White Paper v1.9.9

November 2020

Crust

I Overview 1

A) New Chance for Decentralized Systems and Cloud Computing Services 1

B) Crust Brief 1

II TEE-based Meaningful Proof-of-Work Mechanism: MPoW 2

A) TEE (Trusted Execution Environment) 2

B) MPoW Mechanism 3

III Crust Network 4

A) Node Functions 4

Meaningful Work 4

Work Report 4

Node Environment Verification 4

Node Enrollment 5

Node Service 5

Network Topology 5

B) Technical Architecture 5

MPoW 6

GPoS 7

DSM 9

IV Technical Implementation 11

A) Crust Remote Attestation 11

B) Distributed Storage 12

C) Proof of Data Storage 12

D) Proof of Empty Disk 13

E) Data Sealing 13

F) Incentives for node retrieval services 13

G) TEE Update 14

H) Attacks and Threats 14

1. SCA of SGX 14

2. SGX-ROP Attack 14

3. PlunderVolt & VoltPillager Attack 14

4. The Worst Assumption 15

5. Coping Solutions 15

V Economic Model 15

VI Technology Evolution 16

VII References 16

I Overview

A) New Chance for Decentralized Systems and Cloud Computing Services

Decentralized Ledger refers to the fact that transaction accounting is done by multiple nodes distributing in different places. These nodes all can participate in supervising the legality of transactions which they also jointly verify. Blockchain is a typical example of such distributed ledger, distributing across and managed by a peer-to-peer network. Given the fact that blockchain is one of the distributed ledgers, it can run without a central server with its data quality maintained through database replication and trusted computing. However, the structure of the blockchain also distinguishes it from other representations of distributed ledger. The data on blockchain is grouped and organized in blocks that are at the same time connected in chronological order forming a chain which is secured by cryptographic techniques. A distributed data structure as such can bring a strong decentralized consensus for a zero-trust decentralized network.

A key quality for a technology to attain sustainable development and gain popularity is that it should be able to address actual social problems and improve the efficiency in social production. The impact of information technology (IT) on the network and social production efficiency has been profoundly proved and still gets deepening. While storage and computing are just two core elements of the productivity revolution of IT. Thus, blockchain technology not only needs to provide a mechanism for building consensus and reaching the trustless (i.e. Value Decentralization), but also should build a decentralized infrastructure for the two elements, that is, storage and computing (i.e. Storage Decentralization and Computing Decentralization). Among those existing mainstream decentralized consensus mechanisms, blockchain is the one widely adopted in trust building (production network) infrastructure, which however often inevitably entails considerably high consumption of storage and computing power. Moreover, it often gives inadequate support for the decentralization of storage and computing.

Today, most storage and computing scenarios are carried on a centralized cloud computing platform, and storage services as one of the most important components of cloud computing market have become the cornerstone of most cloud services. Centralized cloud storage is designed to put storage resources on decentralized JBoD (Just a Bunch of Disks), allowing users to easily access data any time anywhere through any networked device. However, there are significant problems in this centralized scenario, such as the instability of services, high network bandwidth cost and limited data transmission capacity. Therefore, taking the blockchain trust building as a starting point and through the innovation and optimization of related technologies, the present project aims to achieve a trustworthy, reliable, efficient and accessible decentralized cloud service ecosystem under a decentralized storage scenario.

B) Crust Brief

Crust is a digitally encrypted application layer built on the MPoW (Meaningful Proof of Work) mechanism and GPoS (Guaranteed Proof of Stake). It is also a new generation of blockchain technology that supports decentralized storage and computing.

Crust network is a fair and open mechanism featuring high security and low energy consumption. Under the decentralized storage and TEE scenario, Crust feeds the consensus foundation built by blockchain technology back into decentralized storage through GPoS, enabling everyone to use idle storage devices to participate in the construction of decentralized file systems in an easy

and fair fashion. Besides, it also supports accessing and processing meaningful data in an efficient, secure and low-cost manner.

The flexibility of MPoW determines that Crust ecosystem by design can combine consensus building not only with decentralized storage, but also with decentralized computing. Thus, starting from the decentralized storage, Crust makes possible the seamless transition from its current situation to an entire stack of technologies of "Activating Layer (consensus) + Network Layer + Persistence Layer (storage) + Application Layer (computing)."

II TEE-based Meaningful Proof-of-Work Mechanism: MPoW

A) TEE (Trusted Execution Environment)

Trusted Computing refers to applying trusted computing platform on HSM (hardware security module) support in computing and communication systems to improve system security. With the deepening of trusted computing studies, the public attention has gradually shifted from the traditional hardware chip security model to TEE (Trusted Execution Environment). TEE is a concept proposed by Global Platform. Currently, TEE has a variety of implementation solutions, among which Intel chip-based SGX and ARM open source framework-based TrustZone are the two most widely recognized and applied in TEE technical implementation.

TEE features a secure combination of multiple computer-related technologies. The following 5 technical concepts are the core specifications of TEE:

1. Endorsement key

The endorsement key must be randomly generated and cannot be changed, and the private key must be securely saved. Interfaces, except for several specified ones which can be called, cannot be obtained in any other way. The public key is used to verify and encrypt the sensitive data to be sent.

2. Secure input and output

Input and output together refers to the interaction between the user and the system, involving such items as keyboards, peripherals and network interfaces. Secure Input and Output means that there is a protected path from system user to the accessed site.

3. Memory curtaining

The memory curtaining has expanded the general storage protection technology and provided completely isolated storage areas. Even the operating system itself does not have the full access to the curtained area, so even though there may be intruders controlling the operating system, the Run Time data will still be secure.

4. Sealed storage

Sealed storage protects private information by bundling private information to the user platform configuration information. This means that seal-stored data can only be read in the very same secure environment.

5. Remote attestation

Remote attestation means that the software credentia of the current system will be generated by the endorsement key. Any change on the system will be perceived and then need to be verified by the remote authorized party through the credentia so that the execution logic of the system can be made secure and credible.

These above 5 key technologies are what a complete TEE technology solution is expected to incorporate. The current mainstream TEE technologies are mainly hardware chip-based Intel SGX and ARM open-source framework TrustZone, and Crust currently shows a fairly sound support for both the two solutions and TPM (Trusted Platform Module)-based TEE software implementation. Since SGX is more widely used on the PC and has relatively higher security, the TEE technology described in the following section is more centering around it.

Compared with those complex algorithm-level solutions, TEE is much simpler and more efficient in implementation. In terms of technology development, TEE has a rapidly developing technical ecology and a strong driving force for sustainable development. As with functions are regarded, TEE supports the trusted execution of complex computing logic, which is more in line with Crust's technical vision, that is, further supporting decentralized computing on the basis of decentralized storage, thus forming a complete decentralized cloud service ecosystem.

B) MPoW Mechanism

There is no such centralized institute like a bank in the blockchain system, instead, it is the consensus mechanism that guarantees the consistency and correctness of each transaction on all accounting nodes in information transmitting and value transferring. The consensus mechanism of the blockchain enables all nodes of the whole network to work in large-scale collaboration without relying on any centralized organization. Some current mainstream blockchain consensus mechanisms, such as PoW and PoC, often need to rely on workloads calculated through specific computing or storing procedures that are generally considered meaningless. Crust, however, by combining TEE technologies on the basis of distributed storage and verification scenario, has proposed the original MPoW (Meaningful Proof of Work) mechanism. Compared with those current meaningless space-based solutions, MPoW can be used to quantify a variety of meaningful data storage and computing process in a secure, fair and efficient way.

The MPoW mechanism is mainly responsible for node workload calculation and environment verification. We will explain these two functions and related processes from a storage scenario:

Workload calculation: A node receives distributed data and stores it to the hard disk. When users' data are stored, a regular spot check program is performed in the local TEE, which aims to verify the Merkle Hash and check whether the storage space declared by the node is used to properly save users' file or not.

Environment verification: A checking program is run in the node TEE to remotely attest the logic of other nodes' TEE environment information and trusted execution code version information.

It can be seen that the integrity check for the data, the verification and calculation of the storage, the examination of the node environment and the identity verification of the nodes are all under the sound protection of the TEE.

MPoW features the following advantages:

Transparency: The storage mechanism is both open and transparent.

Fairness: The computing for the workload and reward of storage node both are under TEE protection, and nodes are freed the angst about their workload being unrequited. Also, no extra rewards will be gained by cheating.

Efficiency: Extra redundancy and storage of meaningless data both are kept away from the proof of storage. Both computing resources and storage resources are utilized efficiently.

Evolvability: The TEE supports complete computing and is potentially evolving. This means that Crust blockchain ecosystem can be based on MPoW to perform more powerful functions, ensuring the evolutionary feasibility from storage consensus to computing consensus.

III Crust Network

Crust network is an unlimited horizontally expandable peer-to-peer network, where nodes can freely enter and exit. This chapter will brief Crust network from the perspective of Crust nodes, Crust network construction, as well as Crust technology architecture.

A) Node Functions

Meaningful Work

"Meaningful work" means that nodes can provide efficient storage and computing resources to meet real storage and computing needs. In early days of network construction, Crust was committed to building a decentralized storage network. Therefore, the following part is a description based on a decentralized storage network scenario, where nodes are mainly responsible for storing user data. Node workload rewards mainly come from both the user's storage space lease and the blockchain reward obtained from contribution of storage space.

Work Report

A node needs to support TEE on a hardware basis, provide storage spaces, and run software or programs that comply with the MPoW open-source protocol framework. In order to ensure that user data are completely stored, the node needs to self-perform selective examinations on the Merkle Hash fragments of stored files in each block period and generate a TEE storage declaration report in the TEE. Since the selective examination mechanism is written into TEE, the examination process cannot be interrupted at the operating system level or be randomly tampered with. Therefore, the degree of credibility of workload calculation of each node is statistically equal to the security level of the TEE technology.

The storage workload recorded in a block is derived from the TEE Storage Declaration Report, which is the basic unit of storage records on Crust blockchain. It contains the storage capacity information of a node and a signature from local TEE.

Node Environment Verification

Nodes assume the responsibility of verifying the identity of other new nodes, verifying TEE, and verifying node workload. They need to run a mirroring that supports MPoW and finish the following procedures:

1. Calculating declared storage according to received TEE storage declaration;
2. Verifying the TEE of other nodes in the network;
3. Receiving, verifying and packaging the TEE storage report to the chain;
4. Receiving, verifying and packaging the storage lease contract to the chain;
5. Receiving, verifying and packaging other transaction information to the chain;

Under the protection of TEE, the verification by one node to other nodes is credible. TEE nodes that carry malicious or cheating code will not be allowed to join the network.

Node Enrollment

There will first be several initial nodes in the whole network. The TEE of these nodes contains all the logic required by verification nodes. As the public key certificate of the TEE node needs to be maintained on the chain, the enrollment procedures of a node are reflected as follows:

1. The public key and related information of the existing nodes in the network are pulled;
2. Verifying mutually with a node in the network through TEE remote attestation;
3. The verification result is publicized and packaged onto the chain in validation after it is verified by other nodes;
4. Providing the public key generated by the TEE of the node and writing it to the chain.

Node Service

Nodes can provide a variety of storage and retrieval services. Through the IPFS protocol, for example, a node can retrieve users' meaningful files to the local storage, and can also respond to retrieval requests from the users or other nodes in the network to exchange files or file blocks.

Network Topology

Since the nodes of Crust network embody two different functions: blockchain consensus and meaningful work, Crust network is by nature incorporated a dual network of data storage and block verification. Crust storage layer adapts to a variety of distributed storage protocols (such as IPFS), P2P network architectures (such as DAT) and DHT technology, enabling fast and robust storage and distribution of data blocks. The verification network is responsible for verifying node information and maintaining blockchain data.

When a node is applying for the enrollment to the network, those already enrolled nodes will verify the TEE instance initiated by the newly to-be-enrolled node, and the result will be recorded on the chain. Once the TEE instance is restarted or destroyed, the node will need to be re-verified by the network. The on-chain information mainly consists of a four-tuple:

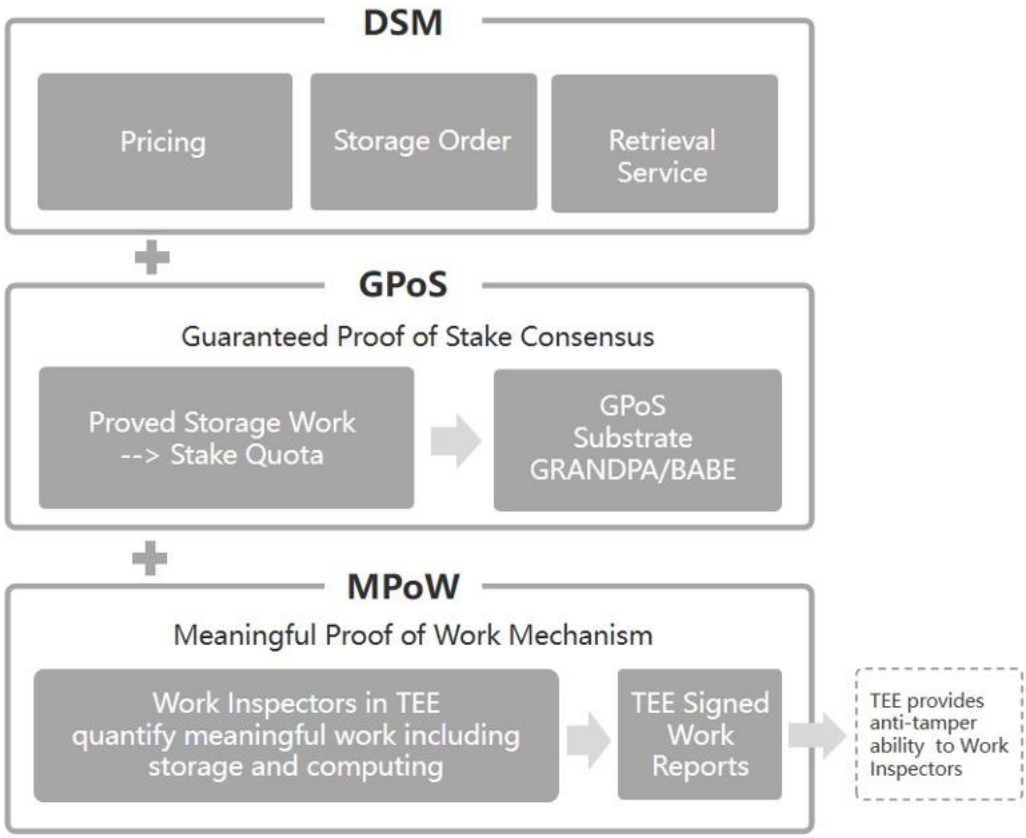
$$\{R, Sig_M(R), Sig_V(R), Sig_M(Sig_V(R))\}$$

R is the report of the verified node, comprising trusted execution environment information of the node, the storage space declared by the node, and the empty disk proof. *Sig* represents the signature computation. V and M are respectively the endorsing node and the newly to-be-enrolled node. This four-tuple ensures that there is a unique endorsing node for each node.

When the storage state of a node changes, such as when users' data storage or storage space changes, it will be required to verify the external storage state change in the TEE and update the storage declaration report while sending the new storage state on the chain.

B) Technical Architecture

Crust is composed of MPoW, GPoS and a distributed cloud storage/computing service layer.



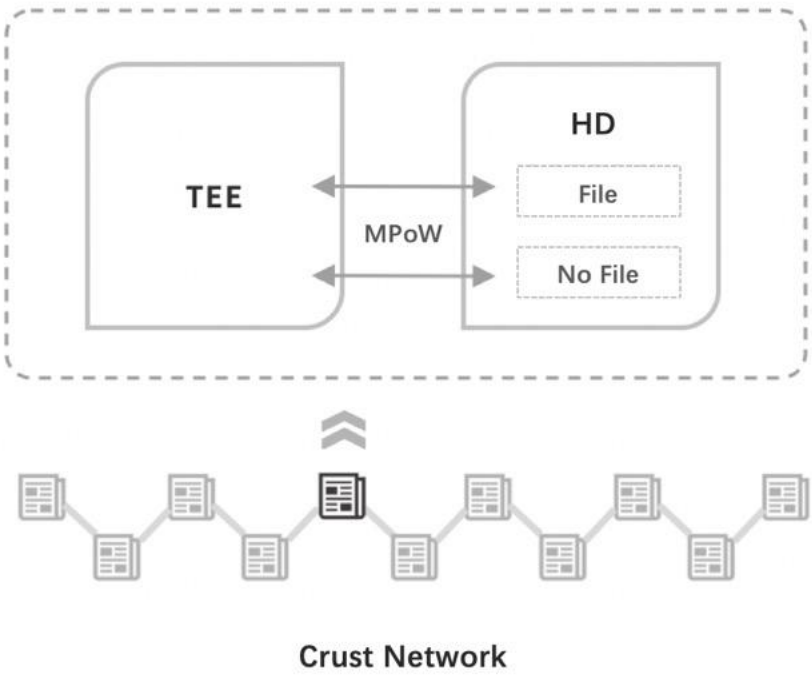
MPoW

The MPoW mechanism is built on the TEE to provide technical assurance for the trusted execution of codes, and the TEE technology is providing the following supports for the MPoW mechanism:

1. Security computing for isolated storage area
Data in the isolated storage area cannot be obtained by any external processes.

2. The binding of TEE to the node identity (public key)
The public key generated by TEE can be uniquely associated to the validity of nodes through remote attestation.

3. The sealing of private and sensitive data
Private and sensitive data can be processed in TEE isolated storage area, but is wholly encrypted during transmission and storage, and no nodes can observe or obtain it.



The MPoW mechanism is composed of two types of proof: environment consensus and workload consensus.

1. Environment Consensus:
When a new node joins the network, a consensus on its TEE needs to be reached based on the MPoW mechanism. Nodes in Crust network will verify the environment of the new-to-network node. The node identity and corresponding TEE public key that pass the verification will be recorded on the chain.

2. Workload Consensus:
a) Every a random cycle, the storage capacity and storage status of a Crust node will be spot-checked by the local TEE of the node. The packaging and verification logic of MPoW is also handled by the local TEE. After receiving user files, Crust storage nodes perform re-encryption algorithm embeded in TEE and save them. It is in this way that only the TEE can restore the files in the external storage, and the node cannot carry out Sybil Attack either. In each cycle, the TEE signs a work report onto the chain after fast local storage verification with other nodes only needing to verify the signature reported by the workload, which greatly simplifies the storage consensus process. Therefore, TEE-based verification has reduced the occupation of network and computing resources, compared with verifications based on complex remote challenge algorithms.

b) The workload of working nodes can also be calculated and verified based on MPoW. Crust has proposed a Proof of Running Tracking (PoRT) algorithm. By combining the TEE with LXC (Linux Container), computing workload by working nodes can be calculated also with the consensus reached.

GPoS

Multiple participants are seen in the entire Crust system, each having varied needs. According to the way of participation, they can be divided into: verifiers, candidates, guarantors, and users (to be specific, users of storage and computing resources).

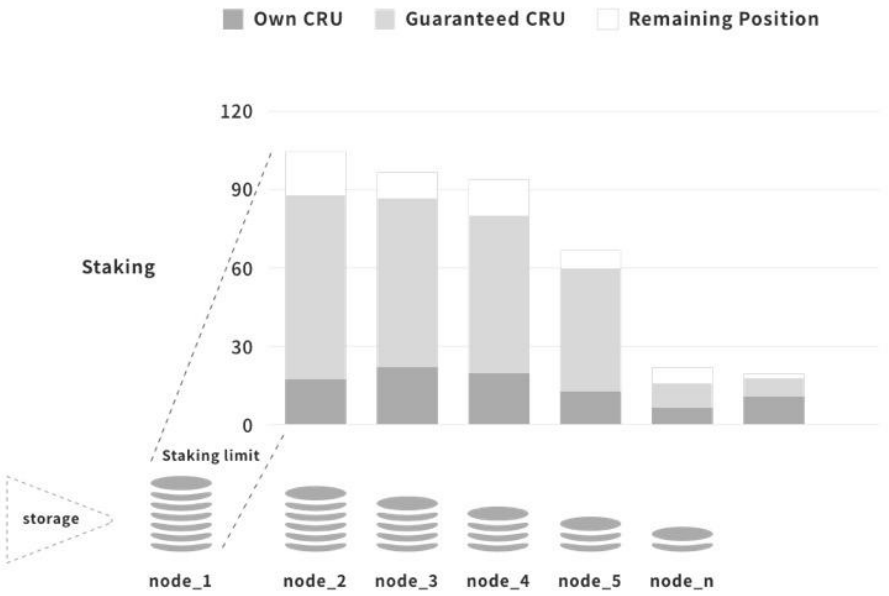
1. Verifiers
Verifiers are nodes that are responsible for packaging and generating blocks in Crust network while also maintaining the entire blockchain network. According to the GPoS (Guaranteed Proof of Stake) consensus of Crust network, verifier nodes need to hold some storage resources as a guarantee, and they can stake CRU tokens (tokens in Crust network, which will be detailed in the next chapter) while staying online. Therefore, verifier nodes are also where storage resources are provided. Besides, verifier nodes participating in the network can obtain the rewards separately given to block package and the reward share of each blockchain cycle, also bearing the risk of assets being confiscated. Verifiers can also sell storage resources to gain income in the storage transaction market.

2. Candidates
Candidates are nodes that participated in the competition of verifiers, but were not in the end qualified for verification in Crust network. Similar to verifier nodes, candidate nodes also need to have storage resources as a guarantee, and can stake a certain number of CRU tokens while staying online. The difference is that candidate nodes do not participate in the block generation and thus cannot obtain the reward separately given to nodes that generate blocks. But candidate nodes can get the reward share of each blockchain cycle, and they can also sell storage resources to gain income in the storage transaction market. It should be noted that candidates and verifiers are not fixed. They may change every cycle, which is mainly determined by the number of tokens staked by nodes at the end of each cycle.

3. Guarantors
Guarantors are accounts that provide guarantee for any one or a few nodes in Crust network. Any account with CRU tokens can become a guarantor, and its CRU can be used as an encumbered asset. Guarantors can also obtain guarantee income by providing guarantees for nodes.

4. Users
Users are consumers who use Crust network resources, mainly those using storage and computing resources. Users can use CRU tokens or other token assets available in Crust network to purchase resource services.

Crust chain has adopted a GPoS (Guaranteed Proof of Stake) consensus mechanism, which is also known as PoS consensus with storage resources as guarantees. Similar to existing PoS projects, nodes need to compete for the position of verifiers by staking CRU tokens, while the difference is that nodes additionally need to provide storage resources in order to obtain corresponding guarantee limit which makes staking CRU itself possible in the first place. Through the MPoW mechanism and node storage capacity monitoring mechanism, it is the case that the more storage resources a node contributes, the higher the limit will correspondingly be.



A node can extend its Steke Limit by providing storage proofs of the following two types of files:

The first category involves meaningful order files from users. Storing these files can improve the usability and servicing capabilities of Crust network (see DSM description for details);

The second category concerns non-meaningful empty disk proof files (see Chapter IV for details).

GPoS performs the final block generation based on the BABE/GRANDPA algorithm of the Substrate framework. And if there is anyone trying to attack Crust network by targeting the consensus, he not only needs to own a large quantity of CRU tokens, but also has to have control over enough storage resources. Such a design is making the attack relatively difficult to occur.

DSM

Crust DSM (Decentralized Storage Market) aims to provide high-quality storage services for applications and platforms based on the Crust network. Storage services mainly involve a storage order mechanism and a retrieval mechanism.

1. Pricing Mechanism
In the Crust network, users sign storage orders regarding the entire network rather than specifically centering on a single node. In this mode, when a user storage order is generated, a corresponding price will be calculated by the network in relation to the current storage supply-demand situation (see *White Paper of Crust Economy* for details).

2. Storage Order Mechanism
DSM provides users with access to order storage, and users can store their files in the Crust network on a long-term basis with some payment.

Crust storage order mechanism is pool-based. With this mechanism, a user generates an order containing information of storage demands and a short description of the files to be stored to

Crust network. One part of the fee paid by the user will go to the reward pool of the entire network where it will be distributed to nodes that provided CRU token staking; the other part will flow to the reward pool, where the user's files were stored, and be paid to those who provided storage proofs for the files.

Nodes in the Crust network can obtain the files through IPFS and save them locally. Then, nodes can declare file storage shortly after the sealing, verification and proof by the local MPoW. All nodes that provided file storage proofs will enter the reward list in order, and those top-ranking ones will obtain rewards from the reward pool.

The efficiency of a node obtaining files and MPoW providing proofs will have an effect on the node's ability of order taking. Responding to this, Crust network adopts a credit mechanism similar to BitSwap to enable highly capable nodes to obtain user order files more efficiently (see Chapter IV for details).

Storage services of Crust network are mainly adapted to such technologies as Inter Planetary File System (IPFS) and Distributed Hash Table (DHT), enabling basic data integrity, content addressing, tamper resistance and deduplication. The difference is that by the strength of MPoW, storage capacity computing and verification can be performed in the local TEE, which greatly increases the efficiency and reliability of workload computing.

Apart from those basic storage features, Crust Network took a step further in user privacy protection. The TEE-based Crust network can support the establishment of encrypted channels and the sealing of data between node Enclaves. Users can decide their private data to be transmitted through encrypted channels and to be stored on a TEE sealed manner. User data encrypted in this way will not be obtained by anyone (even storage nodes themselves) other than the only user.

3. Retrieval Services

There are two types of retrieval demands in the Crust network: those from users and those from nodes.

The former reflects users' needs for data use, which is also a manifestation of the value of storage applications on the Crust network.

Similarly, nodes also have retrieval demands for files. For one thing, the competition for rewards from new orders is based on retrieval; For another, storing meaningful files can extend the Stake Limit of a node.

Crust network nodes draw on the credit game mechanism similar to BitSwap, that is, to provide more retrieval services to those nodes that have ever offered services. Consequently, the result of the game is that nodes which provided retrieval services will be more likely to retrieve data from other nodes (see Chapter IV for details).

It is also in such a game mechanism that an incentive cycle has been formed between the storage and the retrieval of meaningful files. The income of a node is contingent on how efficiently the node can retrieve files, which is further determined by the node's response to previous retrieval requests. For nodes in the Crust network, they will store as much user data as possible and increase their data download speed, a way user experience has been greatly improved.

IV Technical Implementation

A) Crust Remote Attestation

The remote attestation mechanism has well addressed those reliability-related problems of software execution and plays a crucial role for TEE in resisting potential malicious behaviors. In Crust, remote attestation is also the core of decentralized network construction. By embedding the public key which currently runs the TEE during the remote attestation, a node can bind the node identity, execution logic, and platform parameters to the TEE public key on the blockchain. The remote attestation process is initiated by any node in Crust network, which requires the to-be-verified node to prove:

1. its identity;
2. its running logic not been tampered;
3. itself running on a genuine platform with Intel SGX enabled.

The primary procedures of MPoW remote attestation are:

1. The challenger first submits an attestation request to the to-be-verified (challenged) platform, including a random number to prevent replay attacks;

2. The challenged platform then collects the Endorsement Key information, also known as EK, which was written during chip manufacturing (used to identify the unique identity of the trusted platform). Then, EK is used to generate the Application Identity Key (AIK) to avoid the risk of revealing privacy, after which EK is sent to the Privacy Certification Agency (PCA);

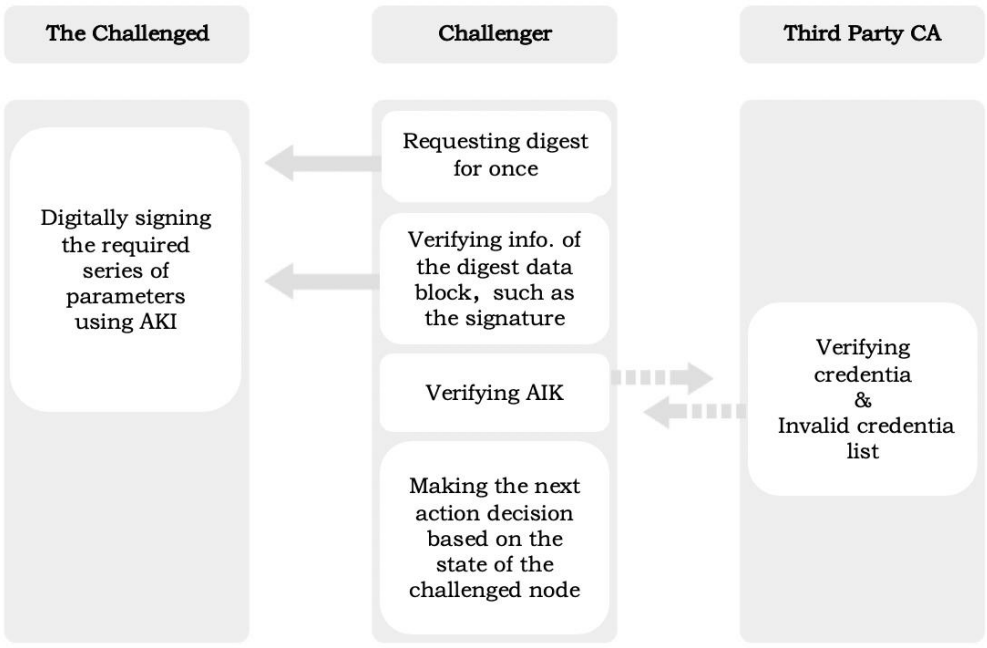
3. The PCA verifies the legality of the chip by verifying the EK and issues a credentia to the AIK;

4. The challenged platform then signs the software metrics using the AIK through the Quote operation, and then sends the signature value, the metric log and the AIK credentia back to the challenger;

5. The challenger first verifies the validity of the AIK credentia, obtains the software metrics by using the AIK public key to decrypt the data, and then ensures that the metric log is indeed securely returned to the challenger through the software metrics. Then the challenger compares each item of the metric logs with the expected value to judge whether the platform should be trusted or not;

6. The challenger writes the AIK public key of the challenged into the blockchain.

Note: 1, 2 and 3 constitute the initialization stage, and 4 and 5 are called the attestation stage.



The AIK private key generated in the above process will be saved by the node in the TEE memory curtaining area and can only be securely accessed by the verified and trusted execute program inside the TEE. The execution results gained by trusted execution program within the node TEE will be signed by the AIK private key and can be verified by the AIK public key recorded on the chain.

B) Distributed Storage

Crust is compatible with some decentralized storage technologies such as P2P basic network architecture of IPFS and DHT technology for storing and distributing data blocks in a fast and robust manner. Further, Crust has also made some expansion and optimization in intelligent redundancy, structured data support, supervision and blacklist mechanism, file encryption and rights management.

C) Proof of Data Storage

A well-developed and reliable data storage proof often needs to include data integrity verification mechanism and data space-time verification mechanism. In MPoW, the data integrity verification is mainly based on the TEE local verification mechanism of MPoW, and the data space-time verification is very much similar to the classic PoSt.

Regarding PoSt and PoRep (Proof-of-Replication), Filecoin has given a series of definitions. But the essential idea is that a valid prover P should persuade a verifier V to believe that P has stored some data D over a period of time. MPoW implements the self-verification of local storage through TEE technology, which effectively reduces the complexity of PoRep, simplifies the existing PoSt procedure and lowers network and computing costs in some way while resisting Sybil Attack, Outsourcing Attack and Generation Attack.

D) Proof of Empty Disk

In order to measure the storage supply of nodes, we define an empty disk proof mechanism so that nodes can effectively track the storage space declared by nodes within the TEE.

A meaningless data block δ is randomly generated in the node TEE, and the available space is filled with δ. The storage proof r composed of δ is traced by TEE and the chain, and TEE periodically checks and verifies the local storage according to r to ensure that all declared storage spaces are truly available.

E) Data Sealing

In order to defend against Generation Attacks and Sybil Attacks, Crust will seal user files within the TEE. The fact that nodes cannot actively generate sealed files from source files, and that TEE's verification of file integrity is based on sealed files has guaranteed that TEE-based data sealing can effectively resist Sybil Attacks and Generation Attacks.

F) Incentives for node retrieval services

The node data exchange in the credit mechanism of BitSwap is based on a game mechanism. The currently widely adopted practice is that each node calculates the credit score and (debt ratio, r):

$$r = \frac{data\_sent}{data\_recv + 1}$$

The data transmission probability P according to data sent and received by other nodes:

$$P(send|r) = 1 - (\frac{1}{1 + \exp(6 - 3r)})$$

The credit game mechanism for nodes in the Crust network aims to:

- make the overall performance and efficiency of node data exchange the best;
- prevent possible attacks and the behavior of downloading with no uploading;
- allow nodes to allocate bandwidth resources in a reasonable way;
- make the game process shorter and more reliable by using the storage order data in Crust network.

By using the storage order data in Crust network, a node can determine whether or not a certain retrieval is for a valid file. Based on this, we can effectively prevent nodes from excessively storing those hot-hit files to cheat in credit scores.

Additionally, the storage order data in Crust network can also help obtain the overall picture of storage, that is, "who" has stored "what." Based on this, the Crust network can calculate the repetition rate of files, and by extension, provide some additional storage incentives to files with a low repetition rate, thus ensuring the reliability of those sleeper files.

A good credit game mechanism can enable capable nodes that have provided retrieval services in the network to obtain storage order files more effectively, which in the long run will bring the nodes with more order rewards, thereby effectively motivating nodes across the entire network to provide quality retrieval services.

G) TEE Update

The Substrate framework has provided a powerful forkless upgrading mechanism. But in addition to those on-chain protocols, Crust protocol stack also includes some protocols inside TEE. Therefore, this protogenic Substrate mechanism cannot be directly applied to Crust.

Based on the characteristics of Substrate and TEE, Crust team has implemented "AB-Upgrade" solution, which can securely update the code in the TEE in a seamless upgrading manner, bringing into reality the off-chain fork-free upgrade.

H) Attacks and Threats

1. SCA of SGX

Intel SGX is a hardware-based TEE implementation. Technically, even if an attacker gains such permissions as OS, hypervisor, BIOS or SMM, it still cannot directly attack Enclave. Therefore, attackers often turn to side channels, such as page tables, Cache and DRAM. The main techniques of SCA (side channel attack) is first to obtain data through the attack surface, derive control flow and data stream information, and then obtain the Enclaved code and data, such as encrypted key, privacy data, and the like.

Under Crust protocol framework, the core sensitive data in each node TEE, that is, the TEE private key, will be posed with major threats by the SCA. One possible approach to combat the SCA is to introduce enhanced cryptographic algorithms at the source level of program, such as enhanced elliptic curves or AES algorithms. By the strength of this enhancement, the data stream and the control flow can be hidden, and the sensitive data in the node TEE can be effectively protected.

2. SGX-ROP Attack

ROP (Return-Oriented Programming) is a new type of code reuse-based attack that allows an attacker to extract gadgets from existing libraries or executable files to build malicious code. By scanning the existing DLL (dynamic link library) and the executable files, the attacker can extract available gadgets, which are all terminated by "ret" instruction. The "ret" instruction is used to connect the execution stream of the gadgets. In SGX-ROP attacks, malicious programs need to be loaded into the TEE to launch damages to the host, and the malware protection software cannot scan out any useful information from the SGX Enclave.

Since Crust is an open-source framework, any code and source published in the community can be reviewed, thus fundamentally eliminating the possibility of malicious code attempting to destroy the host. In addition, the ROP attack is directed to the local system. If the node TEE is secretly embedded with a malicious code, it will still be detected by other nodes and kept wholly out, so the entire network will not be very much affected.

3. PlunderVolt & VoltPillager Attack

PlunderVolt and VoltPillager attacks target the encryption key of SGX from software and hardware respectively. Both of the two attacks by nature attempt to inject controllable hardware failures into Intel's advanced encryption instructions by manipulating the processor's power frequency and voltage, aiming to generate false output and enable attackers to recover the encryption key outside the enclave.

The Work Report mechanism, file sealing mechanism, and MetaData Sealing storage of Crust all rely on such encryption algorithms as ECC and AES. In order to prevent nodes from obtaining private keys through PlunderVolt or VoltPillager attacks to forge work reports, we will use rewritten encryption algorithms in some key steps and avoid using Intel advanced encryption instructions to fend off PlunderVolt or VoltPillager attacks.

4. The Worst Assumption
Crust not only can defend against the currently known SGX security holes, but also gets prepared for potential future threats. Assuming that the worst case occurs (though it does not seem to happen at least at the moment): a malicious node finally breaks the SGX with pyrrhic victory and obtains the private key, which means that the node can now falsify the TEE software and hardware environment proof at will.

Workload forging: this means that file integrity verification information may be forged so that a false storage can be declared to defraud the reward.

False nodes into the network: false verification may cause nodes carrying malicious codes to join the network.

5. Coping Solutions
a) Setting a reasonable single-point storage limit
This means limiting the false computing power by limiting the storage capacity of single nodes, thereby controlling the impact of effective malicious attacks on the network.
b) Dual TEE architecture
This architecture requires each node to run two TEEs from different suppliers. TEE provides work reports by taking turns and verifies each other. Any attempt of forging work report by a TEE will be discovered and reported by another TEE. Therefore, breaching any single-point TEE or leaking the Keystore of a TEE supplier in fact will not pose any threat to Crust network.
c) TEE solution based on RICS-V
Currently, mainstream TEE solutions are challenged because they are source-closed, which often means possible vulnerabilities and backdoors. However, the RICS-V open standard instruction set architecture (ISA) can resolve this problem at its root. With RICS-V-based TEE solutions continuously maturing, Crust Network is expected to support them in the future.

V Economic Model

This chapter focuses on the role and action in Crust. For more specific details about the economic model, please see *White Paper of Crust Economy*.

As outlined in the definition of the blockchain consensus layer (see Chapter III, Section B), the participants of Crust economic ecology include verifiers, candidates, guarantors, and users.

CRU is a functional token circulating in Crust system. Its main functions appear below:

1. staking and maintaining the GPoS consensus of Crust network;
2. guarantying selected nodes;
3. as a guarantee or commission for providing resource service;
4. as a transaction fee for using network;
5. used to purchase resource services;

6. used for electing and voting for on-chain governance mechanism and proposal.

For more details, please see *White Paper of Crust Economy.*

VI Technology Evolution

Crust is dedicated to formulating and continuously improving protocols while remaining open to new technologies and new participants.

In addition to the technical implementations outlined above, there are some other efforts that may help fuel Crust, including but not limited to:

Supporting multiple TEE solutions. Previously, Crust was mainly based on Intel SGX technology. In the future, Crust will access various solutions through the TEE abstraction layer, such as TrustZone of ARM chips, SEV of AMD, Software TEE based on TPM modules and the coming TEE solution based on RICS-V.

Supporting quantification of computing. Supported by the TEE technology, such as TEE-based code obfuscation algorithms, some decentralized executions similar to FaaS tasks can be quantified.

Supporting the improvement of Layer2 services. Apart from providing basic decentralized storage incentives, Crust will also improve support for Layer2, making the cloud services provided by Crust more user-friendly and easier.

Supporting fully-fledged on-chain governance. In order to further promote technological and ecological progress, Crust will create a decentralized on-chain governance in a fair and efficient manner.

Plugging into the Web3 ecosystem. Crust can address all the decentralized storage scenarios in the Web3 ecosystem, and will also get the ecological acceleration brought by the Web3 ecosystem.

VII References

[1] Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[2] Sabt M, Achemlal M, Bouabdallah A. Trusted Execution Environment: What It is, and What It is Not[C]// IEEE Trustcom/bigdatase/ispa. 2015.

[3] Mckeen F, Alexandrovich I, Anati I, et al. [ACM Press the Hardware and Architectural Support for Security and Privacy 2016 - Seoul, Republic of Korea (2016.06.18-2016.06.18)] Proceedings of the Hardware and Architectural Support for Security and Privacy 2016 on - HASP 2016 - Intel Software Guard Extensions (Intel SGX) Support for Dynamic Memory Management Inside an Enclave[J]. 2016:1-9.

[4] Winter J. Trusted computing building blocks for embedded linux-based ARM trustzone platforms[C]//Acm Workshop on Scalable Trusted Computing. 2008.

[5] Bruschi D, Cavallaro L, Lanzi A, et al. Replay attack in TCG specification and solution[C]// Computer Security Applications Conference. IEEE, 2005.

[6] Douceur J R. The Sybil Attack[C]// International Workshop on Peer-to-peer Systems. 2002.

[7] Dias D, Benet J. Distributed Web Applications with IPFS, Tutorial[C]// International Conference on Web Engineering. 2016.

[8] Cai M, Chervenak A, Frank M. A Peer-to-Peer Replica Location Service Based on a Distributed Hash Table[C]// Supercomputing, Acm/ieee Sc Conference. 2004.
[9]"Filecoin: A Decentralized Storage Network," [online] Available https://filecoin.io/filecoin.pdf
[10] Lerman L, Bontempi G, Markowitch O. Side Channel Attack[J]. Cryptographic Attacks, 2013.
[11] Prandini M, Ramilli M. Return-Oriented Programming[J]. IEEE Security & Privacy, 2012, 10(6):84-87.
[12] Wood Gavin. Polkadot: Vision for a heterogeneous multi-chain framework. 2016.