An End-to-End Privacy-preserving Computing Network

Eigen Labs

Version 1.0

Last Update: September 5, 2021

Content

Chapter 1.

Privacy-preserving Computing is Becoming the New Infrastructure of Web3

1.1 Privacy Leaks: The Openness of Blockchain

The advances of Blockchain from Bitcoin to Ethereum, from Ethereum to the emergence of Parallel Chains, follow “Code is the Law” as the first principle one must always obey. The Blockchain data must be fully open, interoperable and programmable.

The openness of blockchain data breeds a large number of data analysis companies, while the openness and interoperability of smart contracts are the embodiment of the blockchain spirit. Nevertheless, the disclosure of blockchain data and logic not only divulges data privacy, but also brings malicious acts such as frontrunning and MEV. Therefore privacy protections, such as chain assets (Token, NFT), centralized financial behavior and decentralized organization information, has become an indispensable and rigid demand of the Web3 world.

1.2 A Hole of Gas: Layer 1 Privacy Protection Requires Mass Gas

The current mainstream privacy protection scheme on Ethereum is mainly based on zero-knowledge proof to hide the amount and address, namely Mixer, from the public survey[1], the cost of single-round ZK verification has reached up towards 650,000 Gas via the latest zk-SNARK protocol. If gas prices are setted at 20 Gwei,

should one verification consume 0.013 Ether, and at the current price of $3500 per Ether, this would mean one single verification would cost as much as $45.50USD. It is clear that there is a huge price attached to the privacy protection of even a simple balance. If it is the privacy protection of a more complex contract data, it will consume even more gas, and is thus unable to be executed on the mainnet. It can be seen that the existing privacy protection on the chain utilizes large amounts of gas, therefore Ethereum's privacy protection scheme lacks feasibility as it has become a gas black hole.

Presently, many scaling solutions[2], including Sharding, Sidechain, and Rollup, have been proposed to handle the heavy gas costs,. Polygon, Arbitrum and other projects have even reached a consensus and been implemented by the Ethereum community.

As the first privacy-preserving computing network on Layer2, Eigen can resolve the conflict between privacy protection and high gas cost, so that every Web3 participant can enjoy data security at a low gas fee.

1.3 Privacy-preserving computing -- the Fundamentals of Web3

In the Web2 era, the services that individuals obtain are based on exposing their private data to centralized service providers. However, these centralized services could face privacy leaks or unavailability at any given time.

Web3 is built on the basis of blockchain and privacy, and aims to change the awkward bartering of private data for centralized services, to separate service and data ownership, and to return data ownership to the users themselves. However, Web3 is still immature and faces the following problems:

- **Limited scalability**: Because transactions on Web3 need to be confirmed by miners, and through the ultimate consensus of other verification nodes, this mechanism causes the computing performance of Web3 to be far inferior to that of Web2, and the scalability to be limited.
- **Difficulty in protecting the privacy of the calculation process:** Service providers need to combine individual data for personalized services, and calculations often occur on the service provider side, thus the privacy protection of individual data faces a greater technical challenge.

- **Flaws in the blockchain privacy computing infrastructure:** In the Web3 world, data-driven decision-making and data-to-decision-making require the help of algorithms, such as data analysis and machine learning. If these types of calculations are to be completed on the blockchain, the current facilities are not yet adequate for such functions.

First, users should have complete sovereignty over their digital identities. Personal data should be managed by individuals as assets.

Second, online services should be decentralized.In the future, data should be returned to the users. Internet companies should revert to service itself, provide users with services and content (such as information, entertainment, and local amenities, etc.) by combining personalized data, and complete the service delivery process without exposing privacy.

Third, encrypted tokens should become the primary circulation certificate to actualize value transfer from person to person.

Web3 aims to create a connected, trustless and privacy-protected network to maximize the value of digital data. In the Web3 era, NFT not only refers to a type of currency, but also the symbol of data value which represents both identity data and behavior data of an individual (a person or an object).

All of these must be based on privacy computing and privacy protection. Eigen aims to create an "end-to-end privacy computing network for Web3," andto build the infrastructure for Web3:

- Based on the Eigen Layer2 expansion solution, not only the blockchain could be improved, but the gas fee could also be simultaneously lowered.
- With the Eigen privacy computing engine, storage security, computing process security and network transmission security could all be realized, as well as data privacy protection.

The advent of the Web3 era is long awaited, as it represents a significant paradigm shift in governance, from large centralized corporations to a more democratic internet. The Eigen team believes that private computing will become the fundamental application of Web3. As it constructs an end-to-end private computing network, and Eigen is also dedicated to making contributions to the realization of this paramount shifts.

1.4 The Brief History of Privacy-Preserving Computing Technology

Compared with other computing sciences, privacy computing is still in the early stages of development, but a variety of mainstream technologies are steadily developing.

Privacy-preserving computing was born of cryptography science, which is built on computational complexity theory. Modern cryptography originated in 1977, and homomorphic encryption (one of the most important encryption technologies in privacy computing) was proposed the following year. In the following decade, many representative theories such as "Secure Multi-party Computing" and "Zero-Knowledge Proof" emerged, laying the foundation for privacy computing.

At the start of the 21st century, hardware-based encryptions entered the public eye and have become widely adopted in commercial fields because of its excellent performance.

In 2006, the predecessor of TEE, the dual system solution, was first proposed by the OMTP working group. The solution pointed out that an isolated and secure operating system should be provided in addition to the multimedia system to process private data. Now, Intel SGX, ARM TrustZone are the most well-known TEE implementations.

In 2017, Google proposed "federated learning"[3] in its papers, and enables mobile phones to collaboratively learn a shared prediction model while keeping all the training data on device. In addition, Professor Yang Qiang of the Hong Kong University of Science and Technology put forward the theory of "migration learning" and also appealed for the combination of "federated learning" with "migration learning" in several papers.

Today, multiple technologies of privacy computing have co-evolved, and have been applied in many fields such as finance and insurance, medical services, and government data. These methodologies co-developed in hardware-software collaboration and computing areas, to meet the emerging business needs.

1.4.1 Current Popular Privacy Computation Technologies

The Eigen team classifies privacy computation into the following two categories based on the differences in information entropy (whether there is information loss in the calculation process):

- **Protection category:** there is NO information loss in the calculation process (examples include ABE, HE, and MPC).
- **Desensitization category**: there is information loss in the calculation process (examples include K-anonymity and Differential Privacy).

Based on communication referring to in the Open System Interconnection (OSI) Reference Mode, Eigen classifies privacy computation into "Network Layer" and "Application Layer,":

- **Network Layer privacy computation:** Based on privacy computation, Primitives such as Secret Sharing and Garble Circuit provide calculations including Boolean operations, arithmetic operations, comparison operations, single instruction multiple data (SIMD), as well as conversions between them.
- **Application Layer privacy computation:** Provide application-oriented privacy computing solutions, with federated learning as a typical representative. For machine learning, changes need to be made at the algorithm level.

There are four main approaches to implement privacy computation at the technical level:

- **Differential Privacy (DP):** It aims to maximize the accuracy of data searches s while minimizing the chance of identifying records when queried from statistical databases.
- **Homomorphic Encryption (HE):** This allows people to perform a specific form of algebraic operation on the ciphertext to obtain the encrypted result. After decryption, it performs the same operation on the plaintext with the

same result. In other words, this technology permits people to perform computations, such as retrieval and comparison, in the encrypted data to get the correct results. There is no need to decrypt the data during the entire process.

- **Secure Multi-Party Computing (MPC):** Secure Multi-Party Computing is aimed at solving the problem of "how to safely compute a function via their inputs without a trusted third party". MPC is the cryptographic basis for the implementation of many applications such as electronic voting, threshold signatures, and online auctions.
- **Trusted Execution Environment (TEE):** TEE is a secure area in the CPU,. which can ensure that the confidentiality and integrity of the programs and data in it are protected. As an isolated execution environment, TEE's security features can safeguard the integrity of the applications executed in it, and the confidentiality of its assets.

Chapter 2.

Eigen Network: The First End-to-End Privacy-preserving Computing Network

"Eigen" means "own" and "individual" in German. The name "Eigen" implies that the user's data sovereignty returns to the user himself/herself, while ensuring that the user can obtain the same services as before. Eigen Network aims to build an end-to-end privacy computing network, from transaction addresses and balances, to the input and output and processing process of smart contracts, to the storage and synchronization of ledgers, to realize the privacy protection of the whole data link and the whole life cycle process.

2.1 End-to-end Privacy Protection

Privacy-preserving computing is the theory regarding the full-life cycle protection of information. It is a computable model and axiomatic system for privacy measurement, privacy leakage cost, privacy protection and complexity analysis when the ownership, management, and user rights of privacy information are separated. Individual privacy protection involves the whole data life cycle such as data generation, processing, and destruction. Data is generated by individuals, and the processing process is controlled by those individuals. Ultimately, it will also act on the individuals, and be destroyed by them, thus realizing end-to-end privacy protection.

Eigen's end-to-end private computing network requires collaboration among computing, network, and storage. TEE realizes the computing process security.

Homomorphism and smart contracts can ensure the integrity, confidentiality, and audit-free of the execution process. The network security relies on TLS encrypted communication and remote authentication to prevent man-in-the-middle attacks. The security of storage relies on encrypted ledger to prevent data privacy disclosure while ensuring high availability of data storage.

Eigen's vision is to become a privacy computing infrastructure in the Web3 era, accounting for all individual privacy protection from application scenarios to technology stacks.

2.2 Eigen's Strengths and Capabilities

Eigen Network is a general Layer 2 privacy computing network for mainstream public chains, with the following advantages:

- Eigen Network can significantly reduce gas fees for on-chain transactions, providing low-cost for complex computing scenarios such as DeFi and NFT.
- Eigen has native privacy computing capabilities, and can provide end-to-end privacy protection.

Additionally, Eigen also has the following features:

- Natural resistance to MEV: The EigenCC-based fair sequencer is adopted to eliminate the possibility of MeV from the source.
- Parallel processing: EGVM supports parallel smart contract execution to improve the throughput of smart contract processing.
- Full-paradigm privacy computing engine support: Suitable for mainstream computing scenarios via confidential computing and federated learning.
- Co-evolution of Hardware-Software technology: Based on the combination of hardware and software technologies, Eigen can balance performance, security, and reduce deployment and migration costs.

Chapter 3.

Technical Architecture

This chapter introduces Eigen from the angle of technology stack and core modules. It will provide insight into how Eigen implements privacy-protecting smart contracts, and separating joint-computing with private input among multiple parties.

3.1 Overall Architecture

Eigen's tech stack realizes the end-to-end privacy-preserving network using a 4-layer structure. This includes privacy-preserving computing infrastructure, computing engine layer, rollup protocol layer, and the Eigen box from the bottom-up. Based on those four layers, Eigen offers the users high-scalability and verifiable privacy enhancing toolkits and services, and provides strong support to develop the privacy computing ecosystem and infrastructure for Web3.

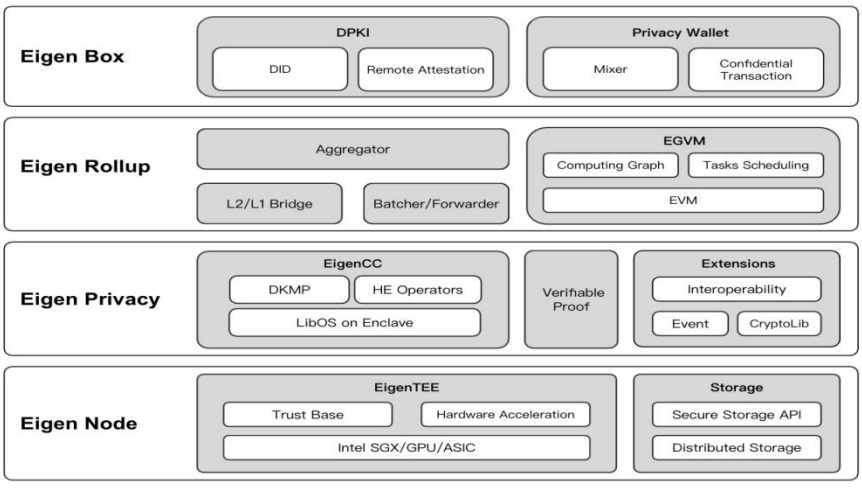

Figure 1 Eigen's Architecture Overview

As shown in figure 1 above, Eigen offers a modular design that allows new components to be incorporated and updated as they become available, which provides the end-users a full-featured privacy computing infrastructure.

- **Eigen Node:** Provider of computing power, which is a miner of Eigen network. Users can purchase specific servers to participate in mining, and gain rewards for contributing computing power.
    - **EigenTEE:** Core Data Processing Unit (DPU) could be built using both off-the-shelf TEE hardware, such as Intel SGX/FPGA, as well as on the SoC with FPGA,which provide trust base, cryptography, and AI with hardware acceleration. This also allows different configurations for different computing scenarios;
    - **Storage Support:** Eigen integrates multiple Distributed Storage drivers and SDK and opens it to end users for their data persistence under security demand.
- **Eigen Privacy:** Provider of privacy-preserving computing engines and toolkits for developers. These capabilities will be open in the form of development frameworks which enables algorithm providers and service providers to implement cross-domain privacy data processing capabilities. Among them are:
    - **EigenCC:** Eigen Confidential Computing, a privacy computing pool built on EigenTEE. It supports distributed task parallel operation, joint data analysis, joint modeling and reasoning. EigenCC performs enclave scheduling through the Enclave framework, and supports homomorphic evaluation operators and secret key hosting within the Enclave；
    - **Verifiable Proof:** Blockchain security is guaranteed by redundant computation and storage, which is a "repetitive" game, resulting in a huge waste of resources. Verifiable Proof provides another more efficient approach to achieve the same security. Verifiable Proof is generated by the computing power providers, and verified by the computing power consumers without re-executing, which uses less resources；
    - **Extensions:** Provide interoperability, event mechanisms, cryptographic libraries, and other extension components.

- **Eigen Rollup:** Eigen Rollup protocol separates the consensus from smart contract execution, and utilizes the Optimistic Rollup solution. This resolves the dispute with an efficient one-step proof, and imports non-interactive zero knowledge to protect the privacy of the prover. The core modules includes:
    - **Rollup Bridge:** A data transfer channel between L2 and L1. This allows for asset interchange between L2 and L1;
    - **Batcher/Forwarder:** A module that packs the primary transactions, and sends it to the InBox on L1;
    - **Aggregator:** Aggregator from L2 then fetches a batch of transactions from the L1 InBox and transports them into EGVM, and waits for the execution results, then returns the results into RollUp Blockchain.

      Specifically, Eigen Rollup works in one of three modes:

    - **Submit mode:** Accept the transaction from users. Pack privacy transactions as a batch, and transfer them to the InBox via batcher. Users also can submit their transactions into the InBox directly;
    - **Primary Validator mode:** Obtain transactions in bulk from the InBox. Perform the cryptographic computation of the batch privacy transactions using EigenTEE. Pack the result and proof of computation into a block, then submit the block onto the Rollup Blockchain (at least including Eigen node, Eigen privacy, Rollup Bridge and Aggregator), to form a block to be confirmed;
    - **Secondary Validator mode:** Validate the proposed Rollup block. If any malfunctioning block is found, it will locate the earliest transaction that was executed incorrectly, and submit a challenge for a reward. This process involves Eigen Node, Eigen Privacy, and Rollup Bridge.
- **EigenBox:** For end-users to access Eigen Network, mainly providing Distributed Public Key Infrastructure (DPKI) and Privacy Wallet functions.

3.2 Eigen Node

As a basic computing unit, Eigen Node provides the computing power for privacy-preserving computing. It is first remotely verified, and then is admitted to join the

Eigen Network, as part of the computing pool. The basic component of Eigen Node is EigenTEE. This chapter introduces what EigenTEE is and how the Remote Attestation works.

3.2.1 EigenTEE

EigenTEE consists of Trust Base, dynamic isolation driver, and secure measurement protocol. It creates a dynamic process group, namely an enclave, as the minimum schedulable unit to run smart contracts.

In the early stage, Eigen was designed based on the existing mature TEE scheme. Intel SGX[4], to implement EigenTEE. Intel SGX is widely used as it offers a set of instructions that increases the security of application code and data, giving them more protection from disclosure or modification via remote Attestation. Intel SGX also enables developers to migrate their applications into the enclave by the Rust SGX SDK based framework, or LibOS, with little modification.

A frequently mentioned issue of Intel SGX, even with all hardware-based TEE, is the Microarchitecture Side-Channel Attack[5]. Side-Channel Attack consists of three main kinds: Cache-based Timing Attacks, Controlled-channel Attacks, and Transient Execution Attacks. The former two attacks can obtain sensitive privacy data, albeit with noise, but the third can get the exact sensitive privacy data from TEE. EigenCC takes effort to use TEE to maintain execution environment integrity instead of using key management. As well, performing computations by homomorphic encryption, thus fundamentally reducing the possibility of side-channel attacks. Moreover, with the help of randomization technology, EigenCC framework adds random useless operation in execution logic and erases the cache after privacy data deconstruction to mitigate the side-channel attack.

In the near future, Eigen will release its self-developed SoC System On a Chip) product with TEE inside,and secrity-enhanced Data Process Unit, integrated with hardware acceleration for AI and cryptography.

3.2.2 Node Attestation Management

All nodes need to register and authenticate when they join Eigen Network to be considered as trusted transaction initiators and distributed task-computing collaborators. This enhances the security of transaction executions.

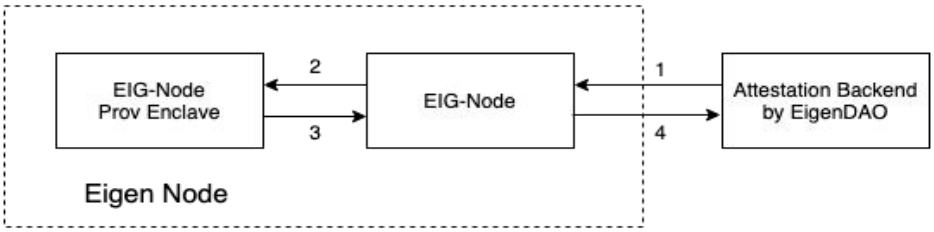

Figure 2 Registration Process

As shown in figure 2, registration is mainly divided into the following four steps:

(1) The new node first obtains the provisioning key corresponding to the currently registered miner from the Attestation Backend, which is owned by EigenDAO. The Provisioning Key is a ring signature address that prevents the node from being traced.

(2) The miner uses this key to start the Attestation Trusted Execution Environment (Provisioning Enclave) through the EIG-Node to generate a random public-private key pair <SK, PK>, and then use the key to sign the PK, and re-sign the signature with the device identity key AIK, and (3) return both signature information and public key information to the EIG-Node. Finally, the node registers the certificate in the Attestation backend.

(3) EIG-Node Prov Enclave returns the two signature information and public key information to EIG-Node；

(4) The EIG-Node registers it with the authentication backend. The authentication backend uses the Provisioning Key for signature verification, and uses the AIK public key for secondary signature verification.

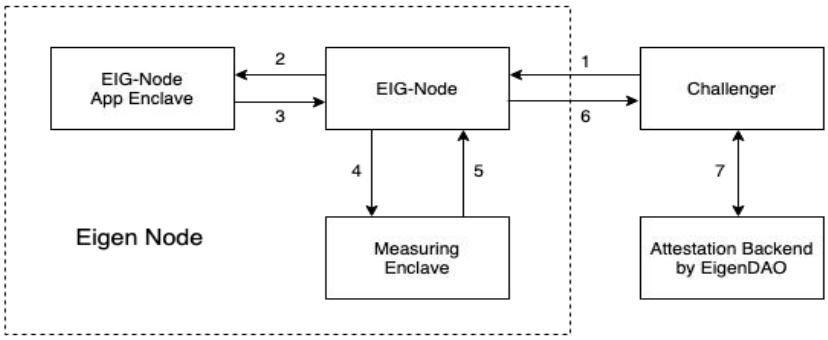

Figure 3 Remote Attestation

As shown as figure 3, in the remote attestation phase, the challenger (client) submits an authentication request to EIG-node. The EIG-node launches an EIG-node app enclave (2,3) to generate a Quote, which includes the hash user data hash, and platform information.

It then calls the measuring enclave (4,5) to sign the quote by AIK. From there (6,7) Challenger uploads the Quote to the Attestation Backend controlled by EigenDao for legitimacy verification. In EigenCC, we complete RA during the TLS establishment process, and the upper layer TA is not aware of it.

After the node is registered through EigenTEE, it becomes a valid miner node in Eigen Network.

3.3 Eigen Privacy Protocol

Data has become an asset. This is a key feature of Web3, which is not present in Web2. However, the value of data cannot be realized until the data is aggregated, circulated, and mined as a set. Privacy-preserving computing has become an absolute prerequisite for the migration from Web2 to Web3. Hence, under the data privacy protection regulations, privacy-preserving technology is urgently in need.

Privacy computing in Eigen mainly contains two important parts: EigenCC, allowing each Web3 participant to build their own privacy DApp, and use them to help build the privacy computing infrastructure for the Web3 era.

3.3.1 Privacy-Preserving Computing Engine

EigenCC provides the trusted execution environment, supporting heavy computing, such as privacy-preserving smart contracts and AI model training and interference. EigenCC deeply utilizes the LibOS to implement parallel execution, strong isolation, and easy migration for new EVM instructions.

EigenCC consists of multiple enclaves. Each enclave consists of a LibOS. LibOS adopts a Completely Fair Scheduler algorithm to schedule EGVM tasks. LibOS encrypts all the data before being written to the encrypted storage through the encrypted IO channel.

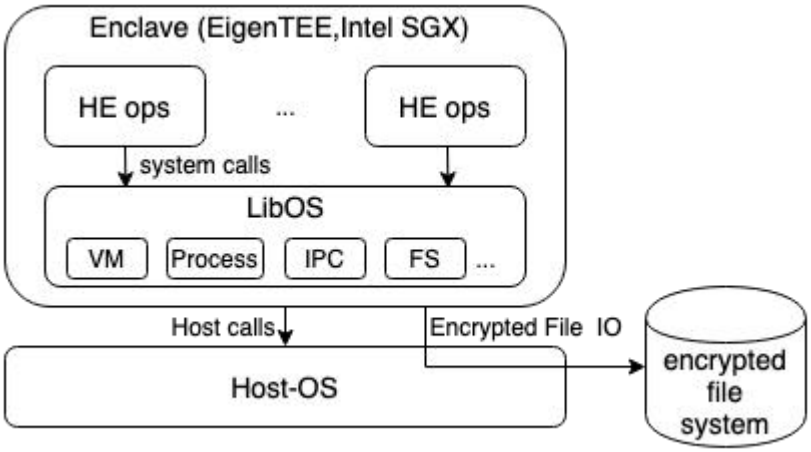

Figure 4 EigenCC Overall Architecture

LibOS implements a lightweight, but fully functional operating system. LibOS has two critical features. Firstly, LibOS adopts CFS to schedule multiple EGVM, which enables Eigen Network to handle millions of smart contract executions per second. Secondly, LibOS supports almost all features of the operating system, which make it easy to backport or migrate all the EVM instructions to EGVM.

LibOS also provides sealing and unsealing toolkits seamlessly on the filesystem layer. EGVM could leverage this to encrypt the ledger or any data in the smart contract to realize the privacy protection of ledger information.

In order to facilitate developers to develop privacy-preserving applications quickly, EigenCC provides developer toolkits and rich system libraries, including TLS, HTTP, cryptography, etc. Based on the developer suite, EigenCC can realize the functions of distributed secret key hosting, encryption operator etc.

It is worth emphasizing that EigenCC provides homomorphic operators to realize privacy computing and fundamentally solves the possible privacy disclosure risk such as side-channel attack.

3.3.2 Decentralized Scalable Key Management Protocol

EigenTEE enables the transaction to be executed in a confidential environment, but it is not secure enough for long-term keystore. For the user's crypto assets, such as private Defi activities and secret NFTs, a safer cryptography system is required. For instance, in Eigen's Private ERC20, Alice transfers some encrypted tokens to Bob,

which Bob can accept safely. Some off-the-shelf TEE based blockchain [6][7] adopts a somewhat centralized key management service to deal with encryption. Such methods can lead to the key being un-upgradable, which is not promising as a public blockchain.

Eigen adopts a completely decentralized key management mode, allowing each user to control the private key of their own encrypted assets and encrypt locally. The algorithm flow is as follows:

Algorithm 4: GEKP Generation

// Decentralized and upgradable Key Management Protocol
initialize K, N **for** *nodes i= 1, .., N* **do**
    $SK_i, PK_i = key_pair_generate(random)$;
    let $y_i, P_{SK_i} = VRF()$;
    send(y_i, P_{SK_i}, nodes[i != j] for j in nodes)
end
;
for *msgs i=1,.., N-1* **do**
    receive y_i, P_{SK_i};
end
$min_K = min_K(y_i, \forall i \in [1..n-1])$
let seed = $muilt\_key\_exchange(min_K)$;
let GEKP = $key\_pair\_generate(seed)$;
$key\_distributes(GEKP)$

Specifically, for each epoch, EigenTEE chooses multiple Eigen Nodes by VRF[8], and generates a seed by Multiple Key Exchange[9]. Then finally generates a Global Ephemeral Key Pair, GEKP, which is a key pair of public-key cryptography. After the key pair is produced, EigenTEE runs a key distribution protocol to deliver the key to each EigenTEE via secure transport. The protocol is shown as below:

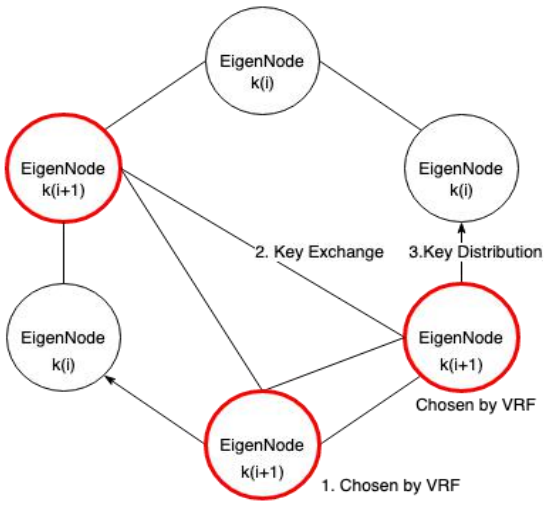

Figure 5 Key Generation Process for EigenTEE

Users' encrypted assets need to be operated inside TEE, the P2PKH scheme is used[10]. The user generates a symmetric key locally, encrypts his own assets, and then transfers the encrypted assets within TEE by encrypting the symmetric key through the GEKP public key. At the same time, for the receiver, the encrypted assets are accepted through the receiver's public key, and then decrypted and used by the user through the local private key.

3.4 Eigen Rollup

Eigen Rollup is a hybrid Rollup protocol, powered by EVM compatibility, Optimistic Execution technology, and Verifiable Proof.

- **Optimistic Execution:** inspired by the CPU's Branch Prediction, the transaction in Eigen Network can be pre-executed by primary validator, and is verified by secondary validators. If a dispute is proposed by some validators, then a Dispute Assertion will be executed.
- **Verifiable Proof:** As mentioned in the previous chapter, TEE and ZKP are applied to generate proof of correction and integrity of transaction execution.

3.4.1 Rollup Transaction Process

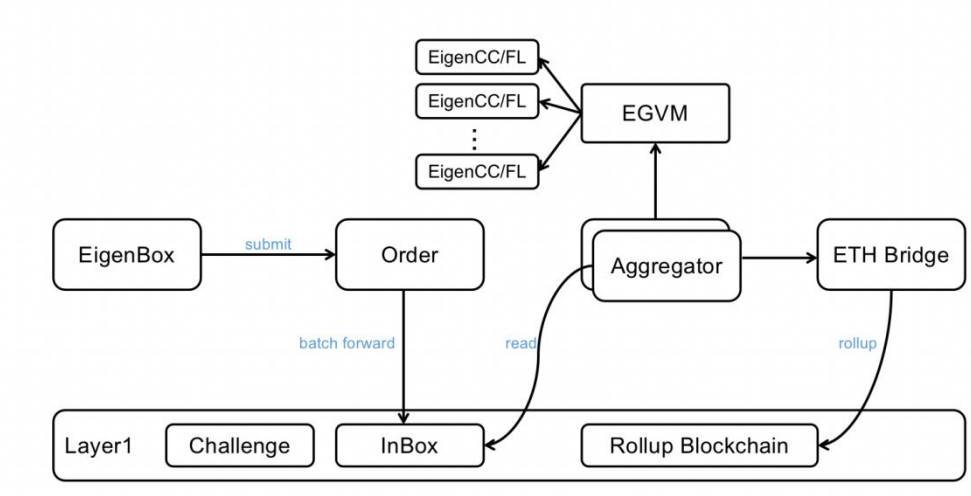

Figure 6 Rollup Transaction Process

The rollup protocol is shown in figure 6, the details are as follows:

1. **Contract deployment:** Rollup contracts are deployed at the initialization stage, which consists of InBox contract, Rollup Blockchain contract, and challenge contract.

2. **Validator Registration:** Validators need to be registered to be a Rollup Blockchain miner, which includes primary validator and secondary validators. Primary validators execute the transaction and mine the block, then send the blockchain as a proposal to Rollup blockchain. The secondary validators verify the block proposal, and challenge it if the proposal is invalid.

3. **Transaction Submission:** The Client (e.g. EigenBox) submits the transaction to the Batcher. The Batcher collects a bunch of transactions as a batch, and uploads the transaction to the InBox, where the transactions will be queued in chronological order.

4. **Transaction Execution:** The Primary Validator fetches a batch of transactions, and posts the batch to EGVM. EGVM chooses the proper computing engine (EigenCC) to execute.

5. **Proposal Block Submission:** The Primary Validator submits the proposal blockchain to the Rollup blockchain via EthBridge, meanwhile stakes Eigen Token on the Rollup contract, endorsed by updating status and verifiable proof.

6. **Block confirmation:** The Secondary Validator repeats the transaction execution, and checks the validity of the block. If any fraud is found, the Rollup blockchain submits a dispute, and once a dispute is submitted, it enters the dispute adjudication stage.

3.4.2 Verifiable Proof

In computational complexity theory, the goal of interactively proving [11] is to determine whether x is in L, with a given input X. An interactive proof system consists of a Verifier and a Prover, both of which can be considered Turing machines. It is calculated by giving the input x, exchanging information between the verifier and the certifier, and ultimately, the verifier determines whether the given input is in language L based on the information given by the certifier.

The power of a certifier in an interactive proof is infinite relative to that of a certifier, but it is untrustworthy. As shown in Figure 7:

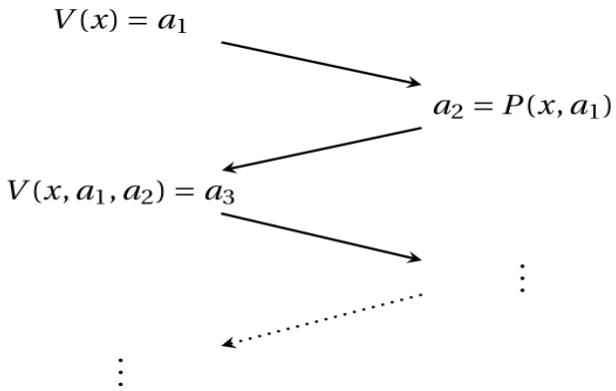

Figure 7 Multi-round Interactive Proof

Verifier V intends to verify that Prover P does calculate according to the protocol implemented. Verifier V first initiates a challenge. He calculates based on his input X and submits the result a_1 to Prover. Based on the input information, Prover finds the corresponding Turing State S and starts the local calculation. Then, the calculation generates a_2 and gives it to Prover for verification. If the validation passes, the Prover will continue to generate a new Turing State S', requiring the Prover to give the next State S'', which will be validated multiple times to ensure that the Prover has been honestly executed.

The interactive proof is the foundation of EigenRollup's separating of consensus and computing from Layer 1 and hosting computing on Layer 2. From previous work, Arbitrum [12]、Truebit[13] also adopt the interactive proof protocol. Compared to the sidechain scheme, TrueBit no longer re-execute the transactions on all participating nodes when disputes arise but instead randomly selects a Solver to perform the execution, then calculates the submitted disputes and broadcasts the verifying result to the verifier. Arbitrum uses One Step Proof in Challenge Smart Contracts to assert the disputes. The validators stake on the proposals which have the same input and output as theirs. After the staking period, the Challenge Smart Contract begins to execute the One Step Proof to determine the valid Rollup chain branch, and incentive the staker on the right branch by transferring the invalid branch's staking token to them. This protocol further reduces the cost, but the challenge smart contract still needs to re-execute almost all the updates instructions and leads to runtime out-of-gas sometimes.

EigenRollup uses TEE to execute key privacy instructions based on Arbitrum's interactive proof scheme, mainly considering:

- **Low Gas consumption of interactive proof:** Privacy operation instructions may involve a large amount of encryption and decryption as well as complex operations. If used in the form of playback instructions or a zero-

knowledge proof scheme [14], it is bound to cause challenge contracts to consume a large amount of Gas, which will result in the challenge process not being completed. Using the TEE scheme, TEE automatically generates verifiable statements of calculation correctness, reduces the Gas consumption of challenge contracts, and reduces the challenge period.

- **Privacy protection of interactive proof:** the verifiable proof generated in TEE can be the signature by TEE's private key and the instruction's execution context, which does not expose any private information of the prover.

3.4.2.1 Verifiable Proof

Verifiable Proof is the integrity proof of the smart contract's execution, forming the basis of verifiable computing. Verifiable Proof plays a key role in 3 scenarios: 1) the concealment of transaction addresses; 2) the verifiable computing for Eigen Rollup, when the Prover provides the proof without exposing specific privacy information; 3) when used in model inference, it is to prove the correctness of the computation of the inference process.

From the formal definition [15] of Verifiable Proof, Verifiable proof means that Verifier transfers calculated F and its input x to Prover, calculates y=F(x), and gives a declaration to prove the correctness of its calculation process.Firstly, F has some universality (Turing is complete and supports high-level language to write), and secondly, Prover has mass computing power or has some efficient algorithm.

If TEE is used to carry out this computing and proof, TEE itself guarantees the integrity of the execution process, so it is easy to produce a signature. That which is dynamically generated from TEE enclave, and cannot be changed to obtain from untrusted sources. In order to make sure the call stack can be protected by a single hash, Eigen adopts a cascade signature:

$$H' = Hash(Hash(input)||H)$$

$$VP = Sig(H^{'}, SK)$$

Where H is the hash value output by the previous operator, and VP is the signature generated by the TEE private key and H.

3.4.2.2 Dispute Assertion

When entering the dispute assertion phase, all the validators can use the Challenge Contract to find the earliest fraud checkpoint, and submit their local block to the Challenge Contract. The Challenge Contract accepts all the checkpoint information, and re-executes each checkpoint on the transaction from one to the next checkpoint, and checks if each checkpoint matches the one in the proposed blockchain. It does not matter if fraud or invalidity is found, all the staked tokens in the fork are transferred to the stakers on the main chain. After the dispute is resolved, the Rollup blockchain re-enters the transaction confirming phase again.

3.4.3 Fair Sequencer Based on EigenCC

Currently the mainstream Layer 2 projects, such as Optimism, and Arbitrum are using centralized sequencers. The sequencer plays two important roles:

- **Low the transaction submission fee to Inbox:** After the user submits the transaction to the sequencer, the sequencer forms a sorted transaction batch and submits the batch to Inbox at once. Compared to the gas fee by submitting transactions to Inbox one by one by the client directly, the Gas fee is much lower.
- **Reduce the malicious MEV:** The sequencer does not actually execute transactions, and therefore has no incentive to re-sort transactions, unless the sequencer conspires with the validator.

At the same time, existing sequencers are too centralized and run the risk of committing malicious acts in conjunction with validation nodes.For example, Arbitrum previously used a project-side standalone operation sequencer scheme to avoid such problems.

Eigen Rollup uses a fair algorithm to implement the sequencer. Specifically, Eigen places the sequencer inside EigenCC, and the transactions firstly are submitted to the sequencer in EigenCC and then are forwarded to Inbox, where the execution environment is remotely attested. EigenCC guarantees that the sequencer cannot reorder the transaction or cheat with the validator, and thus, it fundamentally prevents the risk of MEV.

Eigen Rollup's fair sequencer achieves low handling charges while preventing attacks such as MEV.

3.4.4 EGVM: Distributed Task Execution Engine

EGVM is the Eigen privacy-preserving smart contract virtual machine, supporting distributed and parallel task execution. EGVM distributes the task to EigenCC Cluster, and generates verifiable proof for the task.

To achieve the parallel execution, EGVM adopts the Actor Programming Model. It splits the Task into subgraphs, and sends the subgraphs into the MailBox. Then it executes the task in EigenCC/EigenFC cluster.

The whole protocol executing as figure 8:

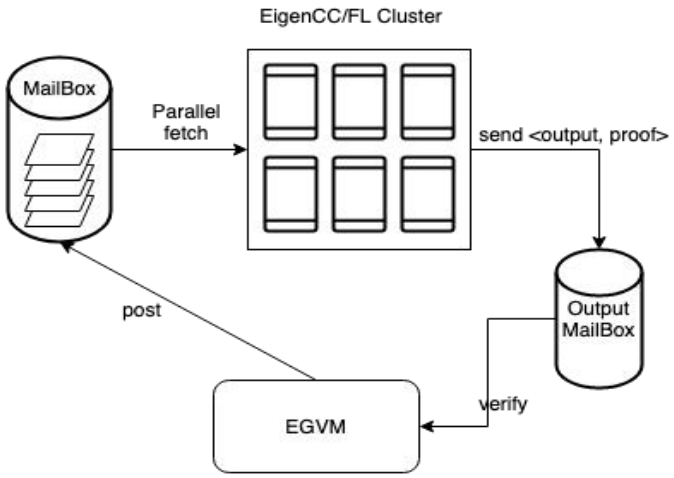

Figure 8 EGVM Computing Process

Algorithm Description:

Algorithm 3: Asynchronous Transaction Exectution Algorithm

Input: mailBox: Channel, outputMailBox: Channel, handle: Func → Future, stop: Func
Output: bool

```
1  let txResultBuffer: Map = Empty;
2  while not stop() do
3  |  tx = mailBox.pop() ;
4  |  assert tx is instance of Tranaction;
5  |  if not txResultBuffer[tx.requireTx].done then
6  |  |  mailBox.pushBack(tx);
7  |  |  continue;
8  |  end
9  |  txResultBuffer += handle(tx);
10 end
```

Algorithm description:

- Line 1, initialize the task node in the DAG, Map: txResultBuffer;

- Line 2-3, EigenCC/FL polls the MailBox for new tasks, or updates the task status. If a EGVM finds a new task, the current contract enters the RUNNING status;
- Line 4, check if the transaction satisfies the structure of the Transaction, and exits directly if not.
- Line 5-8, if the predecessor of current transaction tx is not finished, re-posts the tx into the MailBox, then go to Line 2;
- Line 9, if the predecessor tx is executed, then schedule current tx to RUNNING status, and submit the output <result, proof> to Output MailBox.

3.4.4.1 Computing Graph

Eigen uses the Computing Graph to represent a distributed computing task. The Computing Graph is defined as a DAG (Directed Acyclic Graph) in mathematics. The nodes of DAG are operators, such as addition or multiplication, and the edge of the DAG is the data flow from one node to the other one.

For example, the expression $g = (x + y) \cdot z$, is describe using a Computing Graph as below:

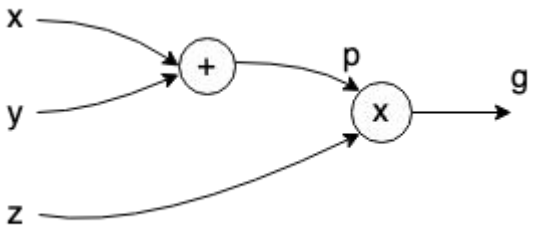

Figure 9 DAG Sketch

DAG meets the strict partial order $\leq$, and if there is an edge from node u to node v, then it will be represented as u$\leq$ v. Depending on this, DAG can strictly divided subgraphs, and one subgraph maps to one computing node:

$$f : DAG(V, E) \rightarrow DAG(V', E')$$
$$s.t. \forall v \in V', v \in V$$
$$\forall e \in E' : \{U' \rightarrow V'\}, |e| = \sum \{e : u \rightarrow v\},\ where\ u \in U', v \in V'$$

The advantages of using DAG to express distributed task are:

- **Parallel computing:** Different subgraphs can be allocated to different devices for parallel computing. The data flow between subgraphs is realized through RPC simulation, so as to minimize idle time and improve parallel efficiency;
- **The calculation process can be quickly converted into distributed verifiable statements:** DAG can be easily converted into arithmetic circuits, which can be naturally converted into verifiable statements in combination with dzkp [16] protocol.

Split a task into subtasks by Kernighan–Lin[17].

Chapter 4.

Practical Ecosystem

The two main applications of Eigen Network are Confidential Smart Contracts and Distributed Privacy Computing. Confidential Smart Contracts can protect the privacy of both the data and logic inside smart contracts. Distributed Privacy Computing networks are able to provide infinite privacy computing power, making possible the migration of the conventional distributed computing task into Web3.

4.1 Confidential Smart Contract

The development of blockchain has brought infinite possibilities to the worldAll fields including finance, gaming, medical and health, credit investigation, Internet of Things and other industries are full of new vitality because of blockchain. Blockchain empowerment lies in its credibility and verifiability. In its earlier stages, it had attracted a wide range of enthusiasts due to its openness and verifiability. However, due to the logic of smart contracts and data, the behavior and privacy of participants were completely exposed. In Web3, as personal information (interests, hobbies, consumption habits, etc.), financial activities (swap transactions, pledges, provision of liquidity, etc.), physical assets (NFT artwork, game skins) and other information are on the chain, the information, as an individual's label and attribute, has strong privacy. If the blockchain itself discloses its transaction information, users would face the threats of privacy disclosure.

Eigen can carry out a privacy protection for address and contract state:

- **Address Hiding:** EigenBox provides a decentralized currency mixing technology based on the combination of Merkle and zero-knowledge proof technology to help users hide addresses.
- **Confidential Smart Contract State:** To hide the key data of smart contracts in the blockchain, EigenBox provides contract privacy data protection based

on EigenCC. Users can encrypt the information locally via EigenBox to store the encrypted information to the blockchain. At the same time, encryption information can be calculated on EigenCC, so as to realize the confidentiality and correctness of smart contract calculation and storage on the blockchain network.

However, blockchain computing is very expensive and inefficient. If one intends to perform complex business logic on the blockchain, one would need to pay very high gas fees, and compute-intensive tasks cannot even be supported by the current smart contract virtual machine. Eigen extends the capability of Layer 1 by shifting the computing to Layer2. On EigenCC, the transactions are executed to update the contract state, and while generating the execution proof. Depending on the new contract state and proof, Layer 1 can verify the correctness of the transaction at a low cost. It is easy to observe the practicality of building the trust foundation of Rollup through trusted hardware.

One could imagine that Eigen has also greatly improved the efficiency of consensus and calculation while ensuring that the existing transaction and contract data are credible and protected. This points to Eigen's ability to create a vibrant privacy application ecosystem, and become the new infrastructure of Web3.

4.1.1 PrivDeFi

With the continuous expansion of the blockchain ecosystem, DeFi has turned a new page in financial services. Users can now control assets without a third-party intermediary, and then use services such as lending, asset management, and funds. The current DeFi features are as follows:

- The blockchain acts as the ledger that can synchronize all operations on-chain. Compared with traditional finance, DeFi does not need a settlement and clearing process Transaction processing, clearing, and settlement can all occur at the same time as the transaction broadcast.
- Open source and transparency are default, with the code and rules visible. Compared with the private systems and closed source codes of traditional financial applications, the blockchain is more likely to be discovered and attacked.
- Interoperability and programmability. The interoperability of front-end and smart contracts exposes more opportunities to arbitrageurs and hackers.

- The data on the chain is completely open and accessible. All operational and interactive behaviours, such as investment portraits, investment strategies, and label information, give arbitrageurs and hackers more opportunities to perform targeted attacks on users.

Eigen's ability to provide privacy protection is superior to the performance and architecture of the existing blockchains. Blockchain generally satisfies the trustworthiness of transactions through ledger openness and verifiability, and privacy is an extremely important part of financial services. The existing Monero and Zcash have some capabilities of privacy protection, but their technical architecture is not scalable, and it is difficult to be widely used in smart contracts.Eigen can provide privacy protection for smart contracts and on-chain operations to solve problems of privacy in DeFi. Eigen uses a decentralized mixing technology based on Merkle and zero-knowledge proof technology to help users hide addresses, while using trusted computing technology to achieve data protection, to provide DeFi transaction address and data privacy protection capabilities.

In short, while the transparency and developability of DeFi bring convenience to users, it also exposes some risks. Eigen provides privacy implements so that users can not only enjoy the openness and freedom of DeFi, but obtain a sense of autonomy and security.

4.1.2 Anti-MEV

MEV (Miner Extractable Value) is the most direct problem faced by the Defi ecosystem.It means that miners (or verifiers, sequencers) arbitrarily exclude or re-order transactions from unconfirmed txpool in the new blocks they produce to maximize additional profits.

The typical process usually involves: Bot manipulators use bloXroute's fast data streaming service to monitor new transactions in the mempool. Then, they copy the transactions of MEV robots, and replace them with their own transaction, thereby grabbing Arbitrage space for themselves. The generalized frontrunning transactions are considered malicious, or a transaction that destroys the stability of the ecosystem.

Typically, there are two methods to prevent malicious MEV. The first one is adopting off-chain fair sequencer.The classic implementation of this is Flashbot, which enables a sealed-bid block space auction mechanism for communicating

transaction order preferences. The other is encrypting the transaction. If the input and output of the transaction is encrypted, the miner cannot determine what means of sorting would bring about the maximum profit.

EigenSecret, based on the Eigen Network, is a cross-chain bridge and privacy wallet. All transaction information on EigenSecret, such as transaction input, output, trader information, chain data, transaction address protection, and other information, will be protected through Eigen. In Eigen Rollup, transactions are packaged through a fair sequencer, and are executed by miners. Miners cannot reorder transactions in Layer2 during privacy calculations, so Eigen limits the impact of malicious MEVs and PGAs.

4.2 Distributed Privacy-Preserving Computing Networks

Distributed Privacy-Preserving Computing Network is the infrastructure of Web3. The data circulation and fair transactions between cross-chain and cross-DAO on Web3 all rely on the private computing network to provide the driving force.

4.2.1 Decentralized Oracle

The Decentralized Oracle plays a crucial role in improving the interoperability between on-chain and off-chain, and is already used in many scenarios such as logistics tracking, insurance auto-compensation, and obtaining cross-chain information. On the one hand, the development of the Decentralized Oracle depends on the blockchain smart contract technology, while on the other, it helps the business extension of blockchain smart contracts.With the boom of applications in DEFI and Meta-universe, the Decentralized Oracle has a very bright future ahead.

The Decentralized Oracle adheres to the same decentralization principle as the blockchain. Multiple signatures or distributed algorithms are often used to ensure the correctness and consistency of data without introducing third-party institutions. However this is hard to implement, and performance could be plagued with bottlenecking.

The Decentralized Oracle powered by EigenCC makes it easy to protect the data collection process, and data anchoring on the blockchain. Based on the

EigenCC development framework, it's easy for developers to obtain the distributed data consistency, and gain the advantage of high performance and portability.

4.2.2 AI Model as Co-created Non-Fungible Token

With the advent of computers, human civilization was quickly pushed to a new peak. The reason why the computer has the status it has today is due to the fact that it can quickly store and process data. For tasks that humans have been unable to complete for tens of thousands of years, the computer may only need one second. The soul of a computer lies in data and calculations. With computers as the hardware foundation, the core of computing lies in models. Models are not only the foundation of artificial intelligence, but also the core engine to promote social development.

From the latest NFT report [18][19], the summer of 2021has been a boom for NFTs i. In July, NFT sales hit $2.5 billion, up from $13.7 million in the first half of 2020.The AI-generated NFT, which is a utility NFT, has also come to the public eye.

The Fetch.ai[20] is a leading British artificial intelligence lab aiming to build an open-access decentralized machine learning network and it has already made some progress in machine learning to generate NFTs. However, the collaborative construction model NFT faces serious privacy protection issues. The data provider is most concerned about how to prevent the risk of data privacy leakage in the training phase, which requires the use of Eigen Network.

On the Eigen network, the collaborative creation of the model NFT can be easily realized. The model requester can initiate a model training task, let each participant provide encrypted data through EigenBox, and then conduct encryption training through EigenCC, and finally obtain a valuable machine learning model. Meanwhile, the data providers are rewarded with the token as incentive for data sharing.

As shown in Figure 10, users provide encrypted pictures in Eigen Network, the EigenBox privacy environment decrypts and trains the model, then generates a new NFT asset, which is upgraded to form a positive cycle.

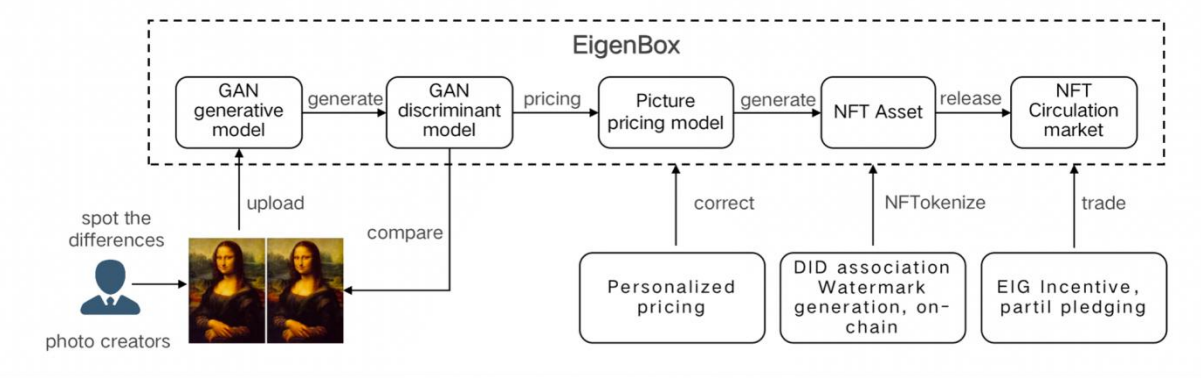

Figure 10 Model NFT Training

Through Eigen, once the AI model NFT production process is well protected, data sharers will become more willing to contribute their own data, which would bring stronger liquidity, and naturally bring higher value models.

Such a model NFT has a clear demand and great value, such as an on-chain investment model, a decentralized advertising recommendation model, and so forth. The model NFT transaction and circulation would create tremendous social value and promote the developments of various industries.

4.2.3 Privacy-Preserving Inference For AI Model NFT Circulation

Privacy-preserving Inference means the input and the model are all ciphers, and both of the participants are not aware of each other's plain data or model, and jointly compute the inference process in a smart contract. Privacy-preserving Inference realizes the circulation of an AI model's right of use under the protection of the AI model's ownership. This would eventually eliminate the data discrimination and data monopoly in the Web2 world.

For example, Eigen can help realize distributed medical information integration and AI diagnosis. Individual health information and medical records are very private personal data. It is imperative to return private information control to its owner.Medical practitioners can utilize advanced AI technology to conduct preliminary diagnoses, while the private details remain safe and secure.With the Eigen Distributed Privacy-preserving Computing Network and privacy inference, preliminary diagnoses could be realized without removing data from local storage or networks. This would allow for timely monitoring of health problems, and prompt prevention and treatment.

Similarly, data privacy protection can be implemented on terminal devices powered by the Eigen Privacy-Preserving Computing Network and applied in scenarios such as smart homes and unmanned driving.

Reference

[1] Securing a Blockchain with a Noninteractive Zero-Knowledge Proof. 2019. https://www.altoros.com/blog/securing-a-blockchain-with-a-noninteractive-zero-knowledge-proof/

[2] LAYER 2 ROLLUPS. https://ethereum.org/en/developers/docs/scaling/layer-2-rollups/

[3] Brendan McMahan, Daniel Ramage. Federated Learning: Collaborative Machine Learning without Centralized Training Data. 2017.

[4] Intel Software Guard Extension. https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html

[5] Wu XH, He YP, Ma HT, Zhou QM, Lin SF. Microarchitectural Transient Execution Attacks and Defense Methods. Journal of Software, 2020, 31(2): 544-563(in Chinese). http://www.jos.org.cn/1000-9825/5979.htm

[6] mbrandenburger, g2flyer, 2018. Fabric-private-chaincode. https://github.com/hyperledger/fabric-private-chaincode. 2021.

[7] Phala Network. https://phala.network/

[8] Sharon Goldberg, Dimitrios Papadopoulos, Jan Vcelak. Verifiable Random Functions (VRFs). Draft-goldbe-vrf-01. https://tools.ietf.org/id/draft-goldbe-vrf-01.html

[9] C. Tjhai, M. Tomlinson, G. Bartlett, S. Fluhrer, D. Van Geest, O. Garcia-Morchon, Valery Smyslov. Multiple Key Exchanges in IKEv2. Draft-ietf-ipsecme-ikev2-multiple-ke-00. https://tools.ietf.org/id/draft-ietf-ipsecme-ikev2-multiple-ke-00.html

[10] BitcoinWiki. Pay-to-Pubkey Hash. https://en.bitcoinwiki.org/wiki/Pay-to-Pubkey_Hash

[11] Interactive proof system. https://en.wikipedia.org/wiki/Interactive_proof_system

[12] Arbitrum: Fast, Scalable, Private Smart Contracts. https://offchainlabs.com/arbitrum.pdf

[13] Truebit: the marketplace for verifiable computation. Sina Habibian. Mar 25, 2018.

https://medium.com/truebit/truebit-the-marketplace-for-verifiable-computation-f51d1726798f
[14] Arbitrum: Scalable, private smart contracts. https://par.nsf.gov/servlets/purl/10069458
[15] Michael Walfish, Andrew J. Blumberg. Verifying computations without reexecuting them. Communications of the ACMVolume, 58 Issue, 2 February 2015, pp 74–84. https://doi.org/10.1145/2641562
[16] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. 2016, https://arxiv.org/abs/1602.05629
[17] Wikipedia. Kernighan–Lin algorithm. https://en.wikipedia.org/wiki/Kernighan%E2%80%93Lin_algorithm
[18] Mapping the NFT revolution: market trends, trade networks, and visual features. Matthieu Nadini, Laura Alessandretti, Flavio Di Giacinto. August 19, 2021. https://arxiv.org/pdf/2106.00647.pdf
[19] NFT sales volume surges to $2.5 bln in 2021 first half. July 6, 2021. https://www.reuters.com/technology/nft-sales-volume-surges-25-bln-2021-first-half-2021-07-05/
[20] Fetch.ai to use AI, ML for collaborative NFT art. Ledger Insights. August 25 2021. https://www.ledgerinsights.com/fetch-ai-to-use-ai-ml-for-collaborative-nft-art-blockchain/