Bitcoin on Solana: A Peer-to-Peer Electronic Cash System

October 31, 2008

1 Abstract

We propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes. In this adaptation, we integrate the protocol with the Solana blockchain, leveraging non-fungible tokens (NFTs) to democratize mining participation and enhance network efficiency. This design maintains the core principles of decentralization while introducing innovations in proof-of-work mechanics, incentive structures, and block production rates tailored to the Solana ecosystem. The security, growth, and stability of the network are underpinned by mathematical models that account for the fixed hash power from NFTs, accelerated halving, and dynamic difficulty adjustments.

2 Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust-based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a system for electronic transactions without relying on trust. We start with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we propose a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best-effort basis. Nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

In this Solana-integrated version, mining is facilitated through ownership of specialized NFTs on the Solana blockchain, each contributing a fixed hash power. This caps the total mining capacity at 10,000 units, promoting accessibility while preserving decentralization. Block times are reduced to 5 minutes, and the reward halving interval is accelerated to every 576 blocks, fostering rapid scarcity and enhancing long-term economic stability.

3 Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next

owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks for double-spending after each transaction. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint-based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced, and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

In the Solana context, transactions are broadcast across the network and validated using Solana's high-throughput consensus mechanisms, ensuring rapid propagation while maintaining compatibility with the proof-of-work overlay.

4 Timestamp Server

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

5 Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash, rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

In this Solana-adapted system, proof-of-work is innovatively tied to Solana NFTs. Each of the 10,000 available NFTs contributes a fixed hash power of 10 Solhashes per hour (SOLH/h). The total network hash rate H is thus bounded by $H \leq 10,000 \times 10 = 100,000$ SOLH/h. Mining participation requires staking these NFTs in pools, where the pool's aggregate hash power determines its probability of solving a block.

The network difficulty D adjusts dynamically based on the number of active NFTs, N, such that the expected block time remains 5 minutes. The difficulty adjustment formula is:

$$D = D_{\text{prev}} \times \frac{T_{\text{target}}}{T_{\text{actual}}}$$

where $T_{\text{target}} = 300$ seconds (5 minutes), and T_{actual} is the average time over the previous adjustment period (e.g., 2,016 blocks). With faster block times compared to the original 10 minutes, adjustments occur more frequently to maintain stability.

This NFT-based approach democratizes mining by capping participants and integrating with Solana's ecosystem, reducing energy consumption relative to traditional hardware mining while preserving security. The fixed hash power per NFT stabilizes the network, as fluctuations in H are limited to changes in N, making it more predictable than variable hardware setups.

6 Network

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes. 2. Each node collects new transactions into a block. 3. Each node works on finding a difficult proof-of-work for its block, leveraging staked Solana NFTs. 4. When a node finds a proof-of-work, it broadcasts the block to all nodes. 5. Nodes accept the block only if all transactions in it are valid and not already spent. 6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, other nodes may receive one or the other first. In such a case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

Integration with Solana allows for near-instantaneous transaction propagation, enhancing network responsiveness and efficiency in handling varying transaction volumes.

7 Incentive

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended, channeled through Solana NFTs.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered

circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

In this system, the block reward starts at 1000 BTC and halves every 576 blocks. The reward for block height h is:

$$r(h) = 1000 \times 2^{-\lfloor h/576 \rfloor}$$

Mining pools aggregate NFT hash power, and the pool solving the block receives the reward, distributed proportionally among participants. With blocks produced every 5 minutes, rewards are disbursed approximately every 5 minutes. This accelerated halvingcompared to the original 210,000 blocksinduces faster scarcity. For example, after the first halving at block 576 (approximately 2 days at 5-minute blocks), the reward drops to 500 BTC. The total supply S at block h can be approximated as:

$$S(h) = \sum_{k=0}^{\lfloor h/576 \rfloor - 1} 576 \times 1000 \times 2^{-k} + (h \mod 576) \times 1000 \times 2^{-\lfloor h/576 \rfloor}$$

approaching 1.152 million BTC asymptotically, but reaching 99% issuance sooner due to the tighter schedule. This rapid approach to maximum supply enhances deflationary pressure, potentially increasing value while requiring sufficient transaction fees post-issuance to sustain miner incentives and network security.

This structure encourages early adoption, as initial rewards are high, while long-term value accrues from deflationary pressure and economic stability.

8 Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree, with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.

A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 5 minutes, 80 bytes * 12 * 24 * 365 = about 6 MB per year. With computer systems typically selling with 2 GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2 GB per year, storage should not be a problem even if the block headers must be kept in memory.

9 Simplified Payment Verification

It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.

As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated

transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

10 Combining and Splitting Value

Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.

It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

11 Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.

As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

12 Calculations

The security, growth, and stability of the network fundamentally rely on the mathematical underpinnings of its protocol. We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. Let p be the probability that an honest node finds the next block, and $q = 1 - p$ the probability that the attacker finds the next block. The probability the attacker will ever catch up from z blocks behind is:

$$\begin{cases} 1 & \text{if } p \leq q \\ \left(\frac{q}{p}\right)^z & \text{if } p > q \end{cases}$$

Given $p > q$, the probability drops exponentially as z increases. In this system, with NFT-capped hash power, q is limited by the fraction of dishonest NFTs, enhancing resistance to 51% attacks. For example, if dishonest nodes control 30% of NFTs, $q = 0.3$, $p = 0.7$, and for $z = 6$ (30 minutes confirmation), the attack probability is approximately 0.00024.

13 Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity.

This Solana-integrated version advances the protocol by incorporating NFTs for mining, reducing block times to 5 minutes, and accelerating halving to every 576 blocks. These modifications enhance accessibility, efficiency, and scarcity, positioning the system as a viable evolution in the blockchain landscape. The CPU proof-of-work is channeled through Solana's ecosystem, ensuring compatibility with modern digital assets while upholding decentralization and security.