Secret Blockchain Technology White Paper

Secret Block Chain white paper V1.0

table of Contents

1.Background

Secret is committed to solving the problems of existing blockchain solutions such as low transaction processing speeds, very expensive and limited computing resources and capacity. Secret proposes a blockchain solution suitable for large-scale transactions and calculations, with high performance, high security, and flexible expansion.

Secret core features include:

1)Original POE (Proof of Equality) consensus mechanism

With high-performance transaction processing capabilities (average TPS 4500+) and low-consumption performance, the POE consensus mechanism provides everyone with the same opportunity to participate, truly breaking the monopoly of blockchain computing power.

2)The built-in decentralized computing power network brings verifiable and unlimited computing power to the blockchain. Smart contracts make it possible to implement largescale matrix multiplications, AI model training and 3D rendering, to enable more real-world commercial applications and achieve computing goals.

3)The chain state synchronization function can effectively, solve the problems of the continuous expansion of block data, and greatly reduce the participation threshold of Secret ´s nodes, to truly allow everyone to participate in blockchain and benefit from it.

The Secret ecosystem is a framework that consists of three technical layers: the data persistence layer, the domain layer and the service layer. The data persistence layer has an account system, a POE consensus mechanism, and smart contracts. The domain layer consists of a cross-chain protocol suite based on smart contracts, a distributed computing system, and a data mapping and storage protocol suite. Finally, the service layer of the platform consists of a complete third-party protocol, API and SDK, a dashboard and related components. The service layer also provides

access to developers and other related parties, realizing the processes of information generation, consumption, and delivery of the entire ecosystem.

2.Account System

There are two types of mainstream account systems in the blockchain industry: asset-oriented (Bitcoin UTXO) and user-oriented (Ethereum). The UTXO model is considered to be relatively safe and efficient. Simple Payment Verification (SPV) can be used to quickly verify transactions. But the biggest disadvantage of UTXO is that it is Stateless, which makes it very disadvantageous to develop applications on the UTXO model. So, Secret has chosen a user-oriented account system.

The Secret account system has two types of accounts: external accounts (controlled by private keys) and contract accounts (controlled by contract codes). No external accounts can deploy code. Users can create a transaction and send a message from an external account with a private key signature. When creating a smart contract, a default internal account is automatically assigned. Whenever the contract account receives a message, the code inside the contract will be automatically run by the virtual machine, allowing it to read and or write the internal storage, send Other news or create a contract. The account consists of five parts:

2.1. Nonce, to prevent double transactions

2.2. Current account balance

2.3. Account control authority public key

2.4. The contract code of the account (if any)

2.5. Account status storage (can be empty)

Ethereum has one-to-one correspondence between private keys and addresses. However, Secret allows one private key to correspond to multiple addresses, and it also allows the address owner to transfer permissions. Each Secret account contains two types of permissions: owner permissions (OwnerKey) and manager permissions (ActiveKey). These two permissions have different functions. Owner

permissions (OwnerKey) can control and manage any other authority, while the manager authority (ActiveKey) can authorize the execution of transfers, contract execution and other operations. An account can have multiple owners and managers. This means that the user can manage the rights of the account, and enhance the security of the account by assigning different rights to the operators of different identity accounts. If the private key of a certain account is accidentally leaked, the operator with Owner rights can log off to avoid losses.

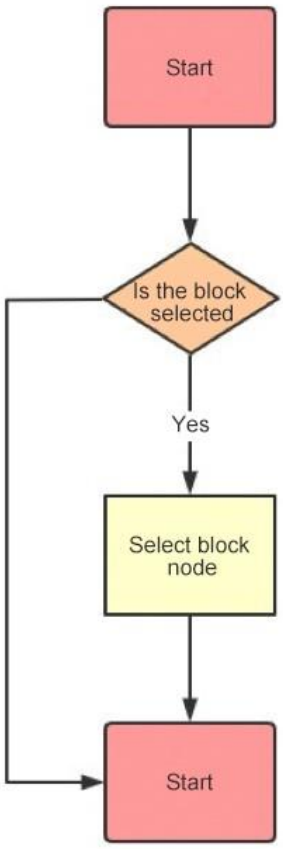
3. Consensus Mechanism

The consensus mechanism is a core part of blockchain and the key to ensuring the continuous operation of the blockchain system. Since the birth of Bitcoin in 2009, after more than ten years of development, the consensus mechanism has derived a variety of consensus algorithms from a single proof of work (PoW), such as proof of stake mechanism (PoS) and delegated proof of stake (DPoS), and practical Byzantine Fault Tolerant Algorithm (pDFT), etc. Each consensus algorithm has its own advantages and disadvantages and needs to be selected according to the actual needs of the blockchain project. Secret hopes to be able to handle large-scale transactions on the blockchain and support rapid verification. Everyone has the opportunity to participate,and so Secret has created a unique POE (Proof of equality) consensus mechanism.

In the POE consensus mechanism, the system will select candidate nodes based on their historical performance. Online rate, bandwidth, computing power and other comprehensive aspects are used as the scoring criteria for nodes. Nodes that meet the conditions will become candidate nodes. The system randomly selects 21 nodes from candidate nodes as block producers every day, and each node has the opportunity to participated in the selection. The POE consensus does not require traditional mining, and it also avoids the power concentration problem of the DPoS algorithm. The POE consensus also takes performance into account and is more decentralized.

3.1 Node Block

In each round of the block generation process, 21 randomly selected block generation nodes record and confirm the transaction. When the block generation node successfully generates the block, it will receive the SIE block generation reward which is generated at the same time as the block. If the block-producing node has frequently crashes or for some reason performs its block-producing obligation for a long time, the system will forcibly cancel its block-producing qualification and let the candidate nodes take their place.



3.2 Selection Mechanism

Block generation nodes are randomly selected by the consensus mechanism according to the latest node scores every day, and every node in the network has the opportunity to participate in block generation. Online rate, bandwidth, and computing power are important indicators of whether a node can become a block producer, so node operators should try their best to ensure that the nodes operate stably and efficiently. We divide nodes into A, B, and C based on daily scores. The higher the node level, the higher the probability of being selected as a block producer.

The specific selection steps are as follows:

Select one node from A-level nodes

Select one node from both A and B level nodes respectively

Select one node from each of the A, B, and C level nodes

Repeat the above steps until enough nodes are selected

Step	Selection range
1	A
2	A、B
3	A、B、C
4	A
5	A、B
6	A、B、C
…	…

For example, when we need to select 6 block producing nodes, we will select 3 from A-level nodes, 2 from B-level nodes, and 1 from C-level nodes. The higher the node level, the higher the probability of being selected. .

Level	selected quantities
A	3
B	2
C	1

The multi-layer block node selection mechanism provides each node with an opportunity to generate blocks, encouraging more and more developers to participate in node building. Nodes that provide high-quality services will have more opportunities to participate, incentivizing node operators to provide better services, which can improve the safety and efficiency of the entire system.

3.3 Participation rewards

Secret blockchain adopts dual token mechanism with built-in native token Sie and fuel token gas. As the fuel token of secret blockchain, gas is used to pay for network transfer fees, smart contract operation and storage fees, so as to realize the resource control of secret network operation and prevent resource abuse.

The stability of nodes is very important for the consensus mechanism of Poe. The system uses a new gas token to give economic incentives to the outgoing nodes and encourage them to provide high-quality services. Non node participants can also deposit Sie and gas tokens into the system mortgage pool to obtain gas token rewards generated when mining new blocks. Specifically, when a new block is mined out, 10% of the gas token reward is distributed to the node of the block, 90% of the gas token reward is distributed to all mortgage users, and the system mortgage pool rewards all users according to the stack allocation.

In the mortgage part, the secret app provides the mining mortgage portal, which provides the function of all secret users to mortgage Sie and gas. After mortgage, the mining mortgage weight of each user – Stack(Stack=$\sqrt{\text{the mortgaged gas} * \text{the mortgaged SIE}}$)is calculated according to the data of SIE and gas in the mining mortgage. The node mining reward is allocated according to the proportion of user weight.

At the same time, all secret users can convert gas into Sie in the secret app, and the conversion ratio is 1:1 (the conversion only supports using gas to convert SIE, but does not support using Sie to convert gas)

In the POE consensus, not only the outgoing nodes can get profits, but also all the SIE and gas token mortgagors can get rewards. We hope that this mechanism can effectively encourage the majority of users to participate in the secret network, so as to make the whole network more huge, secure and decentralized.

3.4 gas allocation rules

4.Smart Contracts

The Secret Virtual Machine (SVM) is fully compatible with the Ethereum Virtual Machine (EVM). It is a smart contract runtime environment with Turing completeness, high security and high scalability. SVM has the following characteristics:

1. Offers a compilation tool with a comprehensive security check mechanism.
2. Supports multiple mainstream languages, such as Python, JavaScript, Solidity and Go, in order to include more developer communities.
3. Improves development efficiency by providing more standard libraries
4. Provides a developer-friendly IDE, online debugging and a compilation environment.

5. SVM allows smart contracts to perform large-scale distributed calculations, expanding the functions and usage scenarios of smart contracts.

The SVM compiler can translate smart contracts written in high-level languages into bytecodes that SVM can recognize. The code consists of a series of bytes, and each byte represents an operation. Generally speaking, code execution is an infinite loop, and an operation is executed every time the program counter increases by one, until the code execution is completed or an error is encountered, or the STOP and RETURN instructions are executed. Operations can access three types of data storage spaces:

1. Stack, last-in-first-out data structure, 32-byte value can be pushed/popped
2. Memory, infinitely scalable byte queue
3. The long-term storage of the contract is different from the stack and memory that are reset at the end of the calculation. The storage content will be kept for a long time.

The formal execution model of SVM code is very simple. When the virtual machine is running, the complete computing state is determined by the tuple (block_state, transaction, message, code, memory, stack, pc, gas). Among them, block_state is the state that contains all account balances and storage. At the beginning of each round of execution, the current instruction is found by calling the PC (program counter) byte of the code. Each instruction has its own definition of how to affect the tuple. For example, ADD pops two elements of the stack from it and pushes them into their sum, then decrements GAS by 1 and increments the value of PC by 1. SSTORE pops the top two elements from the stack, and inserts the second element at the index specified by the first element in the contract storage. Although there are many ways to optimize the execution of the virtual machine through real-time compilation, the basic implementation of the Sesret virtual machine can be completed with a few hundred lines of code.

4.1 WebAssembly (WASM) Web components

WASM is an emerging Web standard for building high-performance Web applications, which can be clearly defined and sandboxed with a small amount of adaptation. The advantage of WASM is that it is widely supported by the industry, so you can develop smart contracts in familiar languages, such as Python, JavaScript, etc.

The WASM component will allow developers to use multiple programming languages to write code and compile it into Secret style WebAssembly. Swasm is a safer subset of WebAssembly, which is a relatively new low-level compilation target for the Web platform. Conveniently, the WASM (and Swasm) module can be used in any JavaScript project.

For most blockchain code, usually more than 75% of the code is not in the smart contract at all — it must communicate with the smart contract in JavaScript. Swasm and JavaScript share a common foundation of binding and module support.

5.Distributed Computing

For the blockchain public chain, the resources allocated to smart contracts on the chain are very limited. For example, storage on the chain is very expensive, and the execution of smart contracts is limited (such as the gas limit of Ethereum, the CPU time limit of EOS etc...) These have hindered the birth of complex smart contract applications. We hope to realize a distributed computing network based on idle candidate nodes to carry out the subcontracting of computing tasks. Those in demand of DAPP computing power can hand over computing tasks to the computing power suppliers in the market to complete. The supplier can be anyone with idle computing power.

5.1 System Architecture

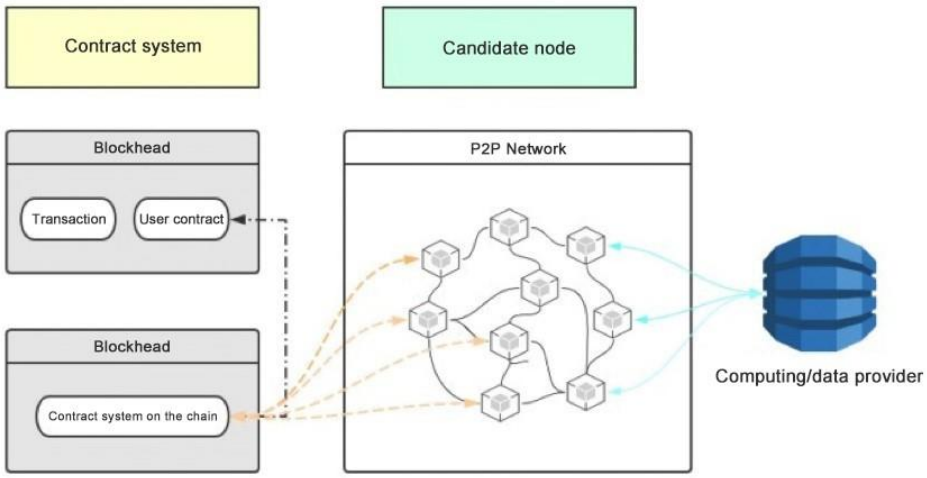
The system architecture can be divided into two layers:

5.1.1 System contract

This consists of a system contract and a management contract deployed on the chain. It mainly includes functions such as processing calculation requests, verification of results, node registration, tokenized mortgages, node status monitoring, and payments.

5.1.2 Candidate node

Candidate nodes are run by third-party users (i.e. node operators) that have implemented the core protocol and are not currently participating in block generation. The client protocol includes several important modules: event monitoring, distributed random number engine, cryptography and group consensus on the chain, requests/computing task processing, etc. The specific modules included depend on which services the user node is willing to provide.



5.2 Processing Flow

The entire agreement follows a basic ´request-reply´ process of the network system. Through the on-chain SDK developed by us, the smart contract (the developer) specifies the data source and data type required by the contract, and sends a calculation request to the system contract in the form of a message call.

This calculation request will be randomly assigned to a candidate node working group, and each member node in the working group will obtain data from the specified data source and parse it according to the specified type and structure. Then the nodes in the group reach a consensus within the group on the data result through threshold cryptography and generate a proof of consensus result. The data and the corresponding proofs that have passed the consensus in the group will be sent back to the system contract in the form of a transaction, and the contract will be triggered to verify the submission group, return result, and result proof.

specific process:

1. The user contract sends a data query request via the system contract message;
2. The system proxy contract triggers events with query parameters;
3. The candidate node client continues to monitor the events defined on the blockchain and will be notified at this time. Ideally, there will be thousands of clients operating. A registered group is randomly selected by using a distributed random engine constructed using a verifiable random function (VRF);
4. Members of the selected group conduct due diligence at the same time, request web api, perform calculations or execute configured scripts;
5. They will reach an "in-group" consensus through the t-out-of-n threshold signature algorithm, and feed back the consensus results to the system contract. As long as more than t members of the randomly selected group are honest, they can get the result of the consensus. The identity and QoS (responsiveness/correctness, etc.) performance of the selected group members will be recorded on the chain for monitoring and data analysis;
6. The system contract on the chain notifies the user that the contract result is ready through the callback function provided by the user contract.

5.3 Reliable Data

The system uses Verifiable Random Equations (VRF) and Threshold Cryptography to drive the selection of safe, unpredictable, and verifiable working groups. Different oracle requests will be processed by randomly selected working groups. No working group or node can predict in advance when and what kind of requests they will process.

The nodes in the selected working group obtain the data and use threshold cryptography to collaboratively generate a proof of data integrity. The proof along with the data result are sent back to the system contract and verified. Any submissions of malicious transactions will fail the verification process and those responsible will be monitored and punished.

5.4 System performance

As described in the system architecture, the entire process is divided into two parts: the system contract and the candidate node. The performance of the system depends on the difficulty of calculation. If the candidate node has calculated the data in advance, only the result is taken. This entire process includes status monitoring, calculation request analysis, data acquisition and result analysis, and team members' collaborative production of proof of data integrity, etc. This can all usually be completed within 1 second. In other words, the performance bottleneck is often in the system contract.

For example, for the system contract, the data result is returned in the next block after receiving the request,this is the minimum delay that can be achieved in theory, and we can do this. For the better-performing Secret blockchain, the delay between receiving a request and returning the request will be reduced to almost real-time。

5.5 system security

In order to prevent attackers from forging a large number of identities to join the network simply and at low cost, node operators must first mortgage and lock a certain amount of network tokens in the system contract before they can join distributed computing network services and earn fees. At the same time, all returned results will be verified in the system contract, and malicious nodes will be detected and punished. Through these methods, Sybil attacks are so expensive that it is almost impossible for them to occur. At the same time, node operators are closely tied to the network, and their interests are consistent with the value of the network token.

6. slave chain status synchronization

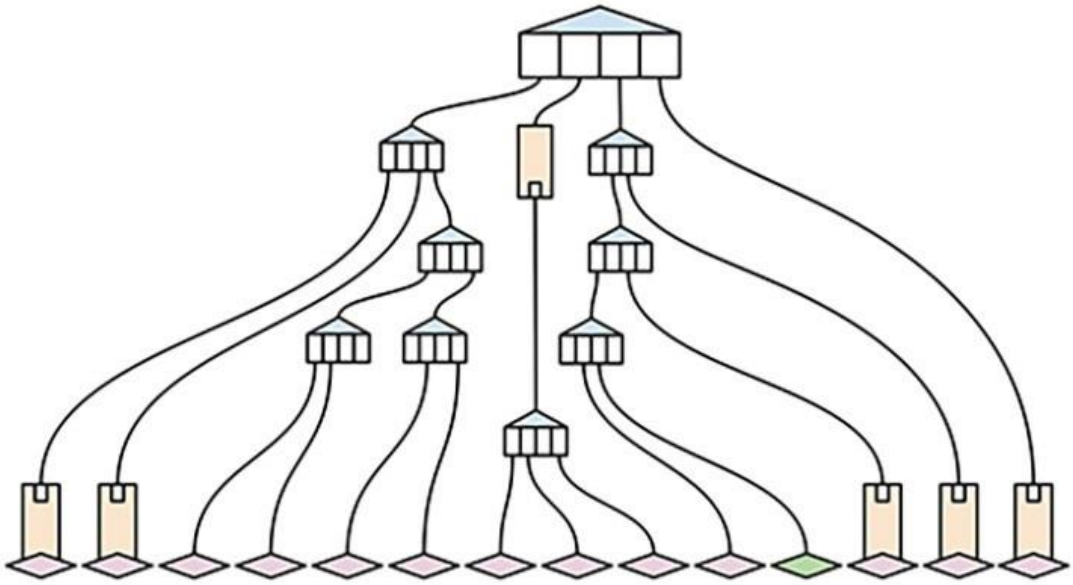
The blockchain is a distributed world shared ledger, and anyone can write data on blockchain. Over time, data on the chain will continue to accumulate. Since there are no measures to limit the growth of data or clean up historical data, the blockchain will become bloated and the consequences are very serious. First, it will increase the hardware cost of the node server and eliminate a batch of low-performance nodes, resulting in a decrease in the number of nodes, affecting the degree of decentralization of the blockchain. Secondly, the computing power of the node server will be constantly squeezed, which will lead to increased verification difficulty, longer synchronization times, increased network delays, and decreased TPS. Excessive data volume will also greatly increase the entry barriers for new nodes. For example, the block data of Ethereum has exceeded 1 TB, and it takes a whole week for all data to be synchronized. In short, data inflation will deteriorate the health of the blockchain.

6.1 Solutions

In order to completely get rid of data expansion, we have created an original slave chain state synchronization mechanism, to be used for semi-real-time data retrieval, so that newly added nodes can quickly synchronize the latest block´s status. There is a main chain and multiple slave chains on the Secret platform.

The main chain is responsible for maintaining the decentralized network and processing transactions. The slave chain is responsible for generating the Merkel-Patricia Tree and witness data of the latest status of the current block. Witness refers to one Kind of additional data status different from block data. This mechanism enables the new node to retrieve and verify all account and storage data without downloading all the block data since the creation block and locally reassemble the final status of the block.

The code and status of the smart contract also exist on the leaves and branches of this (Merkle-Patricia) tree. Through continuous hashing, a root node hash is finally obtained. If you want to know whether two copies of a state trie tree are the same, you can simply compare the root node hash values. Maintaining a relatively safe and undisputed consensus in a "standard" state is the essence of blockchain design.



6.2 Actual Effect

This mechanism allows the downloading of the entire blockchain state without downloading all intermediate Merkle proofs that are regenerated locally. This greatly reduces the network load

- Ingress bandwidth 0(accounts * log account + SUM (states *log states))

(Merkle trie node) reduced to O(accounts + SUM(states)) (actual state data).

- Egress bandwidth 0(accounts * log account + SUM (states * log states)) * 32 bytes (Merkle trie node hash) reduced to 0(accounts + SUM (states)) / 100000 bytes (The number of 100KB chucks used to cover this state)
- Round trip time 0(accounts * log account + SUM (states * log states)) / 384 (status retrieval packet) reduced to 0(accounts + SUM (states)) / 100000 bytes (The number of 100KB chucks used to cover this state)

7. Toolkit

The Secret Ecosystem provides a toolkit with many functions for different types of participants. Through the toolkit, everyone can easily use blockchain.

- For DAPP developers: A complete Secret test chain and related cloud storage resources can be obtained from the developer community, including detailed documentation, user examples, multi-platform and cross-language SDK and API. Developers can use the test chain for development and debugging without deploying a node server. After completing the product testing process, simply replace the relevant address and application key to connect the DAPP to the main chain.
- For DAPP administrators: real-time access to user statistics and the number of tokens obtained by DAPP users can be made through visual charts.
- For users: users can use one single account to access all DAPPs in the Secret ecosystem. The universal wallet controls all assets, thus enabling the free transfer of digital assets. The block explorer allows users to interact with blockchain data in a simple way, track the balance of the wallet address, and use precise search terms to search for transactions, etc.

8. Conclusion

Secret gives developers the ability to create any transaction type and application by providing a highly versatile Turing complete programming language. The centralized computing power network provides unlimited verifiable computing power for smart contracts, which can support various complex computing scenarios. In the future, there will no longer be the problem of very expensive

and limited computing resources and capacity. This makes DAPP large-scale implementation of applications a reality.

Despite the fact that blockchain technology still has limitations to overcome in terms of largescale services, it has huge potential to completely change this world full of privacy leaks and data monopolies through a transparent and efficient token economy. Our team Understands the problems that need to be solved and improved, and we believe that the users of Secret´s ecosystem will be the pioneers of blockchain innovation. We know that everyone's contribution will not only make Secret´s ecosystem better, but will also ultimately make the world a better place.