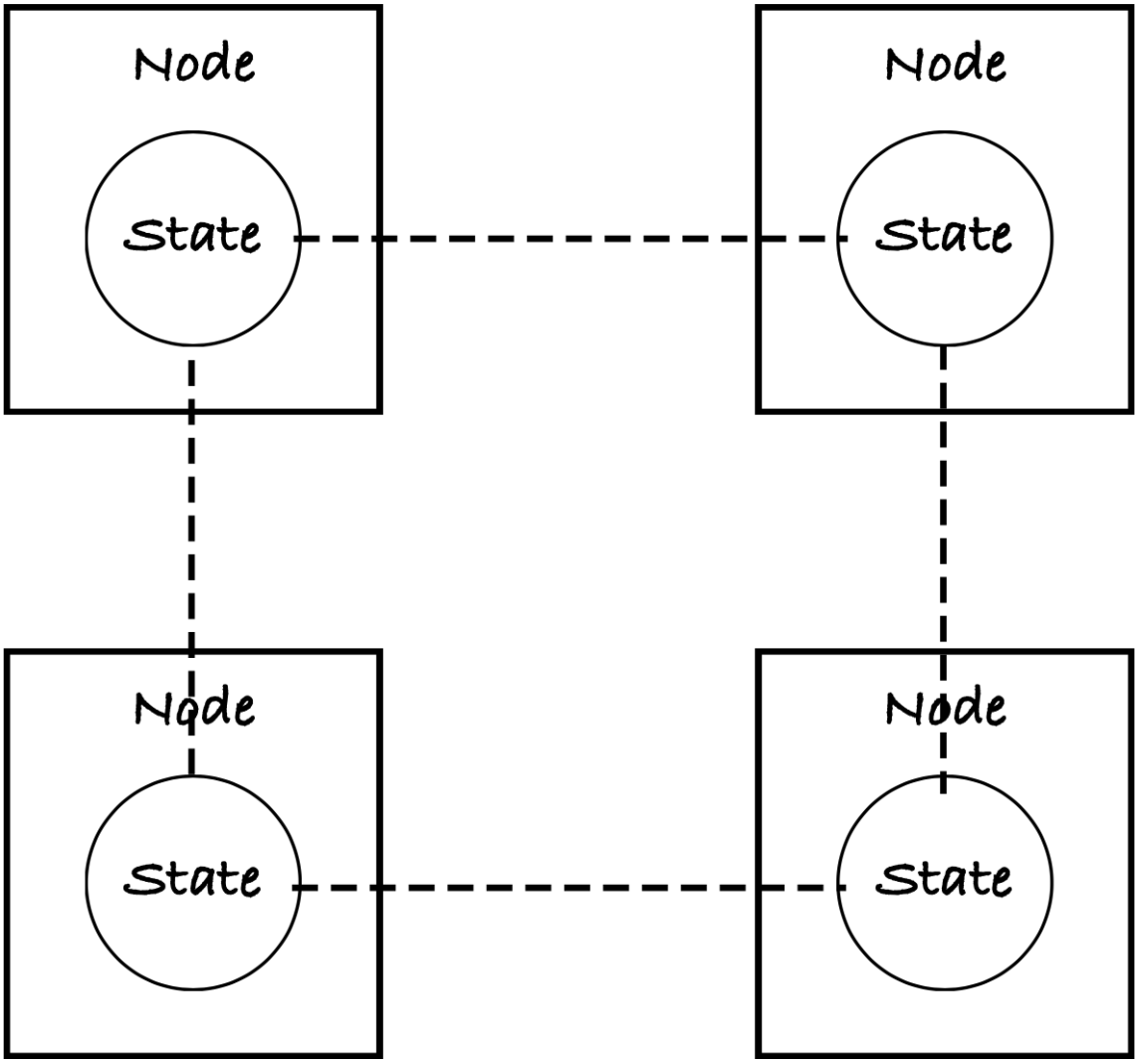
Store data.

C. Decentralized applications

PATHBTC's decentralized applications, in addition to the business requirements of realizable smart contract applications,

It also has a more flexible and higher-level definition. This definition enables PATH Protocol to implement not only complex decentralized applications, but also oracle,

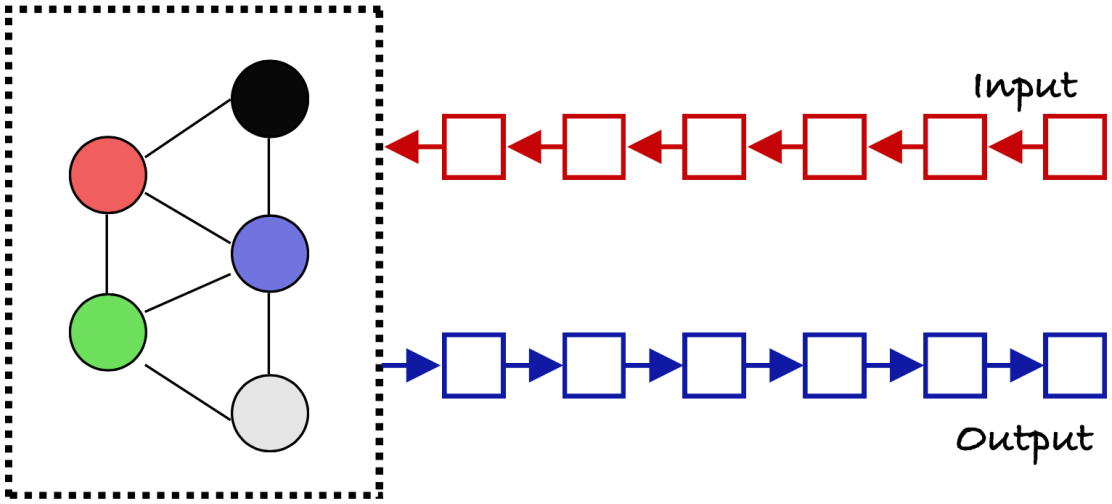
authority node, cross-chain applications. Through the advantages of PATHBTC's high throughput and low latency system architecture, the decentralized application approaches the experience of the centralized application.



Smart Contract Applications

The traditional definition of smart contract applications is based on Turing complete virtual machines, and the final state of the storage processed by the virtual machine is the ultimate meaning of the smart contract. Then this means that it is necessary

to maintain a consistent state within all nodes, no matter how the smart contract is operated, the final state within the smart contract account of each node needs to be ensured to be consistent, which is usually compared with the Root of the state-based Merkle Tree.



The decentralized application of PATHBTC

In PATHBTC, a decentralized application is defined as follows:

A decentralized application consists of a set of P2P network nodes that as a whole can input and output

The sequence is agreed upon. In extreme cases, the application can have only an output sequence or an input sequence.

In PATHBTC's implementation of a decentralized app, a decentralized app is a special account whose inputs and outputs are both blocks on the same chain. Unlike ordinary account blocking, which requires the signature of the account private key, the account signature of the decentralized application is an aggregate signature of the node group selected by the VRF. The nodes of the decentralized application have two roles, production node and audit node, which can be selected

by the VRF. The production node is responsible for producing blocks, and the audit node is responsible for verifying and signing blocks.

Application Accounts

Like ordinary accounts, in the PATHBTC protocol, the decentralized application also has an account, from the external table, similar to the ordinary account, is a blockchain. Unlike ordinary accounts, which are managed by private key holders, application accounts are managed by a set of decentralized nodes, which sign the block through the selected node group of VRF (similar to n-m multi-signature).

There is no upper limit to the number of application accounts that can be registered in PATHBTC. There is also an unlimited number of management nodes for each app account. Application behavior refer to PATHBTC's definition of decentralized application. Its input and output sequence blocks are managed by PATHBTC protocol. Due to the huge throughput of PATHBTC protocol, the interaction between applications and between applications and ordinary accounts will become very efficient.

Application No. **0**

App 0 is a decentralized system composed of decentralized nodes that maintain the blockchain corresponding to App 0's account. App 0's node is initially composed of some Genesis nodes and opens other nodes to join according to preset rules. The node corresponding to application 0 actually represents the consensus carrier of the PATHBTC protocol, and its main role is to regulate the PATHBTC network asynchronously. When other accounts need to create new applications, they need to apply to application 0.

Adding and exiting an application account

Decentralized systems, including App 0, require conditional entry and exit of their nodes, which are determined by the consensus of the app's current verification nodes. When the candidate node initiates the application to join a node that should be used, the decentralized verification nodes of the application reach a consensus through consensus, and the candidate node can become one of the verification nodes to participate in the consensus of the application. When the verification nodes reach a consensus, the verification node can exit the application account.

D. Verification mechanism

Ordinary and application accounts form the account structure of PATH Protocol. This account structure is independent of each other, and if the private key holder of an individual account or the management node of an application account is evil, it is difficult to resist this attack with the commonly used blockchain protocol. Therefore, PATH Protocol ensures that in the case of high throughput and low latency, a double spend attack by the administrator of both accounts cannot succeed through two steps of "random check" and "0 application confirmation mechanism" on the "validity propagation network".

Efficient Propagation Network

PATHBTC's application node 0 is responsible for the management of application 0 accounts and constitutes a P2P efficient propagation network. The network has two responsibilities, one is responsible for broadcasting messages to the whole network as much as possible, and the other is responsible for verifying the validity of each transaction. The nodes use DHT and Gossip protocols to form a P2P network, ensuring that each node is connected to at least D neighboring nodes, the greater the number of D, the faster the broadcast speed. The Gossip protocol is then used to broadcast messages between each node to ensure that any effective Messages are quickly broadcast across the network. When the node receives the transaction, it first confirms the validity of the transaction, such as whether the

incoming transaction has an outgoing transaction corresponding to it, whether the signature of the transaction is correct, and whether the account balance is sufficient. The efficient propagation network ensures that the message being propagated is valid.

Random checks

Random check means that the user who confirms the check randomly selects some nodes in the global node and obtains the account information on these nodes. If the status of these accounts is consistent, then this state can be considered as the correct status of the current account. The security can be improved by increasing the number of checks or the number of nodes checked each time.

There are three situations where random checks need to be used:

Nodes handle ledger forks: Since only the owner of each account can change his ledger, in general, the ledger will not fork. However, in the event of an error in the transaction sending program, or a malicious account manager launching an attack on the network, the ledger of their own account will be forked. The node then uses a random check mechanism to check the situation and reach a conclusion about the correct branch.

The client checks the outgoing transaction: The transaction of PATHBTC consists of the outgoing transaction and the incoming transaction. The recipient of the transaction can detect the account of the sender through the random check, so as to draw the conclusion whether the outgoing transaction is confirmed.

Solidify the ledger**:**

PATHBTC's nodes solidify ledgers of a certain height in bulk based on the changing conditions on the global Merkle Tree. Random checks are initiated when a branch of the Merkle Tree does not change for a certain period of time. If the results

are consistent, the account is solidified. If there is a solidified height before the transaction reaches, discard it directly.

Random checking is a final conformance check that allows point conflict handling and client validation checks to be performed independently of each other, because once random checking has observed conformance in a given situation, the condition is irrevocable.

0 applies the validation mechanism

In some extreme cases, the random confirmation mechanism cannot reach a conclusion, such as the two branches of the fork occupy half of the nodes, and cannot converge for a short time. In this case, the client can request arbitration from App 0 and pay a fee. At this point, App 0 will output an arbitration block and broadcast it to the whole network, and all nodes will select the correct branch of the account through this arbitration block. Since this process is asynchronous, only this abnormal account will be affected.

- Comparison between the current BTC main expansion schemes

	Turing complete smart contracts	Decentralization cross chains	EVM compatibility	High performance	High scalability	Lightning network compatible
Stacks	○	×	×	✓	○	×
RSK	✓	×	×	×		✓
RGB	○	×	×	×	×	✓
BitVM	○	✓	✓	×	×	×
Lightning+ Nostr	×	○	×	×	✓	✓
Liquid Network	✓	×	✓	○	○	×
PATHBTC	✓	✓	✓	✓	✓	✓
Bitcoin chain expansion						

3. The first subchain to achieve scaling

We will soon release the first Layer2 chain based on the PATH framework to meet the requirements of running smart contracts and high scalability, which is very important for building a practical application platform, mainly combining the following technologies to enhance the underlying architecture of the chain:

- Optimize consensus - Use a new consensus mechanism **PATH-Random**, which combines the latest PBFT theory and VRF algorithm to design a consensus mechanism that can take into account fairness and efficiency.
- Plasma - is a way to implement blockchain scaling calculation. In plasma, many blockchains are combined into a tree structure to participate in the calculation, so as to achieve the horizontal expansion of the blockchain.
- Stronger and larger virtual machine -- not only meet the EVM compatibility, but also have full scalability, and have the basis of underlying instructions to meet the performance needs.

The following will focus on the specific implementation of some technologies.

Based on the research of various types of consensus, this paper proposes its own main chain consensus engine
PATH-Random. The design idea of this consensus engine is inspired by Algorand and Ourboros. It only needs to verify nodes with little computing overhead, the whole blockchain network has a very small probability of bifurcating, and can achieve almost infinite scalability.

PATH-Random uses the Byzantine protocol BA*(Byzantine Agreement) to reach consensus on a set of transactions. For scalability, a random algorithm is used to

filter out a group of users, allowing the users themselves to privately check if they are selected and participate in reaching consensus in the BA* protocol. Under this algorithm, as the number of users increases, the whole BA* consensus system will not slow down.

The use of VRF algorithm

The PATH-Random consensus engine is based on the Verifiable Random Function (VRF) algorithm as the basis for random verification node selection. VRF is a random generation function, and this function is verifiable. That is, the same private key is used to sign the same information, and only one legitimate signature can be verified, which is different from the ordinary asymmetric encryption algorithm.

The specific operation process of VRF is as follows:

1. *The* prover generates a pair of keys*,* PUB_KEY and PRI_KEY. PRI_KEY is the private key*, and* PUB_KEY is the paired public key.

2. Prover outputs random result *result* = *VRF_Hash*(*PRI_KEY*,*info*)

3. Prover outputs random proof result = *VRF_Proof*(*PRI_KEY*,*info*)

4. The prover submits the random result and random proof to the verifier. The prover needs to verify whether result and proof match. If they match, proceed to the next step.

5. *The* prover submits PUB_KEY and *info to the verifier, and the verifier calculates* whether VRF_Verify(PUB_KEY*,*info*,proof*) *is TRUE. If* it is TRUE*, the verification passes.*

6. *If the* verification is passed, it can be deduced whether the info and *result match, that is, it proves that the material given by the verifier is correct. In the whole process, the verifier did not get the prover's private key* PRI_KEY.

A random Seed is generated

Some random algorithms in PATH-Random make use of seeds, such as PATH-Random's cryptographic draw, seeds that need to be randomly selected and disclosed. This seed must be known to the participating nodes, but not controlled by the opponent. The seed produced by PATH-Random round r is determined by VRF from the seed of the previous round R-1. This seed and the corresponding VRF proof are included in each proposed block, and once PATH-Random has agreed on the block in round R-1, everyone knows the pseudo-random seedr of the current round once Round r has started. The value of the initial seed0 is calculated by the initial participants working together with multiple nodes, resulting in an absolutely unpredictable random seed. In this way, the seed cannot be predicted by saboteurs or manipulated.

Method of selecting verifiers through a cryptographic lottery by VRF algorithm

PATH-Random uses the cryptographic drawing method to select a random subset of users according to the weight of each user. The system sets a fixed number of PATH coins as a screening candidate unit, and stipulates that each node has a limited number of PATH coins as a screening calculation, and the total weight of all candidate units is $W=\sum^{wi}$. And if node i has j PATH coins in the number of filtering units, the node can participate in the lottery screening with different identities of child nodes. The randomness of the lottery algorithm comes from the random seeds mentioned above. In each loop of BA*, PATH-Random builds a VRF based on the current seed, and the private key of the VRF can only be known by the node itself. Each node uses its own private key to run a random algorithm published by the system to draw lots. The system selects verification nodes according to the proportion of PATH coins held by the nodes that do not exceed the specified threshold.

PATH-Random specifies a threshold to select the expected number of validation nodes. This expected number satisfies the probability $p = w/W$. The probability of selecting a child verification node in W(total node weight) satisfies the binomial distribution:

$$B(k;W,p)=\binom{W}{K}p^{k}(1-p)^{w-k},where\sum_{k=0}^{W}B(k;W;p)=1$$

Method divides intervals [0,1) into the form of continuous intervals:

The way in which the number of selections of the current verification nodes (including child verification nodes) is determined is also determined by the lottery algorithm. The drawing algorithm divides the interval [0,1) into the form of a continuous

$$\mathbf{I}^{j}=\left[\sum_{k=0}^{j}B(k;w,p),\sum_{k=0}^{j+1}B(k;w,p)\right)\ for\ j\in\{0,1,...,w\}$$

interval:

If the hash has a bit length of hashlen, and if $hash/^{2hashlen}$ is in the interval ^{Ij}, then the node has j selected verification children. The number of selected validation nodes can be publicly validated by VRF using π. The characteristics of this cryptographic lottery method are:

1. The verification node randomly selects N verification child nodes according to the weight of PATH coins they hold

2. A saboteur who does not know the private key of node i has no way of knowing whether i is selected and how many child verifiers are selected.

The **BA*** consensus calculation is performed on the randomly selected verifier nodes

Verifier nodes (including child verifiers) are secretly informed that they have been selected, but they can only prove their verifier eligibility by publishing their credentials. For each selected node, the seed is signed with its own private key and the hash function is entered to get its credentials. The properties of the hash function indicate that the credential is a random string of 256 length, that the credential of different nodes will not be the same, and that the credential strings are evenly distributed. A group of candidate leadership nodes are selected in the same way, and the credentials of the candidate leadership nodes are arranged in lexicographic order. The smallest candidate leadership node in the sorting is selected as the leader node, that is, the leader node is randomly elected by the public through the set of candidate leadership nodes.

The verification node and the leader node participate in the calculation of the Byzantine protocol BA*. At each stage and step of BA*, the node independently determines whether it is selected in the committee of the current step through private and non-interactive means. BA* is a two-stage voting mechanism. In the first stage, the verification node carries out hierarchical consensus on the received candidate blocks, and selects the candidate block with the most verified consensus. In the second stage, binary Byzantine judgment is carried out on the candidate blocks selected in the first stage. BA* consensus should ensure that the number of honest nodes participating in the consensus is greater than 2/3. If the randomly selected set cannot meet this condition, then it is necessary to conduct multiple on-machine elections. As long as the number of honest nodes participating in the consensus is greater than 2/3, a consensus can be reached. The verification nodes of each step

of BA*

consensus are designated or screened by lot in parallel to speed up the consensus confirmation.

The steps of BA* consensus calculation

Each step of BA* involves destroying the current step temporary key, and the steps are outlined below:

1. Generate blocks (Step1)

1) The node checks whether it is the lead node Bir.

2) Generate the message mir from the first step ,1 = (Bir, ESGi (H (Bir)), σir, 1)

3) Broadcast Bir and mr,1 where

mr,s is the message that node i broadcasts at (r,s); Br is the block generated by node i in round r; ESG means to sign the message with the current (r,s) temporary key; H is for hash computation; Sigma r,s refers to the signature SIG (r,s,Qr−1) of I, used to prove that i exists in the set of verification nodes of (r,s).

2. Hierarchical consensus protocol

This protocol turns the problem of agreeing on any one block into agreeing on the two values that are the basis for the final determination of the hash of a particular block or the hash of an empty block, in 3 steps, which we will detail later in the Technical Yellow Book. Basically determine whether the message has more than 2/3 ESG V,σ and the same, if so, broadcast this (('),r 2), specific block, if not, broadcast the empty block, this message is used to follow the binary Byzantine judgment.

3. Binary Byzantine judgment

Verify the value emitted by the node statistical judgment hierarchical consensus protocol here. The binary Byzantine judgment is a three-step loop. The verification node constantly checks the received history to see if it has been met. There are two end conditions, that is, whether the block is valid or the block is not valid, and whether the total number of votes reached 2/3; If the block is not valid, the consensus system determines and generates an empty block. In order to prevent the occurrence of infinite cycles, we will set a maximum total number of cycles m. If we do not determine whether an end condition is met after reaching m, the consensus system will temporarily generate a tentative consensus, and form a final consensus in the subsequent process (later rounds), and confirm these earlier transactions.

PATH-Random consensus will adapt to the consensus decision in the case of weak network synchronization. In the case of strong network, block forks will not be caused. In the case of weak network synchronization, tentative consensus will be made temporarily and the final consensus will be reached after the recovery of strong network synchronization. PATH-random can protect against witch attacks, selfish mining attacks, noat-stake attacks, remote attacks and other attack modes. Even if the users of the PATH subchain spread to more than 100 million nodes, Path-RANDOM consensus can quickly reach a consistent Byzantine consensus across the entire network with the help of VRF mechanism.

4.2 Expansion Mechanism

Plasma is a framework for incentivizing and enforcing smart contract enforcement. It can scale up to a large number of status updates per second (up to 1 billion per second) and support a large number of decentralized financial applications worldwide on the blockchain. These smart contracts incentivize continuous automation through network transaction fees, ultimately relying on the underlying blockchain to force transaction state lock-in.

Plasma consists of two core components: reorganizing all blockchain calculations into a set of MapReduce functions, and an optional way to implement a Pos token deposit mechanism on existing blockchains without encouraging block retention under the Nakamoto consensus principle.

This build can enforce state locking on the main chain by writing smart contracts on the main chain, using fraud proof. Plasma groups blockchains into a tree-like hierarchy, treating each as a separate branch and forcing the entire history of the blockchain, along with Mapreducible calculations, to be submitted to Merkle proofs. By forcing the ledger information of a chain into a subchain through the main chain, the chain will be scaled up with minimal trust.

Block withholding attacks are a very complex issue around globally enforced data availability for non-global data. Plasms mitigates this problem with an opt-out mechanism for problematic chains, while also creating an incentive and consistently enforced correctness mechanism for executing data.

By broadcasting Merkle proofs of normal state to the main chain only periodically, this will allow for incredible scalability, reducing transaction costs and computation. Plasma supports the continuous operation of large-scale decentralized applications. Additional, important scalability is achieved by reducing the amount of money spent at a single time to represent one bit in a bitmap, so that one transaction and one signature represent a transaction with multiple parties easily aggregated. Plasma combines this with a MapReduce framework, while using smart contracts with deposits to build scalable computing mandates.

This architecture allows external parties to hold funds and calculate contracts based on their own behavior, much like a miner, but Plasma runs on an existing blockchain, so instead of creating a corresponding transaction on the main chain with every status update (even if that includes adding a new user to the ledger), Only

a small amount of information, such as the combined state change, needs to be written to the chain.

PATH uses Plasma as a mechanism for scaling up performance across multiple chains. This multi-chain parallel computing mechanism allows PATH to perform extremely high levels of status updates per second (possibly billions). This will enable PATH to achieve a significant performance improvement, to replace the current centralized cluster carrying capacity.

4.3 Virtual Machines

Currently Ethereum has a large number of developers and Solidity has become the most widely used language for smart contract development. Therefore, we need to provide EVM compatibility in the PATH subchain system.

The EVM virtual machine is developed on the basis of Ethereum, a standard blockchain structure with a single data structure, so its virtual machine is designed with database-like ACID(Atomicity, Consistency, Isolation, Durability) characteristics at the transaction call level . That is, in the protocol of Ethereum, the call of a smart contract may affect the status change of multiple accounts. These state changes are rigid transactions that have real-time consistency, i.e. these state changes either happen at the same time or none of them happen. However, PATH needs to allow for sufficient scalability in the future and have a foundation of underlying instructions to meet performance requirements. We design the virtual machine of the PATH chain to meet the BASE(Bascically Available, Soft state, Eventual consistency) principle, and we call this virtual machine MEVM.

In BASE concept, basic availability means that the system is allowed to lose some availability in the event of unexpected failure; Soft state means that data in the system is allowed to exist in an intermediate state, but the existence of the intermediate state will not affect the overall availability of the system. Ultimate

consistency means that all copies of the data, after a period of synchronization, are finally consistent. In contrast to the strong consistency of the ACID concept, the BASE concept gains usability by sacrificing strong consistency in real time, but ultimately achieving a consistent state. The block structure and various consensus algorithms in the blockchain are essentially in line with the BASE concept, but they do not meet the ACID concept. Therefore, MEVM virtual machine design is suitable for compound BASE semantics, and in this level compared with the original EVM ACID design, will overcome this aspect of the performance bottleneck constraints.

In addition, Solidity language has been criticized one point is the lack of standard library support, such as comparing two strings such basic functions, Solidity has no standard library functions for developers to call. Projects such as OpenZeppelin provide some standard libraries, but they are far from sufficient. In particular, PATH's blockchain applications require libraries of advanced mathematical and cryptographic algorithms, such as zero-knowledge proof protocols, RSA public-key cryptography, and singular value decomposition. MSolidity can refer to these implementations and add more libraries, which are precompiled or implemented in Native mode to reduce the running cost.

In the future, the PATH architecture will consider supporting Web Assembly(WASM) -based virtual machines to further improve performance and provide support for smart contracts written in languages other than Solidity, such as C, C++, Rust, or Go. As the IELE virtual machine designed by the Cardano project matures, PATH will also consider providing support for this virtual machine. IELE, a variant of LLVM, has the potential to become a unified, low-level platform for the translation and execution of smart contracts in high-level languages. The IELE virtual machine enables the PATH architecture to support a wider variety of high-level languages.

4.4 Quantum computing resistance

The asymmetric cryptographic signature algorithms commonly used on blockchain systems at present, such as RSA algorithm based on the factorization problem of large integers and ECC algorithm based on the computation problem of discrete logarithms on elliptic curves, can be turned into a P problem by quantum Shor algorithm, so that it can be easily cracked. The PATH system will introduce encryption algorithms that resist brute force cracking of quantum computing in due time according to the project schedule and the development of quantum computer practicality, such as Lattice-based cryptography. code based cryptosystems and multivariate cryptography; Among them, lattice-based cryptosystems can be designed for encryption, signature, key exchange and other cryptosystems, which is an important direction of post-quantum cryptography algorithms. At the same time, we will also synchronously track the cutting-edge research directions of quantum-resistant cryptosystems designed based on the Isogen problem on the super-specific elliptic curve, conjugacy search problem and Braid Groups related problems.

Step 5: Milestones

- In mid-2023.12, the first supported protocol, Ordinals BRC20-Layer2 assets, is launched, and PATH enters the cross-chain bridge to the L2 bottom pool
- In the middle of 2024.1, open the BRC20 -> PATH cross-chain exchange channel and open the ecological node
- In late 2024.1, Launch the mining pool and start Layer2 PoW mining
- Late 2024.2, open source mining pool
- 2024.3, Launch Layer2 mainnet, and support all OD inscriptions across chains to PATH, support to build Dapps

References

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

[2] Wenshuai Zhang, Jing Li, "A Method, System and storage Medium for layering Tailoring Data within Blockchain Transactions", https://patents.google.com/patent/CN113360578A/ zh?oq=202110682927+.8

[3] Jiang Jie, Gu Lu, "induction contract: a traceable and collaboration based on feature recognition of contract", https://sensiblecontract.org/files/sensible-contract-v0.2.0.pdf

[4] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[5] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.

[6] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.

[7] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

[8] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[9] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122- 133, April 1980.

[10] W. Feller, "An introduction to probability theory and its applications," 1957.

[11] MONACO J V. Identifying Bitcoin users by transaction behavior[C]//The SPIE DSS, April 20-25, 2015, Baltimore, USA. Baltimore: SPIE, 2015.

[12] ZHAO C. Graph-based forensic investigation of Bitcoin transactions[D]. Iowa: Iowa State University, 2014.

[13] LIAO K, ZHAO Z, DOUPE A, et al. Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin[C] //The Symposium on Electronic Crime Research, June 1-3, 2016, Toronto, Canada. Piscataway: IEEE Press, 2016: 1-13.

[14] MEIKLEJOHN S, POMAROLE M, JORDAN G, et al. A fistful of bitcoins: characterizing payments among men with no names[C]// The 13th ACM Internet Measurement Conference, October 23-25, 2013, Barcelona, Spain. New York: ACM Press, 2013: 127-140.

[15] ROND, SHAMIR A. Quantitative analysis of the full Bitcoin transaction graph[C]//The 17th International Conference on Financial Cryptography and Data Security, April 1-5, 2013, Okinawa, Japan. Heidelberg: Springer, 2013: 6-24.

[16] GENNARO R, GENTRY C, PARNO B, et al. Quadratic span programs and succinct NIZKs without PCPs [C]//The 32nd Annual International Conference on the Theory & Applications of Cryptographic Techniques, May 26-30, 2013, Athens, Greece. [S.L.:S.N.], 2013: 626-645.

[17] PARNO B, HOWELL J, GENTRY C, et al. Pinocchio: nearly practical verifiable computation[C]//The 2013 IEEE Symposium on Security & Privacy, May 19-22, 2013, San Francisco, USA. Washington, DC: IEEE Computer Society, 2013: 103-112.

[18] REID F, HARRIGAN M. An analysis of anonymity in the Bitcoin system[C]//The 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust, October 9-11, 2011, Boston, USA. Piscataway: IEEE Press, 2011: 1318-1326.

[19] ANDROULAKI E, KARAME GO, ROESCHLIN M, et al. Evaluating user privacy in Bitcoin[C]//The 17th International Conference on Financial Cryptography and Data Security, April 1-5, 2013, Okinawa, Japan. Heidelberg: Springer, 2013: 34-51.

[20] CHAUM D. Untraceable electronic mail, return addresses and digital pseudonyms[J]. Communications of the ACM, 2003: 211-219.

[21] VALENTA L, ROWAN B. Blindcoin: blinded, accountable mixes for Bitcoin[J]. Financial Cryptography and Data Security, 2015: 112-126

[22] SHENTU Q C, YU J P. A blind-mixing scheme for Bitcoin based on an elliptic curve cryptography blind digital signature algorithm[J]. Computer Science, 2015.

[23] RUFFING T, MORENO-SANCHEZ P, KATE A. CoinShuffle: practical decentralized coin mixing for Bitcoin[M]// Computer Security -ESORICS 2014, Heidelberg: Springer, 2014: 345-364.

[24] BISSIAS G, OZISIK A P, LEVINE B N, et al. Sybil-Resistant mixing for Bitcoin[C]// The 2015 ACM Workshop on Privacy in the Electronic Society, November 3, 2014, Scottsdale, USA. New York: ACM Press, 2014: 149-158.

[25] DWORK C, NAOR M. Pricing via processing or combatting junk mail[C]// The 12th Annual International Cryptology Conference on Advances in Cryptology, August 16-20, 1992, Santa Barbara, USA. Piscataway: IEEE Press, 1992: 139-147.

[26] CASTRO M, LISKOV B. Practical byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems, 2002, 20(4): 398-461.

[27] BONNEAU J, NARAYANAN A, MILLER A, et al. Mixcoin: anonymity for Bitcoin with accountable mixes [C]//The 19th International Conference on Financial Cryptography and Data Security, January 26-30, 2015, San Juan, Argentina. Barbados: Financial Cryptography, 2014: 486-504. [