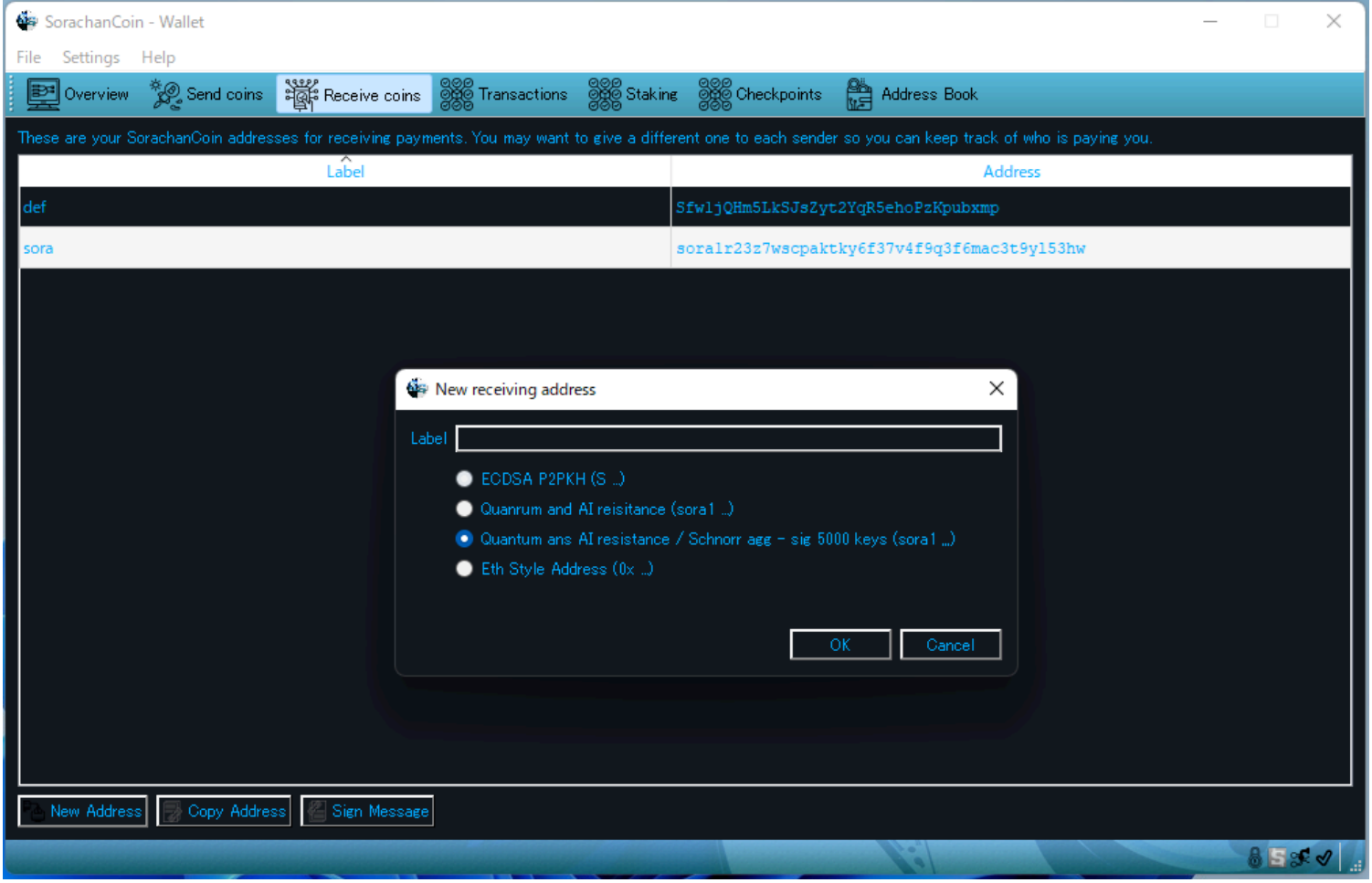
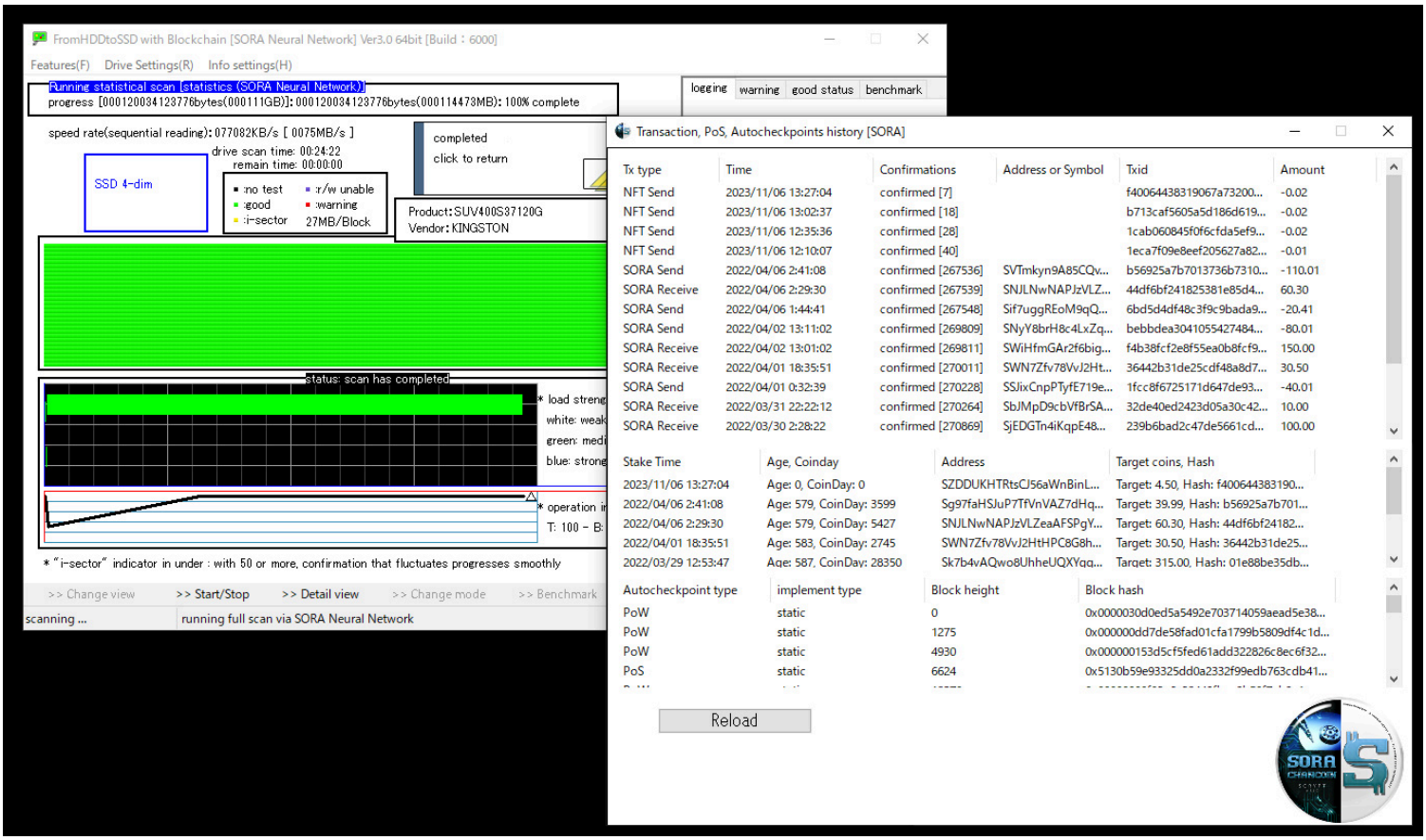
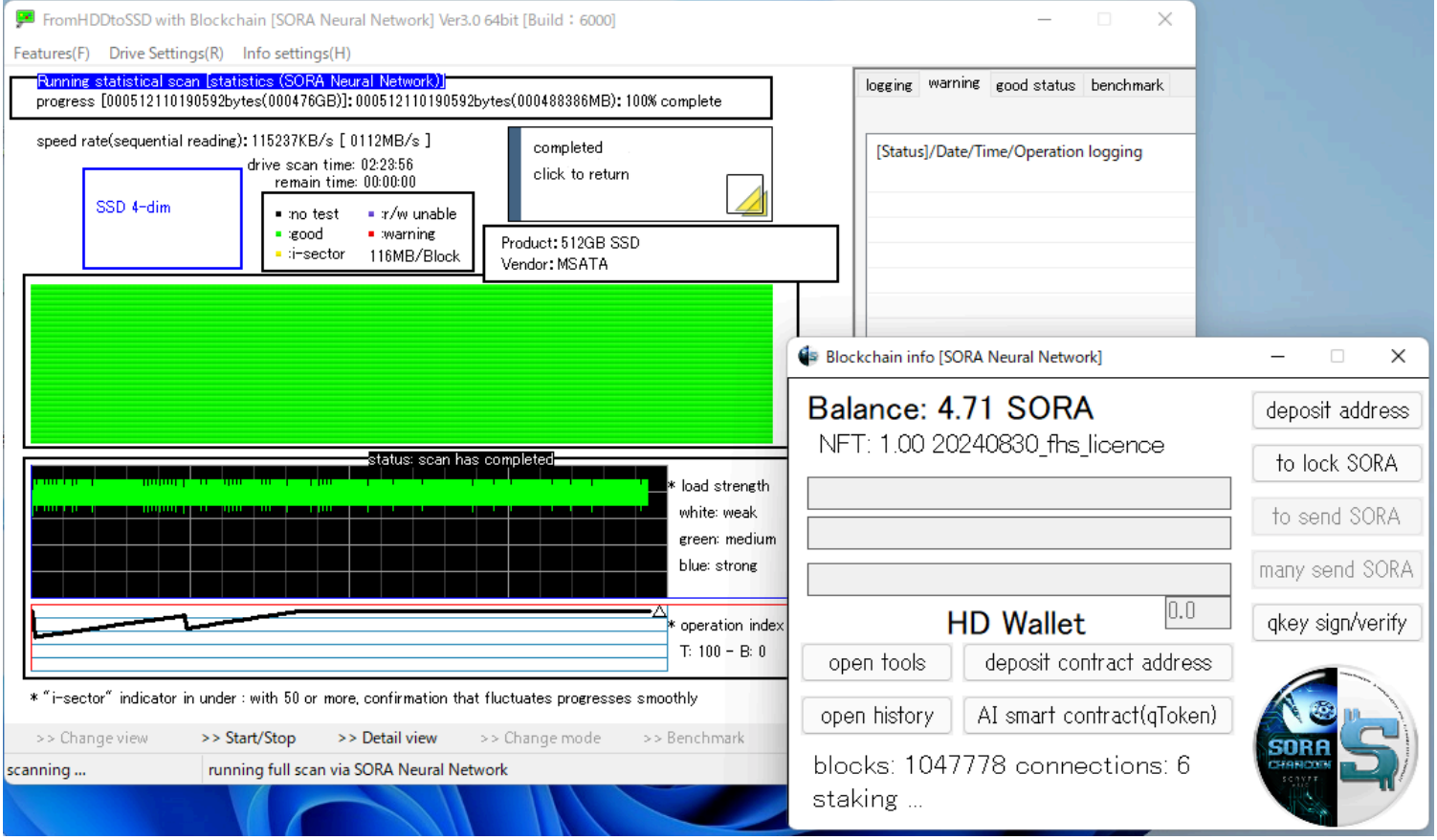
SORA-QAI Whitepaper

SORA Blockchain L1 is SorachanCoin-Core.exe, which encompasses the keys developed for this project. SORA Blockchain L2 is managed by FromHDDtoSSD.exe, which incorporates smart contract AI-NFTs that integrate AI reasoning with the blockchain. Both are operating normally on the SORA mainnet.



SorachanCoin-Core.exe SORA L1 Blockchain [Quantum resistance and Schnorr agg – sig 5000 keys]





FromHDDtoSSD.exe SORA L2 Blockchain [Smart Contract AI reasoning AI-NFT]

1, Overview

This blockchain enhances the traditional ECDSA with powerful security features, including quantum resistance, AI resistance, and protection against side-channel attacks, all achieved through multi-signature technology.

Additionally, by integrating AI reasoning into the SORA blockchain, we have strengthened the security of the memory pool and enabled integration with other statistical processing and business logic as a Layer 2 solution.

2, ECDSA

This is a public-key cryptography method utilizing the secp256k1 elliptic curve, which is defined by the equation $y^2 = x^3 + 7$. The number of scalar multiplications from the base point serves as the private key, and the coordinates on the elliptic curve become the public key.

The security relies on the property that it is computationally infeasible to reverse calculate the number of scalar multiplications from the coordinates on the elliptic curve back to the base point. This is the basic public-key cryptography method adopted by major cryptocurrencies such as Bitcoin and Ethereum. In SORA, it is supported for addresses starting with “S” in the Base58 format.

3, Suddenly, Please Discard the Myth that Cold Wallets Are Safe

It’s well-known that a wallet connected to the blockchain is called a hot wallet, while a wallet that is disconnected is referred to as a cold wallet.

Based on this, it is often said that a cold wallet meets the standard for being safe, right? But is that really true? Indeed, the image suggests that if it's not connected to the blockchain, it should be impossible to steal.

It's easy to think that way, but the reality is different. In fact, switching to a cold wallet does not reduce the risk of theft as much as one might think. We often hear the argument, "How can it be stolen if it's not connected?" But in reality, there are many methods to do so. Therefore, I must say, please discard the myth that cold wallets are safe immediately.

4, The Existence of Side-Channel Attacks That Can Easily Compromise Cold Wallets

Now, let's take a look at a method called a side-channel attack. This attack is not a direct assault; instead, it attempts an indirect attack on the blockchain.

The tricky part about this indirect attack is that it doesn't try to stop the system with a direct hit, but rather operates very quietly—this is the image you should have. Since it doesn't harm the system itself, it is difficult to detect.

While there may be no harm to the system, the danger lies specifically in its effect on wallets. On the blockchain, this type of attack requires special attention. The reason is that the structure of the blockchain itself is particularly vulnerable to side-channel attacks. Since this is an issue with the

very architecture of the blockchain, directly addressing it would mean rewriting the blockchain's structure, which would make it cease to be a blockchain, so that's not an option. Therefore, it is necessary to implement multiple indirect countermeasures, and in SORA, we have devised and integrated two types of solutions.

5, The Existence of Quantum Computers Capable of Ultra-Fast Periodicity Calculation

Next, let's talk about quantum computers. First, I want to strongly emphasize that this threat is not immediate; it's something for twenty years from now or even later, so there's no need to panic. I need to make this clear, because without this note, the mere mention of quantum computers could cause unnecessary turbulence in the blockchain market. Such reactions should be completely ignored.

The advantage of quantum computers lies in their ability to calculate periodicity at ultra-high speeds. You've likely heard the comparison many times: calculations that would take classical computers hundreds of billions of years can be done in a few hours by a quantum computer. This is due to their exponential time complexity, where the execution time increases exponentially with the problem size. By using quantum computers, these exponentially increasing operations can be significantly shortened through parallel processing, leading to the discovery of periodicity. The key here is the

periodicity, not the direct result of the computation. Since we cannot directly observe the computation result, periodicity serves as a substitute.

Even so, when attempting to reverse the calculation from a public key to a private key in RSA or ECDSA, having this periodicity information allows us to drastically reduce the computational effort. With the remaining information, a classical computer can complete the calculation to derive the private key from the public key. A simple analogy would be: if the periodicity is 8 for a value of 100, the remainder is 4, correct? Once we have this information (the remainder of 4), a classical computer can handle the rest of the calculation. Due to such vulnerabilities, it’s essential to consider resistance against quantum computers.

6, Quantum Resistance

Now, let’s discuss quantum resistance. If quantum computers are capable of ultra-fast periodicity calculations, then by eliminating that aspect, we achieve quantum resistance. Therefore, the solution is to use a multi-signature transaction with a public key system that is built on a concept other than periodic keys. Here, we focus on the post-parallel computation by quantum computers. After parallel computation, if none of the resulting solutions exhibit periodicity, the first condition for quantum resistance is fulfilled. As a result, a public key cryptography method that does not involve periodicity becomes a candidate, and hash-based keys are considered as a solution here.

Next, let’s look at resistance to the searching capabilities of quantum computers. There are algorithms that, after obtaining solutions through quantum computing, either calculate periodicity or shift the quantum bit state to its wave-like property. Even if the periodicity issue is solved, this wave-like property must still be addressed. The wave-like property determines the post-observation state based on the amplitude, meaning that if this amplitude is easy to change, the time needed to reach the desired solution is significantly reduced. However, this reduction only reaches the square root, so in practical terms, it doesn’t lead to a significant speedup, and the impact is minimal.

Thus, it becomes practically difficult to reflect the wave-like property in the search results of quantum bits, resolving the problems related to periodicity and searching. This leads us to the conclusion that hash-based public key cryptography can be used.

7, Schnorr Signatures

The issue of quantum resistance has been resolved. Next, we need to address the remaining resistance to side-channel attacks. In this context, we should look at Schnorr signatures. This is an algorithm that can assign a public key that satisfies linearity on the same elliptic curve used by ECDSA. Since it uses the same elliptic curve as ECDSA, Schnorr signatures can be implemented in addition to ECDSA.

8, Preventing the Discovery of the Private Key by Using Constant-Time Operations

When performing computations, computers are designed to finish tasks as quickly as possible. For example, if you're searching for one item out of 100, the computer starts from the beginning and stops as soon as it finds the item. This is the fastest method, right? However, black-hat hackers try to use even these small differences in execution time to gain clues about the private key. Such clues are the greatest threat to the blockchain because they can render all of the security measures ineffective. When all the security is disabled in this way, even cold wallets can be compromised. Therefore, resistance to side-channel attacks requires the implementation of the strongest possible mechanisms on the blockchain side.

This is where the concept of constant-time operations comes into play. It refers to ensuring that all computations take the same amount of time, regardless of the solution, and involves turning off optimizations. Normally, optimizing for speed is the rule, as it's better to solve problems with fewer operations. However, by intentionally disabling these optimizations and forcing constant-time operations, we can prevent side-channel attacks. This method helps ensure that no clues about the private key can be obtained through timing differences.

9, Implementing Constant-Time Operations Flawlessly and Thoroughly Is Extremely Difficult

However, when looking at the implementation of ECDSA (libsecp256k1), it becomes clear that implementing constant-time operations flawlessly and thoroughly is extremely difficult. Moreover, even if constant-time operations are omitted in certain parts, it doesn't result in a bug. The system will continue to function normally. The only issue is that processing time will vary, creating a situation that benefits black-hat hackers. If the system stopped working, it would be easier to catch these mistakes, but since it operates normally, it's very easy to overlook such issues even with careful implementation, making it a challenging task. In fact, OpenSSL has had many such bugs, right? While these issues didn't cause any problems from a functional perspective, they posed security risks. Implementations like these, where there are no apparent functional issues but security vulnerabilities exist, are incredibly difficult.

10, So What Should We Do? The Answer Lies in Applying the Law of Large Numbers from Statistics

The law of large numbers states that as the number of independent trials, each following the same probability distribution, increases, the average result of those trials converges to the true expected value. This means that even for phenomena with random variations, by increasing the number of trials, the average outcome becomes clearer.

For example, if you roll a die just once, you cannot predict which number will appear. However, if you continue rolling it thousands or tens of thousands of times, the numbers 1 through 6 will each appear approximately 1/6 of the time, stabilizing the overall result. This is the smoothing effect that occurs as the number of trials increases.

11, Targeting Smoothing of Processing Time with Schnorr Aggregated Signatures

Now, let’s talk about Schnorr aggregated signatures, which aggregate multiple Schnorr signatures into one. Aggregation is a technique that combines multiple keys into a single key. In SORA, we have successfully aggregated 5,000 private keys, each associated with a public key on an arbitrary ECDSA elliptic curve. After thorough verification and confirmation of no issues, we have deployed this to the mainnet.

Here, the “Law of Large Numbers” comes into play. If there is a mistake in the implementation of constant-time operations, what happens when you aggregate such keys? Even if small variations in processing time or gaps are introduced due to such mistakes, the aggregation process functions similarly to the repeated trials in the law of large numbers. Following this law, the small variations and gaps in processing time are smoothed out. As a result, black-hat hackers will find that side-channel attacks become ineffective.

Focus on the fact that the smoothed processing time information has no causal relationship with the aggregated private keys. It may seem like there is a correlation, but there isn’t. From the perspective of a covariance matrix, the more keys are aggregated, the more pronounced the off-diagonal elements become, breaking the correlation between the smoothed processing time and the private keys, making it impossible to infer the private keys. This is how it looks from a technical standpoint.

12, By Using Quantum Resistance and Schnorr Aggregated Signatures in Multi-Signature, You Can Achieve Both Quantum Resistance and Side-Channel Attack Resistance. This Implementation Is Realized in a New Method Called “SORA-QAI”.

By using quantum resistance and Schnorr aggregated signatures in multi-signature, you can achieve both quantum resistance and side-channel attack resistance simultaneously. This implementation is realized through a new method called “SORA-QAI.” This new implementation effectively utilizes the properties of OP_CHECKMULTISIG. Detailed documentation is provided at the following URL, so please take a look.

https://www.junkhdd.com/sora-qai.html

13, Other Features of SORA: Anonymous Encrypted Communication, etc.

SORA implements basic features such as staking and mining. In addition to that, it offers anonymous encrypted communication using dedicated addresses. With this encrypted communication, there is no concern about information leaking to anyone other than the parties involved. This is because the key exchange is done via the SORA blockchain, and the communication is encrypted based on that key. Please note the decentralized nature of this key exchange. Traditionally, key exchanges were centralized, meaning the central authority (the server administrator) always had the potential to eavesdrop on communication data.

14, Blockchain Key Exchange and Schnorr Aggregated Signatures

To achieve anonymous encrypted communication, key exchange is necessary. This key exchange allows the parties involved to share information known only to them, and by using this shared information to encrypt communication with symmetric key cryptography, anonymous encrypted communication is established. Here, we discovered a way to effectively utilize Schnorr aggregated signatures. By leveraging the linear aggregation feature, key exchange can be promoted, and anonymity can be successfully achieved. In other words, when receiving a message via this encrypted communication, if the message is sent anonymously, the recipient will not be able to identify the sender. This property is similar to how the private key cannot be derived from the public key. To break this anonymity, one would need to perform such calculations, which is challenging due to the exponential time complexity. This results in the establishment of anonymity, a decentralized feature unique to blockchain.

15, SORA L2 AI-NFT: AI Reasoning and Smart Contracts

For the SORA blockchain core (L1), we have implemented various security verifications and deployed them on the mainnet. Using this mainnet, we have implemented AI reasoning and smart contracts as part of the L2 network. While both L1 and L2 use the same SORA blockchain, they have different programs. In other words, they have been developed completely independently, and we are in the process of researching and realizing a new approach that utilizes the same SORA blockchain while keeping each function independent.

L1 includes the functionalities explained so far and is represented by SorachanCoin-Core.exe. L2, on the other hand, is responsible for integrating AI reasoning with the blockchain through the smart contract AI-NFT, which is handled by FromHDDtoSSD.exe.

16, Smart Contracts: Cumulative NFTs and Blockchain-Based Statistical Processing Utilizing Their Properties

The smart contracts on the SORA Blockchain L2 adopt a cumulative model. This allows users to create the desired functionality by issuing transactions that gradually build upon existing ones. The cumulative nature of these contracts aligns well with statistical processing. By constructing statistical processes on the blockchain, they inherit the properties of decentralization and distribution, ensuring that the statistics are free from bias. This pure statistical information is then incorporated into AI reasoning, forming a system that operates without interference.

17, Managing Ownership as a Basic Use of AI-NFT Smart Contracts

The smart contracts on the SORA Blockchain L2 follow a cumulative model, allowing AI-NFTs to be created and built up based on specific purposes. For example, in the case of managing ownership, you would first generate an AI-NFT that has no function by issuing a transaction that creates a single unit (quantity of 1) of the NFT. Next, if you issue a transaction that writes the hash of the digital data managing ownership into this single unit NFT, what happens? The ownership is managed by that single unit, and by issuing a transaction to transfer that unit, you can manage the ownership of the digital data represented by the hash written into the AI-NFT.

18, Advanced Use of AI-NFT Smart Contracts for Statistical Processing

The smart contracts on the SORA Blockchain L2 are based on a cumulative model, allowing AI-NFTs to be created and built up for specific purposes. One

advanced feature is to utilize AI-NFTs to handle statistical data for AI reasoning on the blockchain. One such use case is the inspection of SSDs. Even if SSD sectors are deemed normal during sector-level inspections, it is common for these sectors to fail soon after. In response, the SORA Blockchain L2 uses AI reasoning to analyze and detect such sectors that are prone to imminent failure, storing this control information as cumulative AI-NFTs. This enables a deeper investigation into the causes of SSD failure, benefiting from the decentralized nature of the blockchain. Through this approach, the system provides valuable insights into the factors contributing to SSD deterioration.

19. Conclusion

We have developed the above functionalities while enhancing the blockchain with a focus on security at the L1 layer. SORA Blockchain actively introduces features through consensus, emphasizing the decentralized and non-centralized nature of blockchain, to fully leverage the power of blockchain technology.

Lastly, we have lifted the restrictions present in BIP340. We have confirmed that Schnorr signatures can be handled without limitations using fixed-length public keys and signatures. During development, having no restrictions indeed reduces bugs and the need for asserts.

Blockchain Specifications	PoW(Scrypt) + PoS(Staking)

Consensus	ECDSA **Quantum AI-resistant keys** Schnorr signatures (**no even Y-coordinate restriction for public keys**) **Schnorr aggregated signatures (5000 keys)**
Block Hash	Scrypt
Encryption for Anonymous Communication	AES256 – bitcoin sha256
Uniqueness of scriptSig and scriptPubKey	bitcoin – hash160
Uniqueness of scriptSig and SORA-QAI	Merkle tree using bitcoin – hash160
Keys for Anonymous Encrypted Communication	Key exchange using Schnorr aggregated signatures (5000 keys)
NFTs (AI-NFT)	**AI-NFT can be encrypted and then traded through SORA. Since it is integrated into Layer 1 (L1), you can specify the amount of SORA to receive when trading the token.**
Ensuring Anonymity in Anonymous Encrypted Communication	Shuffle Schnorr aggregated signatures (5000 keys) The discrete logarithm problem serves as a barrier to identifying the sender's public key
Current circulating supply	https://us.junkhdd.com:7350/ext/getmoneysupply