Timber Voting Protocol v0.1

Tomo Saigon, *Timber Network*

Abstract

In this paper we define a Timber Voting Protocol that enables token-weighted accounts on a primary chain to participate in voting processes on a secondary chain which may optionally enable confidential voting.

1. Definitions

A Decentralized Autonomous Organization (DAO) is a self-governing entity that executes decisions and permits actions through the outcome of voting mechanisms.

A Voting Token, as a blockchain-based asset, confers the owner with the right to participate in the governance of a DAO.

The Transport Protocol is a communication framework that interfaces with blockchains, facilitating the transmission of messages from one chain to another.

A Snapshot refers to the recorded state of token balances at a specific moment in time.

Finally, a Message is transaction-embedded data in a smart contract function invocation and is intended to be duplicated and transported to a corresponding smart contract on a separate blockchain.

2. Assumptions

Assume the existence of a transport protocol layer that facilitates message transmission between chains and is characterized by open and unrestricted accessibility and liveness allowing for the fulfillment of our throughput demands. Furthermore, the messages are delivered in a prompt and predictable fashion, maintaining the order of transmission and ensuring the integrity of the message content. The sender is able to verify successful receipt of the messages, while the receiver can confirm the completeness of the received messages.

In addition, we presume the presence of a decentralized autonomous organization (DAO) manager who possesses the necessary permissions to deploy contracts utilizing our protocol which enables the DAO to accept voting results calculated and transmitted from a secondary chain. Furthermore, it is assumed that the DAO manager has been granted the necessary permissions to utilize the previously mentioned transport protocol.

3. Solution

One challenge in implementing token-based voting is the accurate determination of the account entitled to utilize these tokens. For instance, a token holder may attempt to vote using their tokens, only to transfer them to a different account and attempt to vote again. A common approach to resolve this issue is to take a snapshot at a designated block height, thereby disregarding token transfers for voting purposes. Our proposed Voting Protocol takes a snapshot of all eligible voters and their token holdings prior to the initiation of the voting process on a proposal. A basic implementation can be carried out through a smart contract on the primary blockchain that iterates over all eligible token holders to calculate a Merkle root, while a more trusted implementation would involve the use of a DAO manager to execute the same process in a public manner that can be independently verified and replicated to yield consistent results. The snapshot process must generate a Merkle tree that records all eligible voting token holders and their balances, with the root of the Merkle tree being committed and transmitted through a cross-chain message, ensuring agreement between the two chains. As an alternative, the DAO manager could be granted permission to write the Merkle root directly on both the primary and secondary chains.

With the Merkle tree settled, the DAO can now initiate a proposal with a unique identifier, the Merkle root, start and end time for voting, a random seed used to hash addresses during registration, as well as any other parameters needed to specify the voting process on this proposal. This is accomplished through the transmission of a message from one chain to the other, thereby synchronizing the two chains.

In the subsequent step, voters indicate their intention to participate in the voting process by writing a message that includes a Merkle witness, which comprises their address and the hash of a second address (not necessarily contained within the Merkle tree and hashing the address for protects its identity until it is publicly revealed later) on the secondary chain to which they delegate voting rights. The address on the secondary chain may or may not be owned by the same individual, while the address on the primary chain must match the address associated with the transaction signature. The Voting Protocol verifies the witness, records the registration, including the attached hashed address and voting power determined by voting token balance, and prevents future messages from the same signing address from registering again. It is permitted for multiple addresses on the primary chain to delegate to the same hashed address on the secondary chain. The registration message must also reference the unique proposal identifier.

The secondary chain will accept any Registration message originating from the primary chain, as long as it appears to be from the correct contract address. Measures must be taken to detect any tampering with the message during transit. This is achieved by signing the message with the sender's private key and verifying that the signature matches the signer of the message and the address to be registered, thus preventing a monkey-in-the-middle attacker from writing a different registration for that address. Additionally, the contract on the secondary chain must be capable of detecting any disruptions to the registration process, whether due to technical failures or deliberate censorship by the transport protocol. This can be accomplished by tracking message IDs and verifying that they are received in the expected order, in order to detect any missing messages. To handle such scenarios, the contract on the secondary chain should temporarily queue any out-of-order messages and process them once the preceding messages have been received.

The closure of the registration process can be initiated by the first chain through the submission of a close request that contains the Merkel root of a Merkel chain encompassing all registered entities. Both chains must synchronize and update the Merkle root of this Merkel chain to arrive at the same value, assuming all registrations were successfully received. In the event of censorship of any registrations, the close request for the registration process would fail. This verification can also be executed by the underlying session protocol.

The second chain is prepared to accept votes cast from users who can demonstrate that their account, by hashing its address, corresponds to a registered hashed address for a specified token holder address. The address is hashed with the random seed generated in the proposal initiation phase and this prevents the hashed address from being matched with an account which had voted in a previous proposal. Additionally, the second chain can receive a Merkle witness that includes the token holder address, allowing for the verification of their inclusion in the Merkle tree by computing the Merkle root and comparing it to the previously stored value.

4. Conclusion

In conclusion, the proposed voting protocol offers a secure and efficient solution for cross-chain voting that incorporates the use of a Merkle tree to ensure accurate attribution of token ownership and a session protocol to detect any potential tampering. By leveraging the privacy-focused capabilities of a blockchain such as Secret Network, the protocol offers the option to keep voting activity confidential, further enhancing its security and privacy. Once the voting period has ended, the results can be tallied and the outcome can be determined, with the option to keep the details of the results private if desired. This approach provides a flexible and adaptable framework for conducting secure, transparent and private voting across multiple chains.