PATH Protocol

Full chain ecological routing network

Infinitely scalable BTC Layer2 network infrastructure

PATH

www.pathbtc.com

CONTENT

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 to a large extent, it plays a greater role than the law.

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 in the case of the Ethereum network, there still exists a significant gap in legitimacy compared to the Bitcoin network.

1.2. Status

Discussing blockchain infrastructure as a technological foundation (rather than a financial commodity), the technical choices play a role in evaluating its legitimacy, much like the significance of lineage in biology. Over the past 8 years, Ethereum has played a significant role, bearing the expectations and responsibilities of a decentralized society. Firstly, Vitalik was the first to propose smart contracts. Secondly, and not to be overlooked, he himself was deeply involved in the early Bitcoin community and consistently championed Satoshi Nakamoto's ideals throughout Ethereum's development..

It can be said that the past 8 years belong to Ethereum. It has consistently led industry development, proposed new directions at crucial junctures, and sought to address new challenges technically. In the bull market of 2021, DeFi emerged as a significant driver, with Ethereum serving as its foundation. Bitcoin itself did not play a major role in this bull market, and that's okay. As digital gold, it still experienced significant gains. However, as profit-taking funds exit the market, it leaves behind a sense of pessimism in the industry, especially when most practitioners are enthusiastically championing the Web3 movement. We find that the entire technological landscape seems to have deviated from the initial belief, which is decentralization.

Scaling solutions based on Ethereum have undergone significant contemplation and experimentation. Among them, the most aligned with the decentralized principles of Bitcoin are the technologies of sharding and Plasma. However, for various reasons, neither has seen robust development or widespread adoption. On the contrary, relatively more centralized Layer 2 solutions have become the 'mainstream.' This includes Ethereum's transition to PoS (Proof of Stake). Regardless of perspective, capital has played a significant role in these crucial decisions, highlighting one of the most pressing issues in the current blockchain landscape. The shift towards PoS and the dominance of more centralized Layer 2 solutions underscore the influence of capital, and it seems that the brave have eventually become the dragon. From a technical standpoint, both sharding and Plasma are promising scaling solutions, and the reasons hindering their development are certainly not inherent to the technology itself.

These current situations have caused people to return to the Bitcoin network and seek new solutions.

1.3. The Rise of Bitcoin Ecosystem

No one would have thought that 2023 would be the year of the Bitcoin ecosystem, but it seems that this has become a reality. From the Ordinals protocol in the first half of the year to the release of the Taproot asset protocol on October 19, and then the basic technologies for Bitcoin network expansion such as the RGB protocol, Bitcoin smart contract Rootstock, and Bitcoin second-layer Stacks, they began to gain attention. People have come to realize that there is much more that can be achieved on the Bitcoin network, without the drawbacks seen in the Ethereum ecosystem.

These Bitcoin network ecosystem infrastructures didn't just emerge; they were more or less influenced by the SegWit fork in November 2017. However, it wasn't until 2021, ignited by the Taproot Assets, that things took a magical turn. For five years, the Bitcoin ecosystem did not directly compete with Ethereum until 2021, and everything naturally unfolded. Similar to the IPO in 2015, but the difference this time is that everything happened in a decentralized manner. It feels like we've returned to the initial state where the community is driving the development of the ecosystem.

A more pure technical environment, transparent governance, and a decentralized network theoretically capable of supporting the latest interactive experiences on the Internet undoubtedly represent the true Web3 revolution.

Among all known scaling solutions for the Bitcoin network, PATHBTC stands out as one of the most competitive members, boasting both technical openness and scalabliltiy.

2. PATH Protocol solution

2.1. Cross-chain

In a decentralized world, various encrypted assets in the blockchain industry are the cornerstone of the Web3 ecosystem. Therefore, asset cross-chain is a basic capability and one of the driving forces of PATH. According to the plan, this cross-chain protocol is named “PathBridge”. We will choose the most reasonable technical solution to build this cross-chain network

Mainstream cross-chain solutions

Cross-chain has always had two different interpretations. The first refers to atomic swaps between two blockchain systems, where the total circulation of related assets in both systems remains unchanged after the transaction. The second involves the transfer of assets between two different blockchain systems, where the circulation of the transferred assets in both blockchain systems changes, but the sum of their circulation remains constant. PathBridge's cross-chain refers to the latter. Since Ripple proposed the InterLedger protocol in 2012, the industry has had dozens of different technical solutions to achieve cross-chain asset transfer types. However, in terms of essential principles, these solutions can be divided into two major categories based on the confirmation method： Notarization and Relay.

A. Notarization Solution

This solution requires a trustworthy set of decentralized entities to monitor two different blockchain systems and correctly execute predefined transfer operations when asset locking is detected. From an implementation perspective, there are two technical approaches: multi-signature mechanisms and distributed signature mechanisms.

Multi-signature Mechanism: In this approach, multiple notary nodes simultaneously supervise the locked account. Cross-chain transactions can only be completed after receiving a certain number of correct signatures from the notary nodes. This method requires both blockchain systems to support multi-signature mechanisms or have smart contract functionality. ChainBridge, PalletOne, and Ren adopt this model.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 distributed signature mechanism uses the linear characteristics of elliptic curve operations to disperse the private keys of asset-locked accounts to different notaries using threshold signature technology. A certain number of notaries need to be gathered to complete the signature of cross-chain transactions. This method requires that both blockchain systems have certain fixed modes of elliptic curve signature account systems, such as ECDSA or EDDSA. Wanchain (Fusion) and tBTC adopt this model.

The advantage of this solution is that the notary accesses the crossed blockchain system in the form of a client, which is not intrusive to the original chain. This method is technically inclusive and flexible and can be adapted to most current blockchain systems. The disadvantage is that a third party other than the two chains is required for notarization, which adds a possible risk point.

B. Relay Solution

Using the relay scheme, the internal or smart contract of each of the two blockchain systems has a lightweight block mapping of the other and can verify the validity and authority of the other's blocks through these mappings. After both sides have produced blocks, any node can submit block mapping information to the smart contract of the other side. The smart contract on the other side simulates the block verification process. In this way, other contracts deployed on both blockchain systems can obtain status information about the other blockchain through this mapping. This is used to verify whether the smart contract on the other side has truly received the locked funds and then proceed with the subsequent transfer operations. BTC-relay, Cosmos Hub, and Polkadot's RelayChain are examples of implementing relay in this way.

The advantage of this approach is that it doesn't require a third party, and any entity can submit block mapping information to complete the relay. However, the disadvantages are evident. It is intrusive, requiring consideration of the relay scheme during chain design and the provision of interfaces. Moreover, each generated block needs to be submitted to the other side, which, besides consuming storage resources, can result in significant transaction fees for fee-charging chains. Additionally, if one chain changes its consensus protocol, the relay on the other side needs to be rewritten. Another issue with the relay scheme is that if the relayed original chain experiences a long fork while the chain carrying the relay mapping has already been confirmed and released the assets, it may be in a split state, leading to unpredictable results.

Design Approach

Due to the preceding nature of PATHBTC, the following design intentions are outlined before introducing technical solutions:

1. Decentralized Solution: A decentralized solution with a security level not noticeably weaker than the crossed blockchain system.

2. Feasibility and Usability: The solution should have sufficient feasibility and usability, not limited to theoretical and experimental aspects.

3. Wide Applicability: Capable of providing cross-chain functionality for the majority of mainstream blockchain systems.

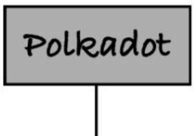
4. Support for FT and NFT Assets: Capable of completing cross-chain transactions for both fungible (FT) and non-fungible (NFT) assets.

5. Migration Capability: Able to eventually migrate to the cross-chain framework within the PATHBTC mainnet protocol network.

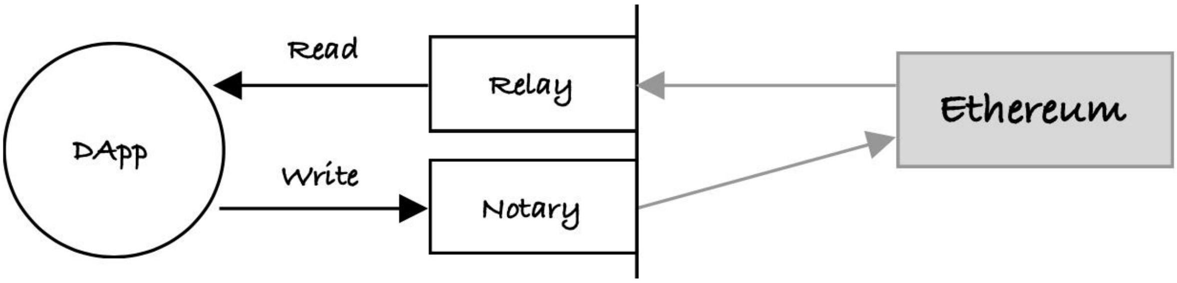
6. Privacy Protection: Providing privacy protection for the owners of both fungible and non-fungible assets and their related transaction behaviors.

7. No Introduction of Additional Consensus Assets: The underlying system does not introduce another type of fungible asset for consensus maintenance.

8. Simplicity and Intuition in Design: A design that is simple, intuitive, and easy to implement and maintain.



Since PathBridge is ultimately intended to migrate into the PATH Protocol protocol network, which serves as the infrastructure for accommodating all virtual assets, and the PATH Protocol network independently interfaces with various other blockchain systems, the integration of PathBridge with each third-party blockchain system should also be conducted independently.



In the PATHBTC protocol network, as the protocol inherently supports cross-chain functionality from the beginning, placing relays for various other blockchain systems within the application accounts of the PATH Protocol protocol is a natural choice. Due to the diversity of other blockchain systems and the uniqueness of the PATHBTC protocol, it is challenging to link the PATHBTC protocol to other blockchains in the form of relays. Therefore, a notarization mechanism is inevitably required to initiate instructions to other blockchain systems. The PATHBTC protocol must be a hybrid system with both relays and a notarization mechanism. The state of other blockchain systems is exposed to the PATHBTC protocol through relays, allowing applications on the PATHBTC protocol network to freely query the state of other blockchain systems. Cross-chain instructions from the PATHBTC protocol are issued to other blockchain systems in the form of notarization. As long as the notary nodes of the PATHBTC protocol are sufficiently secure, the most cost-effective approach can be adopted to drive other blockchain systems.

Given that the cross-chain functionality of PATHBTC ultimately adopts a hybrid model, it is necessary to first implement the notarization part of the mechanism as a cross-chain component before the release of the PATHBTC protocol network. The notary nodes for cross-chain currently have no local state; they can persist as genesis nodes when the PATHBTC protocol network is launched.

Implementation Method

1. Decentralized notary system

Notary nodes are service programs running on the internet network 24 hours a day, and all notary nodes together form the notary committee. The committee is initially composed of several genesis nodes, and other nodes wishing to enter the committee need approval from more than 2/3 of the committee through a vote. Nodes can leave with the approval of more than 2/3 of the committee's vote.

The execution of each cross-chain instruction involves a committee vote. When the number of committee nodes is low, more than 2/3 of the total committee votes are required for the instruction to take effect. When there are too many committee nodes, the VRF algorithm will be used to generate a temporary committee for each cross-chain instruction, and N rounds of (N>=1) committee voting will be conducted to eventually determine the result.

Since PathBridge, like PATHBTC, does not initially issue fungible assets, a fee will be charged for each cross-chain invocation in the corresponding asset as a transaction fee for executing the cross-chain transaction. These fees will be transferred to the account of the notary node initiating the vote upon completion of the cross-chain operation

2. Fairness and security

As a notary committee, there are two main challenges in security, selfish behavior and malicious behavior. Selfish Behavior is an inactive behavior in providing notarization services, which will lead to a decrease in the effective service resources of cross-chain services. Malicious behavior is the act of conducting false or invalid votes, such a node will be locked Asset safety factor is reduced. PathBridge uses the following methods to prevent this situation.

3. Genesis Node KYC

To reduce the likelihood of collusion among nodes in the initial stages, PATH will conduct rigorous KYC (Know Your Customer) verification for the limited number of initial genesis nodes. This will ensure, to some extent, the independence of social relationships among genesis nodes, eliminating malicious behavior during the initial system operation and ensuring the healthy 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.

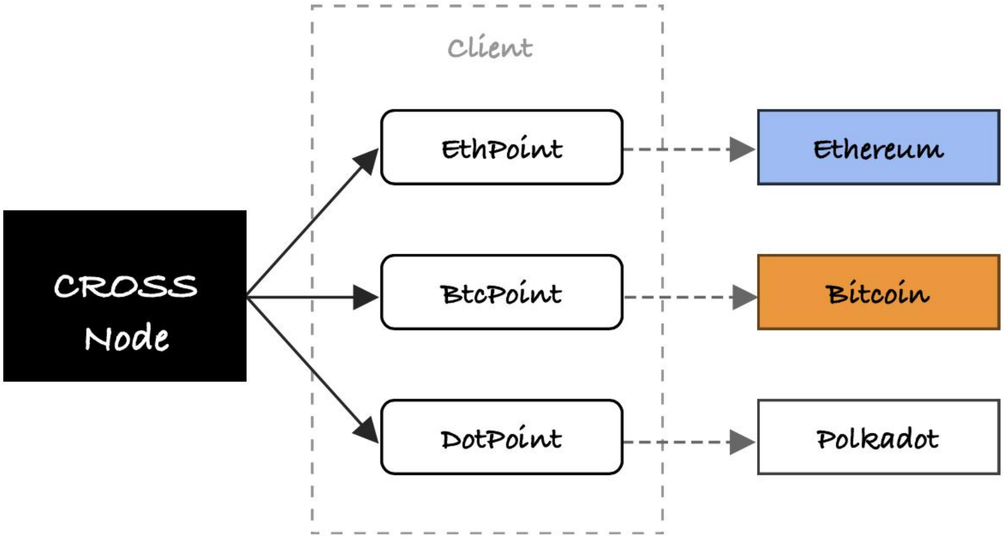
Under normal circumstances, the voting weight of each node is the same, and PathBridge will take a fair cut of the issuing fee according to the notary's row, but in the following two cases, their voting weight may be reduced.

6. Committee oversight

The committee periodically reports malicious nodes. If a node is reported by more than half of the committee within a cycle, it is considered malicious, and the voting weight of that node will be reduced.

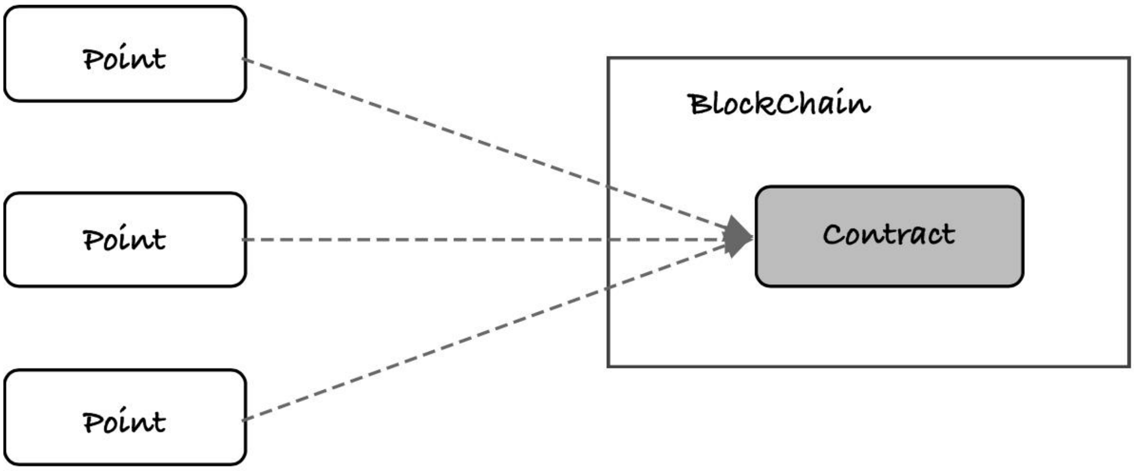
7. PathBridge Node

PathBridge nodes connect to different blockchain systems through components with the suffix 'Point,' such as connecting to Ethereum through EthPoint, Bitcoin through BtcPoint, and Polkadot through DotPoint. Points have standardized interfaces that are called by PathBridge Nodes. The Point component serves as a client for various blockchain systems, and PathBridge reads information from the chain and initiates cross-chain instructions through Points.

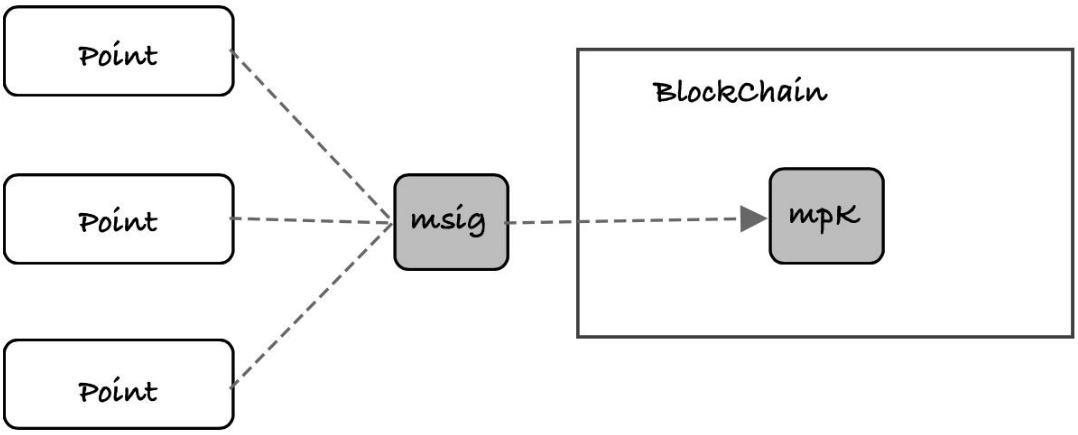


8. Multiple signatures

For blockchain systems with smart contracts, funds are locked in smart contracts. The Point components of different PathBridge nodes in the committee will register in a specific smart contract and make contract calls according to decentralized rules. In this way, functions similar to multi-signature are implemented.



To address the issue of excessive Gas consumption due to multiple node votes, PathBridge introduces an aggregate signature mechanism. When a validating node detects a new cross-chain transaction, it immediately broadcasts its signature for that transaction. Any node can collect signatures from other nodes in the committee and then aggregate them into a call parameter submitted to the smart contract.



For blockchain systems without smart contracts, if the account and signature system supports multi-signature functionality, the Point components of different PathBridge nodes in the committee will access the funds locked in the account through multi-signature functionality.

However, for the notarization mechanism implemented through multi-signature, when the number of signatories changes, the public key of the locked account typically changes. In this case, the notary nodes in the committee need to coordinate through consensus to generate another locked account to receive the assets from the original locked account, completing the process of member joining or exiting.

2.2. Scalability

PATHBTC Protocol is a decentralized blockchain protocol, serving as the standard and rules for nodes within the decentralized network of PATHBTC. These nodes independently operate in the network environment, exchanging information, forming a scalable, efficient, and tamper-resistant decentralized infrastructure.

PATHBTC Protocol is the cornerstone of the decentralized world in the web3 era after BTC scaling. Therefore, in addition to basic public blockchain features, it needs to possess the following characteristics and address challenges

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.

2.3. Solutions 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.

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.

A. 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?

B. 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.

C. 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.

D. 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.

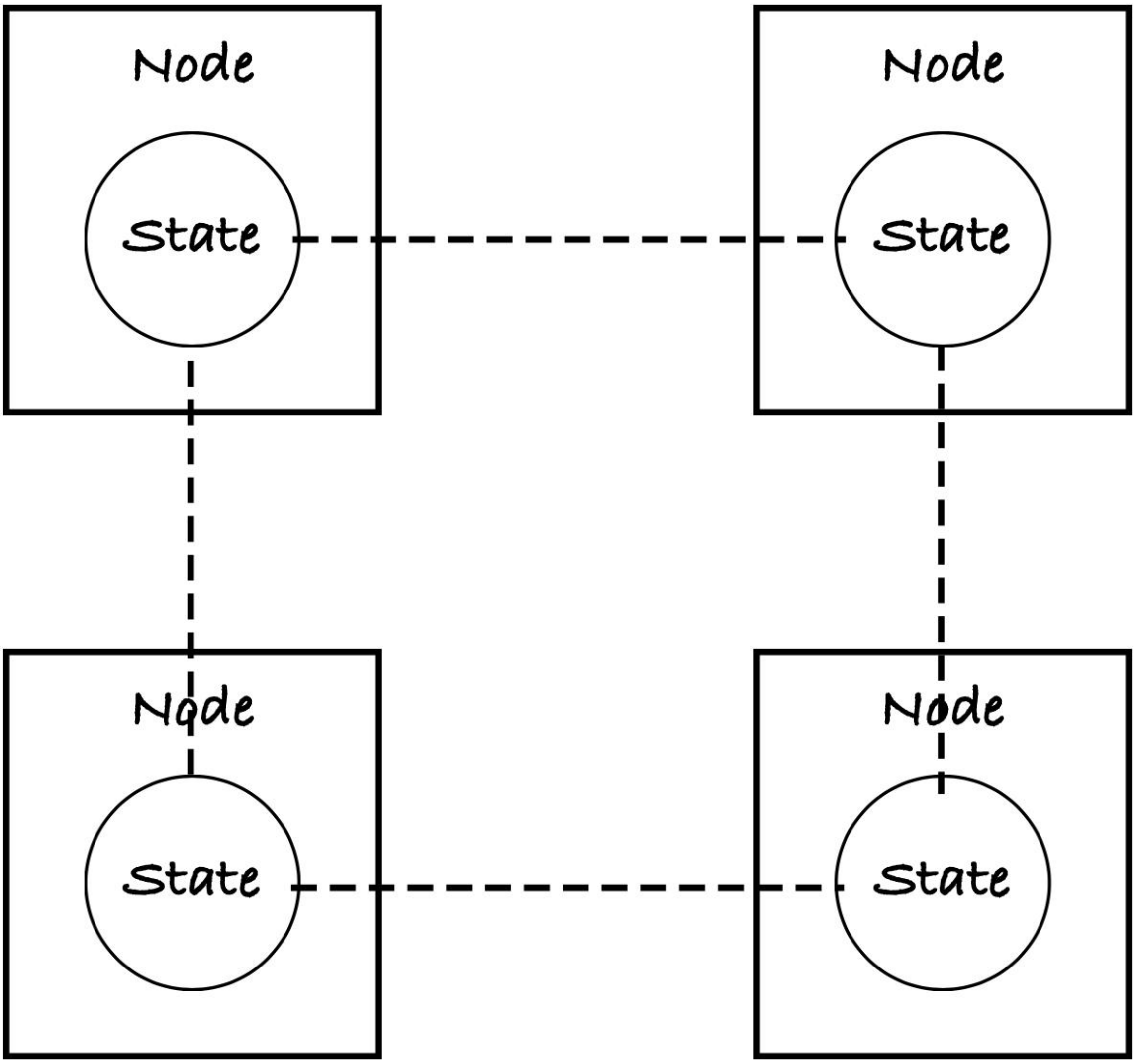
7. Assets are described by (Field, ID, Count). Field represents the application serial number, ID represents the asset identification number, and Count represents its quantity. Such a generic asset description can represent both homogeneous and non-homogeneous assets.

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.

A. Decentralized applications

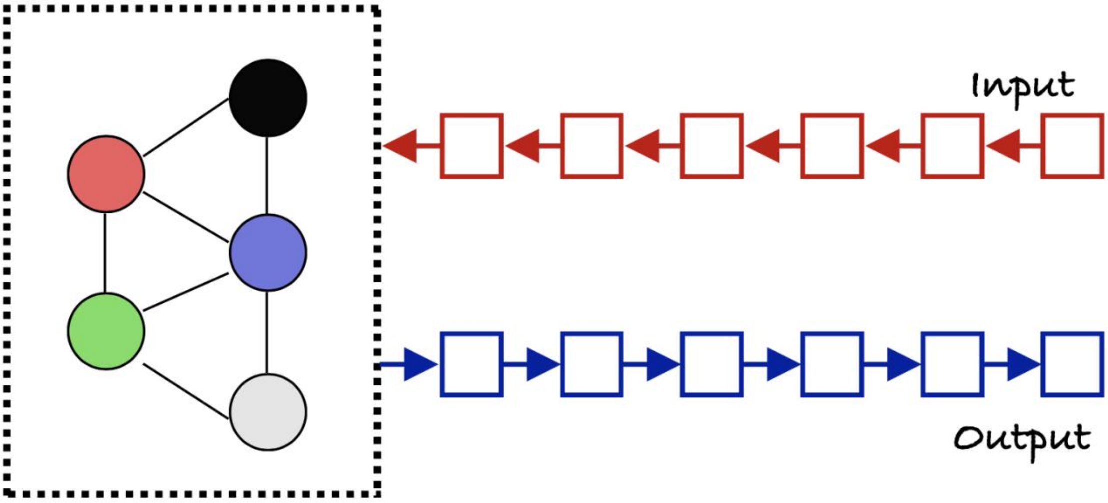
PATHBTC's decentralized applications, in addition to the business requirements of realizable smart contract applications, It also has a more flexible and higher-level definition. This definition enables PATH Protocol to implement not only complex decentralized applications, but also oracle, authority 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.

B. 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:

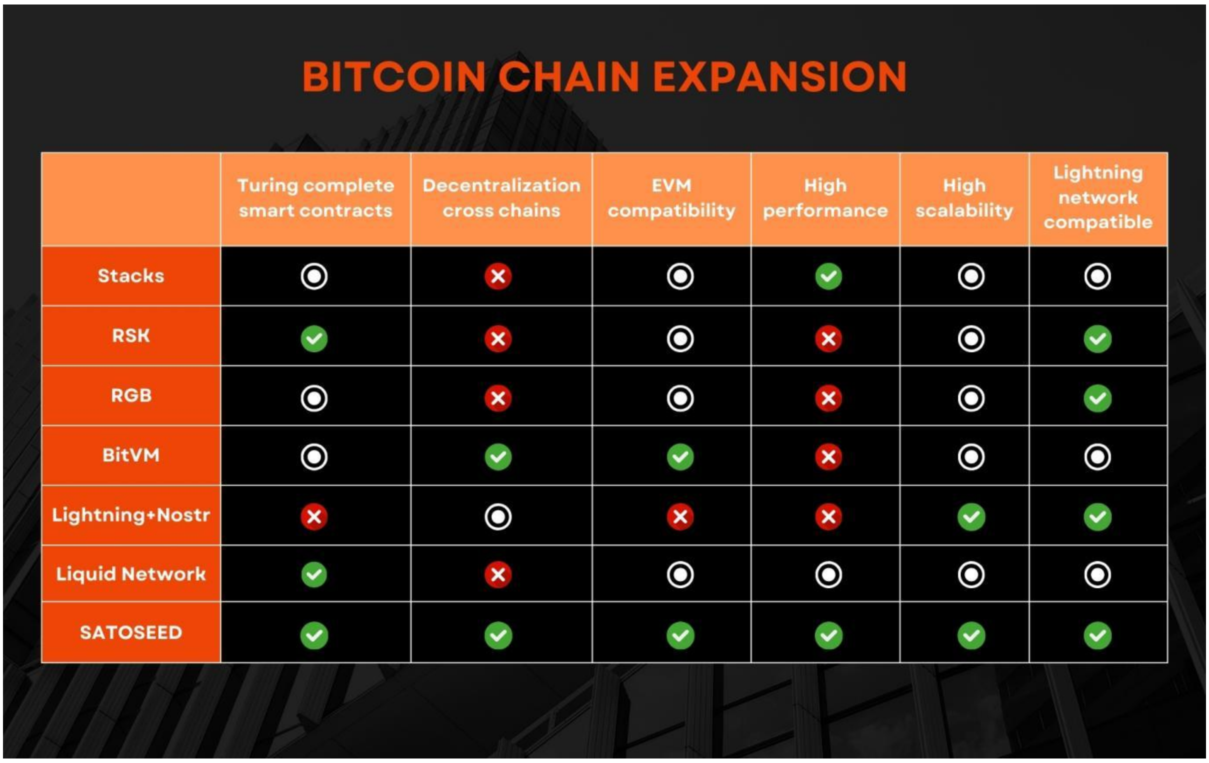
PATHBTC's nodes solidify ledgers of a certain height in bulk based on the changing conditions on the global Merkle Tree. Random checks are initiated when a branch of the Merkle Tree does not change for a certain period of time. If the results are consistent, the account is solidified. If there is a solidified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



BITCOIN CHAIN EXPANSION

	Turing complete smart contracts	Decentralization cross chains	EVM compatibility	High performance	High scalability	Lightning network compatible
Stacks	◉	✗	◉	✓	◉	◉
RSK	✓	✗	◉	✗	◉	✓
RGB	◉	✗	◉	✗	◉	✓
BitVM	◉	✓	✓	✗	◉	◉
Lightning+Nostr	✗	◉	✗	✗	✓	✓
Liquid Network	✓	✗	◉	◉	◉	◉
SATOSEED	✓	✓	✓	✓	✓	✓

3. The first sub-chain to implement scalability

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:

- Optimized Consensus: Utilizing a new consensus mechanism, PATH-Random, which combines the latest PBFT theory and VRF algorithm design, offering a consensus mechanism that relatively balances fairness and efficiency.

- Plasma: Introducing Plasma as a way to achieve blockchain scalability. In Plasma, numerous blockchains are combined into a tree structure, collectively participating in computation to achieve horizontal blockchain expansion.

- Enhanced Virtual Machine: Providing a more powerful virtual machine that not only satisfies EVM compatibility but also offers ample scalability. It should have a foundation of low-level instructions to meet performance requirements.The following will focus on the specific implementation of some technologies.

The following will focus on the detailed implementation of some technologies：

3.1 Consensus Mechanism

Based on the study of various consensus mechanisms, we propose our own main chain consensus engine, PATH-Random. The design of this consensus engine is inspired by Algorand and Ourboros, incurring minimal computational overhead for validating nodes. The probability of the entire blockchain network experiencing forks is extremely low, and it achieves near-infinite scalability.

PATH-Random employs the Byzantine Agreement (BA*) protocol to achieve consensus within a set of transactions. For scalability, a random algorithm is used to select a group of users, allowing users to privately check whether they are selected and participate in achieving consensus in the BA* protocol. Under this algorithm, as the number of users increases, the entire BA* consensus system does 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 = VRF_Hash(PRI_KEY,info)*

3. Prover outputs random *proof = 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*.

Random **Seed** Generation

The random algorithm in some places of PATH-Random will use a seed, such as in the cryptographic drawing of PATH-Random, where a seed needs to be randomly selected and publicly disclosed. This seed must be known to the participating nodes but not controllable by adversaries. The seed generated for the r-th round of PATH-Random is determined by the VRF based on the seed from the previous round (r-1). This seed and the corresponding VRF proof are included in each proposed block. Once consensus is reached on the blocks in round r-1 of PATH-Random, at the beginning of round r, everyone knows the pseudorandom seed r for the current round. The initial value of seed0 is calculated by the initial participants using multiple nodes to obtain an absolutely unpredictable random seed. This ensures that the seed cannot be predicted by "attackers" and cannot be manipulated.

Method to select verifiers by encrypted lottery using VRF algorithm

PATH-Random employs an encrypted drawing method to select a random subset of users based on their individual weights. The system designates a fixed unit quantity of PATH coins as a filtering candidate unit and specifies a limited number of PATH coins for each node to conduct filtering calculations. The total weight of all candidate units is denoted as $W=\sum w_i$. Moreover, if node i possesses j units of PATH coins for filtering, the node can participate in drawing selection for different subnodes. The randomness of the drawing algorithm is derived from the aforementioned random seed. In each iteration of BA*, PATH-Random constructs a VRF based on the current seed, and the private key of the VRF is known only to the respective node. Each node uses its private key to execute the random algorithm disclosed by the system for drawing. The system selects validation nodes based on the ratio of PATH coins held by each node, ensuring it does not exceed the predefined threshold.

PATH-Random requires specifying a threshold to determine the expected number of validation nodes. This expected quantity adheres to the probability: $p = w/W$。The selection of sub-validation nodes within the total node weight W follows a binomial distribution. This distribution divides the interval [0, 1) into continuous intervals.

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

The determination of the current number of selected validation nodes (including sub-validation nodes) is also governed by the drawing algorithm. The drawing algorithm partitions the interval [0,1) into continuous intervals.

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

If the scattered bit length is hashlen, and if $hash/2^{hashlen}$ is in the interval I, then the node has j selected validation sub-nodes. The number of selected validation nodes can be verified using p in the VRF public verification. The characteristics of this encrypted drawing method are:

1.Validation nodes randomly select N validation sub-nodes based on the weight of PATH coins they hold.

2.Attackers without knowledge of the private key of node i cannot determine whether i has been selected or how many sub-validation nodes have been chosen.

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

Validation nodes (including sub-validation nodes) are aware of their selection in secret, but they can only prove their validator status by publishing credentials. For each selected node, they sign the seed with their private key, and the resulting signature is hashed to obtain their credentials. The nature of the hash function ensures that the credentials are a random string of length 256, unique for different nodes, and the distribution of credential strings is uniform. In the same manner, a group of candidate leader nodes is selected, and the credentials of these candidate leader nodes are arranged in dictionary order. The candidate leader node with the smallest credential in the sorted order is chosen as the leader node, meaning the leader node is randomly elected through a public election based on the set of candidate leader nodes.The verification node and the leader node.

Validation nodes and leader nodes participate together in the Byzantine Agreement protocol BA*. In each phase and step of BA*, nodes independently determine whether they are selected in the current committee through private and non-interactive means. BA* is a two-phase voting mechanism. In the first phase, validation nodes grade the received candidate blocks through consensus and select the candidate block with the most validation consensus. In the second phase, the candidate blocks selected in the first phase undergo binary Byzantine judgment. BA* consensus ensures that the number of honest nodes participating in the consensus is greater than 2/3. If the randomly selected set cannot satisfy this condition, multiple random elections are conducted. As long as there is one instance where the number of honest nodes participating in the consensus is greater than 2/3, consensus can be achieved. Validation nodes for each step of BA* are specified or selected through parallel designation or random drawing to accelerate the consensus confirmation speed.

The steps of BA* consensus calculation

At each step of BA*, the temporary keys for the current step need to be destroyed. The steps are summarized as follows:

1.Generate blocks (Step1)

1) The node checks whether it is the lead node B_i^r.

2) *Generate the message for the first step* $m_i^{r,1} = (B_i^r, ESG_i(H(B_i^r)), \sigma_i^{r,1})$

3) BroadcastB_i^r和 $m^{r,1}$

Where $m^{r,s}$ is the message broadcasted by node *i* in round (*r*,*s*);B^r is the block generated by node *i* in round *r*; *ESG* denotes signing the information with the current temporary key of round (*r*,*s*); *H* is the hash calculation; $\sigma^{r,s}$ is the signature SIG(r,s,Q^{r-1}) used to prove the presence of node *i* in the set of validating nodes in round (*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

Here, the verification node statistically assesses the values issued by the Hierarchical Consensus Protocol. The Binary Byzantine Judgment is a three-step loop in which verification nodes continuously check the received history to see if two termination conditions are met: whether the block is valid or invalid and if it reaches a 2/3 majority vote. If the block is deemed invalid, the consensus system will assess and generate an empty block. To prevent an infinite loop, a maximum total loop count, denoted as m, is set. If after reaching m iterations, the determination of whether it meets one of the termination conditions is still pending, the consensus system will temporarily generate a provisional consensus. In subsequent processes (the following rounds), a final consensus will be formed, confirming 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.

3.2. 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 one 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 will adopt mechanisms like Plasma to achieve horizontal performance scaling based on a multi-chain system. This parallel computing mechanism allows PATH to achieve an extremely high level of state updates per second (potentially in the range of several billion). This significant improvement in performance positions PATH to potentially replace the current centralized cluster hosting capacity.

3.3. 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.

3.4 Quantum-resistant

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.

4. Ecosystem-driven community governance

4.1. 1+N Token Architecture (PATHx/PATH)

The token PATH is the mainnet token of the PATH network, primarily used for functional purposes within the network. The usage scenarios for the token PATH will be detailed in the following sections.

PATHx, on the other hand, is driven by BTC gas and is distributed through airdrops to native asset communities on the BTC network. It serves as the consensus cornerstone of the PATH network. Taking the Ordinals protocol as an example, numerous consensus-driven native asset issuances and Mints have been completed in 2023. In the process of converting the liquidity value of these 'assets,' community consensus has continuously formed. In response, PATH will issue the corresponding cornerstone, PATHx, on the Ordinals protocol. This issuance involves participation from various communities within the Ordinals protocol, and through a dynamic algorithm, PATHx will be exchanged cross-chain to become the mainnet token of Layer2 PATH. Users holding other assets on the Ordinals protocol can use their Token PATH to build ecosystem nodes, allowing them to cross onto the PATH network to support applications in the field of Turing-complete smart contracts without permissions.

4.2. Ecosystem Node

DApps on the PATH network need to connect to ecosystem nodes for operation. In return, the majority of the on-chain transaction GAS fees generated by interacting with the DApp will be obtained by the ecosystem node. This is the core significance of setting up an ecosystem node. Communities holding protocol assets are motivated to build nodes to operate applications where they can use these assets.

4.3. Advanced ecosystem node

The PATH network supports another advanced type of ecosystem node. When smart contracts connected to such nodes allow transactions, they can specify non-PATH mainnet tokens, such as Ordinals assets, to pay for GAS fees. As a credit guarantee, setting up such a node requires a higher pledge of PATH mainnet tokens.

Whether it's a regular ecosystem node or an advanced one, the amount of PATH mainnet tokens pledged determines the number of DApps it can integrate.

5. Milestones

- In mid-2023.12, Launch the first supported protocol, Ordinals PATH/PATHx. PATH enters the cross-chain bridge to the L2 base pool.
- In the middle of 2024.1, Initiate the PATHx -> PATH cross-chain exchange channel and launch the ecosystem nodes.
- In late 2024.1, Launch the mining pool and commence Layer2 PoW mining.

- Late 2024.2, Open-source the mining pool.
- 2024.3, Launch the Layer2 mainnet, supporting all OD ciphertext cross-chains to PATH, and enabling the construction of 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.