MW
Mirror World

MIRROR WOLRD NETWORK

White Paper

V1.05

Table of Contents

PREFACE

Let's start by reading a report of network data:

According to a report issued by IDC in "Data Age 2025", the annual data generated worldwide in 2025 will increase to 175ZB from 33ZB in 2018, which is equivalent to 491EB of data per day (converted to our commonly known TB unit, it is 491EB × 1024T × 1024T = 514.8 million T), I believe this number will make everyone feel incredible.

Most of the current data is stored in a centralized manner in the data centers of major Internet companies around the world, which brings a series of problems: expensive, not permanently stored, prone to data leakage, privacy snooping, data abuse, etc. At the same time, the centralized storage code is not open source. Potentially fatal vulnerabilities cannot be detected, and even the data is subject to tampering. More importantly, cloud storage with high security and high availability is expensive and the pricing is not transparent.

To this end, we choose to embrace the future by changing the technology.

I. OVERVIEW

Mirror World Network is a file system that combines blockchain and distributed storage. It aims to provide high-quality and low-priced file storage services and build easy-to-use and usable distributed applications.

The system uses peer-to-peer P2P network, DHT distributed hash table, data encryption sharding, IPFS network protocol, multi-chain incentive mechanism block exchange and other technologies to build global high security, high privacy, high availability, multi-chain deployed distributed storage ecology. Its friendly and inclusive features make all kinds of public chains, storage networks, and personal nodes available for deployment and operation.

II. DESIGN PRINCIPLES

2.1 Node unreliability assumption

Refers to a loose but highly robust network organization structure that allows single points of failure and allows nodes to be unavailable for a short period of time.

2.2 Ownership and privacy

The data owner has ownership and full access to the data, and the data is encrypted and private. Other roles can access and use data only after the owner has authorized it.

2.3 Quantifiable contribution

Contributions of all parties involved in the system should have corresponding quantitative standards and observable contributions. For example, PoST and PoR are used as the quantified proof of storage space and storage time.

2.4 Final state consistency

Data objects are allowed to be in different states at different nodes, but their states can quickly converge to obtain network-wide consistency.

2.5 Monitorable and recoverable

It can detect the availability of the entire network and the entire network status of data objects, and autonomously repair to a certain degree according to the policy.

2.6 Auditability and supervision

With the knowledge and consent of the data owner, a certain degree of supervision and audit can be performed in certain specific areas or scenarios.

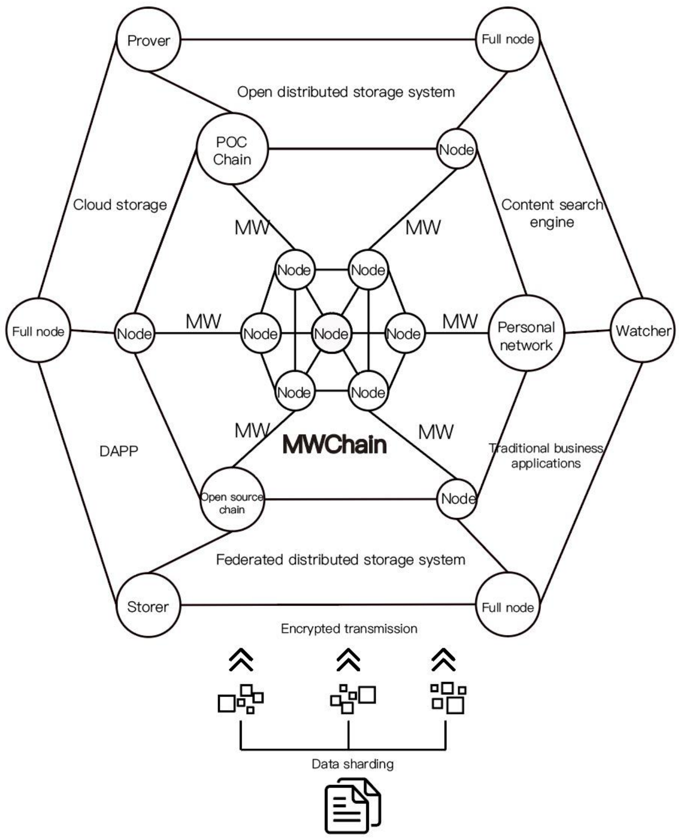
2.7 Extensible API

Highly extensible and easy to use API.

III. THE SYSTEM ARCHITECTURE

- Mirror World Network promotes openness, tolerance and legality.
- The agreement consists of the following parts: role, network, data, contribution quantification, reward and punishment, and multi-chain.
- Proof of Replica (PoR) and Proof of Storage & Time (PoST) are quantified vouchers for data backup and storage duration.
- Data fragmentation, multiple backups, and data erasure to ensure data security and availability.
- Built-in IPFS protocol, forms a multi-chain ecosystem with various existing public and blockchain networks, and completes data and value transmission.
- MW-PAIR and MW-UTXO credit frameworks can meet the auditing and regulatory needs of enterprises or regulators.

Overview of Mirror World Network



IV. ROLE DEFINITION

Mirror World Network defines the roles of storage network participants according to their functions. Multiple roles can be assumed by the same entity node. All nodes need to be evaluated according to their credit PoC (credit proof).

Node type	Miner	Watcher	Storer	Prover
Full node	√	√	√	
Storage node			√	
Observation node		√	√	
Prove node		√		√

/ Role and node relation table /

4.1 Data object

The data body is the data object before sharding. Except for sharding, it cannot be separated individually, and it is unique to external systems.

4.2 Data owner

The data owner is the owner of the data body. The data owner signs the data body and has the right to inspect it at any time to ensure that the data is truly securely stored.

The data owner can make different levels of requirements for secure storage based on the importance of the data. The Mirror World Network will select the appropriate data security policy to store the data body according to the corresponding requirements. At the same time, a suitable storage node is selected to ensure that the storage requirements of the user are met. Of course, in the face of higher levels of data security, data owners also need to pay higher data storage costs.

4.3 Storer

Storer providing disk space to store data in order to get corresponding compensation. Storers need to accept random checks from data owners or watchers to provide proof of storage. For example: Accept PoST verification to prove that the data has been stored on the disk within the agreed time. Note: The "storage node" referred to below is the physical node running the storage client.

4.4 Watcher

The watcher needs to observe and detect the state of the data body in the entire network, detect whether the stored data body meets the security policy, and repair security defects, so the watcher needs to be stable online. As an indispensable role for rapid network-wide convergence, watchers are the best candidates for data volume indexing services.

Watchers occasionally perform heartbeat detection on storage to ensure data availability. You can also accept the commission of the data owner to help the data owner initiate data verification to ensure that the data is safe and available. Based on performance considerations, most of these tasks are done off-chain. The decentralized rights and network organization structure can further ensure the security of data objects and reduce the possibility of malicious attacks on the network. Therefore, we hope that the watcher is an independent node, so that it can form a restrictive relationship with the block producing node and the data storage node.

4.5 Miner

It is the so-called "miner" in the blockchain network. The miner needs to run the client (command line or GUI) to save all block information and undertake the work of processing transactions and packaging blocks. The performance index of the stability, connectivity, and throughput of the Mirror World Network has a great relationship with the memory and calculation speed of the block producer. Therefore, the miner is generally served by the full node to ensure stable online, high throughput, and high processing efficiency. Only full nodes connected to the Mirror World Network can compete for block rights, and the block consensus mechanism will choose PoS or DPoS.

4.6 Prover

Provides proof for the on-chain data to make it credible. The certification data and original data objects provided by the prover will be linked together and recorded on the chain, which is traceable and non-tamperable.

Certifiers serve as credible organizations or government agencies in most real-world scenarios. For example, in the field of data copyright, the most authoritative certifier is the State Administration of Copyright. The certifier node is connected to the State Copyright Administration as a proxy node of the Copyright Administration in the Mirror World Network to provide corresponding copyright certification services. In the case of data storage, the witness is the notary office and the judiciary. After they have been notarized, the on-chain data has social credibility and judicial effect.

4.7 Full-Node

A full node is a physical node that runs a client of the Mirror World Network, can be stable online, and has good network bandwidth and processing performance. The full node has the minerrole turned on by default and can selectively open the storage and watcher roles based on its own hardware.

V. NETWORK TOPOLOGY

We need to build a peer-to-peer network with a large number and nodes joining and exiting at any time. Therefore, a good routing table maintenance and lookup algorithm is very important. We prefer to use the Kademlia protocol (hereinafter referred to as Kad) [1] as the basis to build a P2P peer-to-peer network (Chord algorithm is also an option). Kad's Distributed Hash Table, based on the XOR algorithm for distance measurement, greatly improves the speed of routing queries. This is very necessary for the Mirror World Network with a large number of storage nodes. The implementation of Kad network will also be divided into two steps. First, we will build a P2P network based on a simple routing table and complete the development of Kad network while opening the storage node client.

The node list maintenance of the K bucket in the Kad protocol exactly meets our online requirements for nodes, but in the future, it may be based on the credit rating of the nodes in the PoC as a weight value for sorting and swapping out to help watchers choose the appropriate nearest node Adjust the data distribution.

VI. DATA OBJECT OPERATIONS

6.1 PRCDO = (Put, Get, Watch)

6.2 Put (data) → key

The client executes the Put protocol to store data, and the key is the unique identifier of this data.

6.3 Get (key) → data

The client uses the data unique identification key to execute the Get protocol to retrieve the data.

6.4 Watch

The watcher executes the Watch protocol to verify the stored data and synchronizes the entire network status of the data object. According to different security policies, we have repaired abnormal situations such as data loss, data errors, and unavailability of storage.

VII. DATA STORAGE

The client initiates a request to store data, and the request is recorded to the Store-Book.

The client pays the storage fee, and the Mirror World Network returns the matching storage node (Storer).

The client uploads the file to the storage node.

After receiving the data, the storage node updates the global state of the Store-Book and the data object (Bean-Book).

Broadcast the data backup task (Replica-Task) to the network according to the security policy.

The remaining storage nodes perform data backup and check whether the number of copies defined by the security policy is met. If not, they continue to broadcast data backup tasks to the network.

VIII. DATA RETRIEVAL

The client initiates a data retrieval request, and Mirror World Network obtains the latest data object from the object ledger (Bean-Book) and returns it to the client and synchronizes the data retrieval request to the storage node.

In active mode, the client establishes a connection with the storage node and obtains data from the storage node. In passive mode, storage nodes push data to clients.

After the storage node retrieves data from the client, it updates the Store-Book.

After receiving the data, the storage node updates the global state of the Store-Book and the data object (Bean-Book).

After the client retrieves the data, the client will update the Proof-Book to prove that the storage node does hold the data object.

IX. DATA INSPECTION

The client (Client) or the watcher (Watcher) randomly generates a verification code C according to the time and records the random verification transaction to the proof book (Proof-Book).

The Mirror World Network requires the corresponding storage node (Storer) to generate a corresponding storage certificate M according to the check code C.

The storage node (Storer) provides the storage certificate M to the client (Client) or the watcher (Watcher) for verification within a limited time.

The client or watcher updates the Proof-Book after verification.

After the verification is successful, the Mirror World will generate a reward transaction (Reward-Book) and unlock some storage rewards to the storage node (Storer).

Merkle tree [6] and ZH-SNARK [7] are introduced to allow storage nodes to perform storage proof. Random checks of proof of storage are initiated by the data owner or watcher. For details, see PoR's [Backup Proof] and [Storage Duration Proof]. Watchers need to regularly check data objects on the entire network in accordance with security policies [3.6 Data Availability]. Maintain the consistency of the entire network status of the data, and also have the obligation to repair existing or potential security and availability issues (such as: ① data pieceare lost or unavailable, ② the storage is unavailable for a long time and has exceeded a preset threshold) .

X. STATE CONVERGENCE

The time consuming of backing up a shard of data, as well as the time consumed by the watcher to examine the data and adjust the data distribution, can be considered as the convergence proof problem of the data objects. The following is the summary proof.

Assume that there are N nodes in the network, and the time for storing a shard of data is St. In extreme cases, after the N-1 queries have been made, the last node in the network is considered to be available. The time complexity of backing up a data file is O (N) + St. St is a constant when the network is stable, so the time complexity can be simply considered as O (N), which is the total time consumption to find available nodes in the network. This is intolerable for a network O (N) with frequent joining and exiting nodes. However, the introduction of k-buckets in the Kad network can help reduce the overhead of querying available nodes. Assuming t is the target query node, since each query can obtain information from k buckets closer to t, this mechanism ensures that each recursive operation can obtain at least the effect of halving the distance, thereby ensuring that the entire query process is feasible and fast convergence, the rate of convergence is O (logN).

XI. DATA SECURITY

11.1 Data encryption

The data file is encrypted (AES-256-CTR) by default in the client and stored on the storage node. This means that the data store cannot view the contents of the file. For sensitive data, the data owner can choose to use hardware encryption to generate encrypted file data before storing it in the Mirror World Network.

11.2 Data body fragmentation

The data body fragmentation (referred to as "data fragmentation") strategy and security policy are closely linked. If the data owner has high requirements for data security, fragmentation can ensure the security of the data to a great extent. In order to ensure the availability of data sharding, we introduce data erasure.

We don't think that storage nodes will cheat for files that are too small. Deleting data files by storage nodes and only retaining the corresponding R certificate will not bring significant economic benefits. In most cases, the performance bottlenecks of ordinary storage nodes are bandwidth and disk I / O, which means that the disk space of ordinary storage nodes is not full of data fragment files. But too many small files do slow down the reading and writing of data. This problem can be further solved by relying on the Mirror World Network to provide high-performance parallel data processing.

11.3 Multiple backups

Suppose there are b storage nodes in the Mirror World Network, and the data is sharded into p copies. The number of backups for each data shard is n. The formula for the probability that the data can be successfully retrieved is as follows:

$$R_S(b{,}p{,}n) = \frac{\left(\dfrac{b-p}{n-p}\right)}{\left(\dfrac{b}{n}\right)}$$

b: number of storers in the network

p: number of data piece

n: number of data backups

```
double fac(int p){
    return p == 0 ? 1  : approximation(p * fac(p-1));
}

double choose(int h,int k){
    return fac(h) / fac(k) / fac(h-k);
}

double rs(int b,int p,int r){
    return choose(b-p,r-p) / choose(b,r);
}

double retrieve(int boxerCount, int pieceCount, int replicaCount) {
    return rs(boxerCount,pieceCount,replicaCount);
}
...
```

/Code segment/

```
  Boxer    Piece  Replica   Retrieve
-------------------------------------------------
  100       10       10      5.776904234533874E-14
  100       10       50      5.934196725858287E-4
  200       10       50      3.7276043023296E16
  200       50       90      5.7872010853195E44
  300       80       90      4.094234910939596E131
  500       50       200     3.146459521303754E45
  ...
```

/ Retrieval probability /

11.4 Security policy

The most basic security strategy is to have at least three copies of one shard of data in the usual way of data disaster recovery: one copy at the same node or adjacent nodes, one copy at nodes in different regions, and one copy at nodes across countries. However, higher security policies mean more storage space and more complex data "observation" and "adjustment". In the Mirror World Network, data owners are allowed to define data security policies according to their own requirements. The currently allowed parameters: data backups, data shards. The security policy will directly affect how the watcher repairs the lost data, and it will also affect the speed of data object convergence in the entire network.

XII. DATA AVAILABILITY

12.1 Data erasure

Erasure Coding (EC) [2] is a data protection method that divides data into shards, expands, encodes, and stores redundant data blocks in different locations, such as disks, storage nodes, or Other geographic locations. In order to ensure the availability of data without excessively occupying storage space (which can increase the space utilization of storage nodes), the Mirror World Network performs data erasure processing on data volume shards.

Reed-Solomon (abbreviated RS) code [3] is a more commonly used type of erasure correction code. It has two parameters n and m, which is denoted as RS (n, m). n represents the number of original data blocks, and m represents the number of check blocks. The following is the performance comparison between full backup and RS erasure coding. For specific algorithm implementation, please refer to references [4] and [5].

Types	Disk utilization	Calculate consumption	Network consumption	Recovery efficiency
Full backup (3 backups)	1/3	very low	Lower	Higher
RS Erasure code	n/(n+m)	High	Higher	Lower

12.2 Distribution adjustment

Watchers will continuously adjust the backup and distribution of data to ensure that the current data file is secure and has at least one accessible resource.

XIII. CONSENSUS AND BLOCK PRODUCTION

We believe that the PoW consensus in the Bitcoin network produces blocks, although it shows people a simple and clear economic incentive framework and consensus mechanism, which can guarantee a non-master distributed network to work well. However, as miners start to use expensive hardware equipment and further hardware "arms race", they consume a lot of power and computing resources just to compete for block rights. We think this is not only a waste of resources, but also excessive consumption of hardware resources Add a lot of e-waste. We hope to provide a low-consumption consensus block algorithm while ensuring the security of the blockchain network.

13.1 Consensus production

The multi-chain consensus block generation method consists of a transaction package (Tx-Bundle) and a Mirror World Network Block (MW Block). This method allows each Mirror World mining pool to execute different consensus algorithms internally. A transaction package contains the transaction records in the Mirror World mining pool to which it belongs. The full-node generates a Mirror World block containing different transaction packages and publishes it to the network. Each transaction package (Tx-Bundle) needs to contain the identity information of the Mirror World mining pool and nodes: Node-ID, Pool-ID, Area -ID.

A full node can only connect to one Mirror World mining pool. Nodes connected to the Mirror World chain (ie, Mirror World mining pool 0) can be packaged to generate Mirror World blocks. In the future, we will explore how to make Mirror World mining pools individually packaged into blocks. A feasible implementation idea is to deploy at least one proxy node (MW Agent) connected to the Mirror World chain in each Mirror World mining pool.

13.2 Information on the chain

Not all data and information need to be chained, especially most data and operation results in the Mirror World Network will not be chained. For example, the data file will not be stored on the chain. The object URI stored on the chain is a pointer to the currently available resource address.

In addition to basic block information, the following are on-chain: account transactions, object data, storage transactions, and certification transactions. It is worth noting that a storage transaction will correspond to one object data but may lead to one or more reward transactions (PoST-based storage reward distribution algorithm).

XIV. CONTRIBUTION QUANTIFICATION

The Merkle tree [6] and ZH-SNARK [7] are introduced to form PoR (Proof-of-Replica) and PoST (Proof-of-Storage & Time) as quantified vouchers for storing data. Storage nodes with high credit ratings allow the use of PoR to provide proof in a short period of time, and lower credit ratings will require PoST to provide proof of storage duration.

14.1. Proof-of-Replica

The data owner can request a corresponding backup certificate from the Mirror World Network at intervals:

The data owner generates a checksum C based on time and sends it to the Mirror World Network.

The storer needs to find the corresponding data slice according to C and generate $\vec{M}$ (Merkle check tree).

If the verification passes the Mirror World Network, the Store-Book and Reward-Book will be updated, and part of the reward of the transaction in the Store-Book will be unlocked as a storage reward.

14.2 Proof-of-ST (storage and time)

Although PoR can guarantee that the data store will save the data at least once, but it is inevitable that the perpetrators will cheat. Consider the following scenarios:

After the storer has backed up the data as required for the first time, the storer calculates its Merkle test number for all data pieceand possible split sequences and deletes the data fragment file to save only the Merkle check tree.

After the storer receives the proof instruction, it requests other nodes that have saved the data backup to obtain the data, and calculates the corresponding $\vec{M}$ (Merkle check tree) of C.

In the above scenario, perpetrators can obtain storage rewards with extremely low computation and storage costs. Therefore, PoST is introduced to ensure that as long as the data store does not store the data shard file, the Merkle check tree cannot be calculated correctly, and the storage reward cannot be obtained:

After data slicing, an entropy sequence S is generated, and then S and data slicing are used to generate a hash value R.

At regular intervals, the data owner sends Sx (time-based entropy value, globally unique) to the Mirror World Network. The storer needs to calculate Rx based on Sx and corresponding data pieceand generate a corresponding Merkle check tree based on Rx.

With the pre-entropy sequence, it is also possible to perform data inspection by the watcher agent. The data owner can provide a part of the entropy value sequence to the watcher, and the watcher will complete PoST's proof verification. In order to execute more securely, smart contract will be relied on to implement the agent check logic in the future.

14.3 Proof-of-Credit

In the Mirror World Network, the credit certificate is tied to the account. Based on the Conch CPOS scoring system [8], the emphasis of the calculation formula will be different according to different roles:

Storage nodes: total storage, storage duration, online duration, and punishment.

Full node: maximum transaction processing volume, block production speed, fork convergence speed, online duration.

Observation nodes: index service performance and online duration.

Data owner: The amount of stored data and transaction volume.

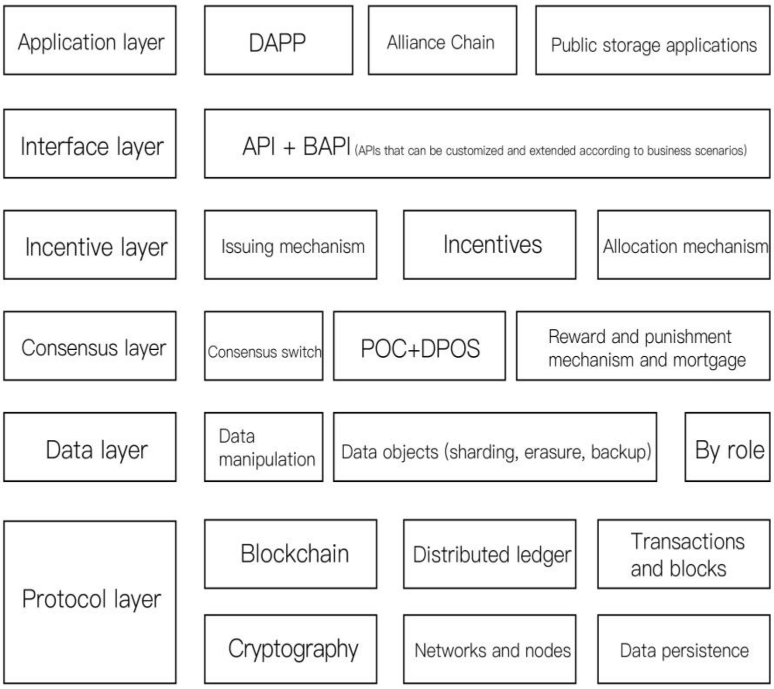
Certifier: The amount of proof.

XV. MW-TOKEN

MW is a built-in encrypted digital token in the Mirror World Network. It is mainly used to stimulate the storage ecosystem built by the Mirror World Network, reward the role that contributes more to the Mirror World, and use MW coins as a means of economic punishment to avoid malicious nodes from doing evil and the infinite loop logic that may appear in smart contracts Bombs etc. At the same time, MW also acts as an anchor native token under a multi-chain (Multi Mirror World mining pool) structure. At the same time, MW has a transaction burn function. While users pay MW as storage costs, we will burn MW according to a certain percentage of the circulation and transfer some MW to the black hole address to play a deflationary function.

XVI. MW BLOCKCHAIN (MWBC)

Nodes and networks: MWBC will use a modified version of the Kad protocol to form a point-to-point peer-to-peer network. In order to quickly form a stable network and a sufficient number of full nodes in the early days, Mirror World will release a storage and mining integrated mining machine MW-BOX.



Application layer | DAPP | Alliance Chain | Public storage applications

Interface layer | API + BAPI (APIs that can be customized and extended according to business scenarios)

Incentive layer | Issuing mechanism | Incentives | Allocation mechanism

Consensus layer | Consensus switch | POC+DPOS | Reward and punishment mechanism and mortgage

Data layer | Data manipulation | Data objects (sharding, erasure, backup) | By role

Protocol layer | Blockchain | Distributed ledger | Transactions and blocks | Cryptography | Networks and nodes | Data persistence

/Functional module/

XVII. MULTI-CHAIN ECOLOGY

The mining pool where the Mirror World Network is deployed constitutes a multi-chain ecosystem.

17.1 Transaction Exchange

The transaction bundle (Tx-Bundle) inside each subchain / parallel chain contains all the transaction types defined. The final transaction package will be packaged into a Mirror World block (MW Block) and broadcast to the network.

17.2 Value exchange

Each sub-chain / parallel chain can define internally circulating tokens. Depending on MW , transactions of different tokens can be completed between the sub-chains / parallel chains to form a value exchange. In the future, cross-chain exchange of information and value will also be introduced. Hash locks and smart contracts based on homomorphic channels should be better choices.

XVIII.CREDIT GRANTING FRAMEWORK

At present, many commercial scenarios still require a certain degree of supervision and auditing. Such supervision and auditing should be conducted with the knowledge of both parties.Mirror World Network provides a basic framework to help complete account credits and audits. It is convenient for individuals, enterprises, and regulatory agencies to access and use the user's credit data in daily use.

18.1 Account Credit

The grading and auditing of information needs to be done with the knowledge and authorization of the data owner. Drawing on the ideas of BIP39 [9] multi-level deterministic wallets to allow data owners to grant credit. We will refer to the path definition method of BIP44 [10] and the discussion in EIP85 [11]. The concept of introducing accounts constitutes the following path: m / purpose '/ coin_type' / account '/ change / address index

18.2 Audit

Due to the anonymity of both parties to the transaction, even if the block and transaction information is public, the information cannot be audited well. Such as transaction information, the auditor needs to obtain the authorization of both parties to the transaction to unlock the account information and compare the transaction information and account information. The auditor needs to repeat the credit process to obtain the credit of the two parties' accounts to complete the audit of the transaction. All audit results will also be recorded on the chain. Of course, for commercial use, you can further design models such as batch account credit, especially in the scenario where the C-end user trusts the B-end to avoid the lengthy process of auditors repeatedly applying for credit.

18.3 KYC

The user identity is not identified in the Mirror World Network, and the corresponding identity can be determined by the upper-level users.

18.4 Information Classification

Information classification is similar to data read and write permission control. Different data objects are encrypted with different derived keys (different HD paths). In this way, the derived key can only decode this shard of data. This will lead to more complex private key generation and data encryption logic, and further implementation solutions will be discussed in the future.

XIX. MALICIOUS ATTACKS

19.1 51% Attack

This is a problem that all blockchain systems will face, and it is impossible to completely avoid this problem. In order to reduce the chance of being attacked, Mirror World Network uses PoC and DPoS to generate blocks.

19.2 Sybil attack

Mirror World Network requires that a transaction be verified against at least 3 nearby nodes. Unless an attacker can calculate the network topology near the attacked node and disguise it as a nearby node, this solution can greatly reduce sybil attack. As more nodes join, the probability of a sybil attack will also be greatly reduced. The address list of the official nodes in the early days will be public and built into the released client. As long as the three inspection nodes include an official node, sybil attacks can be effectively avoided.

19.3 Data fraud

If a storage node obtains a data file from a nearby data storage node and calculates the correct Merkle verification path within a very short time after receiving a random verification request, the node can pass the verification and obtain storage compensation. The use of PoST can greatly reduce this probability, and the nodes that perform data fraud will be punished accordingly, such as: lowering their credit rating, or being never allowed to access the Mirror World Network.

19.4 Data hijacking

Refusing to provide the last data shard makes it impossible to restore the data volume object before the shard, thereby extorting high costs from the data owner. In the Mirror World Network, data is fragmented and backed up. Storage nodes do not necessarily know which shard of data is the last shard. Even if the storage node knows, data can be retrieved from other nodes in sharded backups. Unless all the storage nodes that own the shard are controlled by a malicious attacker, the probability is lower in this case. As the Mirror World Network becomes more discrete, the probability of data hijacking will be further reduced.

19.5 Data erasure

When the storage node thinks that the current data shard storage reward amount is too low or simply no longer wants to store the data shard, it chooses to delete the data shard from the disk. Multiple backups of the Mirror World Network can reduce the loss to data owners in this case and can also adjust the reward strategy of PoST to issue a large amount of storage rewards when the storage time limit is reached. The most effective is to punish the storage node, reduce its credit rating, and reduce the possible future revenue of the node.

XX. TECHNICAL DIRECTION AND POSSIBILITY

According to the CAP theory, we must make trade-offs between Consistency, Availability, and Partition Tolerance. We assume a strategy: N = number of copies, W = number of write copies that must be completed for a successful write operation, and R = number of copies required for a successful read operation. The strategy is that we set the value of NWR to get a choice of CAP. For example, Amazon chose N3W2R2, which means that when two data copies fail, the affected part of the data becomes read-only and can no longer be written. In the future, we will continue to study and reference the current leading cloud storage service providers (Amazon, Facebook, Aliyun) to optimize to ensure better data availability based on better performance.

In order to reduce the overhead of computing resources and network I / O during data erasure, after implementing the classic RS erasure code, we will consider whether to implement the SIMD technology acceleration and LRC (Locally Repairable Codes) erasure algorithm according to actual needs, such as Facebook. And XORing Elephants proposed by the University of California [12].

We have connected the new decentralized WEB application protocol [SOLID] invented by Sir Tim Berners-Lee, the inventor of the World Wide Web, to the Mirror World Network, and successfully tested it to meet the needs of individuals in the future. To truly return data ownership to individuals and solve privacy issues.

XXI. VISION APPLICATIONS

Establish a global distributed storage infrastructure.

Mirror World Network is different from most decentralized application ideas. We believe that the world needs alliances and order. For example, the rapid development of China in the past three decades has brought a new order to the world and promoted various technical Competitive and enterprising. With the advent of the Internet of Everything and the 5G era, the digitalization of human life will explode the demand for global storage facilities. We plan to build a storage facility node to meet the needs of 100 million people between 2020 and 2023 Group, we hope that global storage, open source public chains, and enterprises and individuals with storage resources can join the Mirror World Network to provide humanity with globalization, high security, high privacy, high availability, permanent, low cost Storage services.

XXII.THANKS

Thanks to IPFSGALAXY, IPSOU, Ben, Sharder and other partners for their help. This article draws on and references the design of distributed web system IPFS [15] and distributed cloud storage Storj [16] and the code on Github, which is specifically proposed and thanked.

XXIII. REFERENCES

[1] I. Baumgart, S. Mies. S/kademlia: A practicable approach towards secure key-based routing, (2007). http://www.tm.uka.de/doc/SKademlia 2007.pdf.

[2] Wiki. Erasure Code. https://en.wikipedia.org/wiki/Erasure_code

[3] James S. Plank*. A tutorial on reed-solomon coding for fault-tolerance in raid-like systems, (1996). http://web.eecs.utk.edu/~plank/plank/papers/CS-96-332.pdf.

[4] James S. Plank. Tutorial on Erasure Coding for Storage Applications, (2013)http://web.eecs.utk.edu/~plank/plank/papers/2013-02-11-FAST-Tutorial.pdf

[5] Wiki. Reed–Solomon Error Correction. https://en.wikipedia.org/wiki/Reed–Solomon_error_correction

[6] R.C. Merkle. Protocols for public key cryptosystems, (April 1980). http://www.Merkle.com/papers/Protocols.pdf

[7] Zcash Blog. Explaining SNARKs. https://z.cash/blog/snark-explain.html

[8] CPOS. Conch Chain. http://www.conchchain.org/

[9] Bitcoin. bip-0039. https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

[10] Bitcoin. bip-0044. https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki

[11] Ethereum. Eips. Standardizing HD wallet paths for Ethereum Standard Tokens. https://github.com/ethereum/EIPs/issues/85

[12] University of Southern California & Facebook. XORing Elephants: Novel Erasure Codes for Big Data. https://arxiv.org/pdf/1301.3791.pdf

[13] Yung, M., Dodis, Y., Kiayias, A., Malkin, T., & Bernstein, D. J. (2006). Curve25519: New Diffie-Hellman Speed Records. In , Public Key Cryptography - PKC 2006 (p. 207).

[14] KCDSA Task Force Team. The Korean Certicate-based Digital Signature Algorithm. http://grouper.ieee.org/groups/1363/P1363a/contributions/kcdsa1363.pdf

[15] IPFS. https://ipfs.io

[16] Storj. https://storj.io

APPENDIX

Appendix A Network Operation Definition

1. PING-online node detection.

2. STORE-Stores key-value pairs to DHT.

3. FIND NODE-returns the K nodes closest to the requested key value from its own bucket from the DHT.

4. FIND VALUE-Returns the value of the corresponding key from the DHT.

Appendix B Data Operation Definition

1. PUT-stores data.

2. GET-retrieve data.

3. WATCH-check and adjust data.

3.1 SETUP-Initial setup to generate a check code.

3.2 PROVE-Generate Proof.

3.3 VERIFY-proof of verification.

3.4 REPAIR-adjusts the data distribution.

Appendix C Transaction Operation Definition

1. ADD ORDER-Generate a trade order.

2. MATCH ORDER-match the transaction order.

3. PROC ORDER-Process a trade order.

4. REPAIR ORDER-correction of trade orders.

5. DROP ORDER-void trade order.