ShadowNouns: An Efficient Privacy Preserving Voting Protocol Optimized for Nouns

By Dexpresso Team

1 OVERVIEW

The proposed voting protocol is a privacy-preserving, anonymous voting framework for eligible voters (holders of Nouns NFT) based on zero-knowledge proofs and homomorphic encryption to ensure that voters can cast their ballots without revealing their identities or the contents of their votes. The protocol is designed to be deployed on any EVM-based blockchain (namely, Ethereum), and is intended for use in situations where only holders of a specific NFT (holders of Nouns NFT) are eligible to vote or contribute to the voting ceremony. The proposed protocol has following key features:

1. **Encryption of the vote:** registered voters encrypt their vote using a homomorphic encryption scheme, which result in votes being tallied (summed-up before final decryption) without revealing the contents of individual votes.

2. **MPC contributor registration:** we propose a novel approach to give Nouners more control and options for better UX for each voting ceremony, in addition to decreased time and gas effort. To this end, any Nouner, who wishes to participate in the creation of public key for the MPC-HE and later in MPC decryption of summed-up votes, shall register as an MPC-contributor. Each MPC-contributor can be rewarded for their effort. Moreover, contribution in the MPC ceremony does not affect Nouners right to vote and any Nouner (who may or may not be already participating in the MPC ceremony) can register and vote anonymously in every voting ceremony.

3. **Voter registration:** eligible voters submit a one-time use ballotID (as a nullifier) to the contract, which they will use later to cast their vote anonymously. The eligibility of the voters is checked on-chain by the smart contract during the registration request.

4. **Zero-knowledge proofs for vote validity and Merkle Path:** the voter generates a zero-knowledge proof that their encrypted vote is valid, without revealing any information regarding their vote. Additionally, another proof is generated by the voter that ensures inclusion of their BallotID in the Merkle tree and validity of its nullifier. If both proofs are valid, the contract records the submitted encrypted vote and optionally, performs a homomorphic summation of the encrypted vote to previously submitted (and also encrypted) votes.

5. **Decryption of the votes:** after the voting period reaches its on-chain deadline, the contract will no longer accept any votes and the final homomorphically encrypted summation of votes will be ready to be decrypted to reveal the results. Based on the scenario that Nouns DAO suggests, the MPC decryption can be done on-chain (each contribution to the decryption requires a transaction) or off-chain (the server for this scenario will be open-sourced and multiple instances can be run at the same time to ensure integrity).

6. **Anonymous delegation of voting power:** eligible voters (Nouns holders) gain one voting power per each Nouns NFT instance that they hold. The right to vote in the voting ceremony is bounded to the nullifier submitted by the NFT holder during the voter registration phase. Therefore, the NFT holder can delegate their voting power anonymously and off-chain to any identity as they wish. In addition, each voting power can be delegated to a natively multisig EOA (such as Schnorr-based signatures or MPC-ECDSA accounts) to construct more complex voting scenarios. We note that the entire delegation process is done off-chain and therefore preserves privacy of actual voters.

7. **Support for multisig contract wallets:** because of the ability to delegate voting power (6th feature), owners of multisig contract wallets can (and should) anonymously delegate their shared voting power to EOA wallets that can provide valid signatures and proofs necessary for the voting process.

In the following sections, we provide more detailed description regarding each step of the proposed voting protocol.

2 ASSUMPTIONS

Our protocol is designed without any limiting assumptions. However, for the sake of completeness, we would like to mention the following general assumptions that underpin the proposed protocol:

- The Ethereum (or any EVM-compatible) blockchain is used for storing and processing transactions related to the voting ceremony.
- The underlying cryptographic schemes (mentioned below) are remained secure based on the hardness of Discrete Logarithm, and Integer Factorization problems:
    - Elliptic curve cryptography (ECC), including ECDSA and Schnorr signatures
    - zkSNARKs
    - Pailliar/ElGamal Homomorphic Encryption schemes
    - Pederson Commitments

  It is noteworthy that all of the abovementioned assumptions also hold for most public blockchains, such as Bitcoin and Ethereum. Moreover, there are multiple reductions from zkSNARKs, ElGamal, Paillier and Pederson Commitments to widely used problems, including Discrete Logarithm that is base assumption for the same public blockchains.

3 TECHNICAL DETAILS

Here are the technical details for main steps of the protocol:

3.1 BASIC FUNCTION SIGNATURE (SMART CONTRACT)

Our proposal for a private voting system is designed to provide a secure and efficient method for conducting voting ceremonies using smart contracts. By leveraging the factory pattern for smart contracts, we are able to create a flexible and scalable system that allows for the creation of a new contract for each voting ceremony. Figure 1 provides an abstraction view of the smart contract system for the proposal.

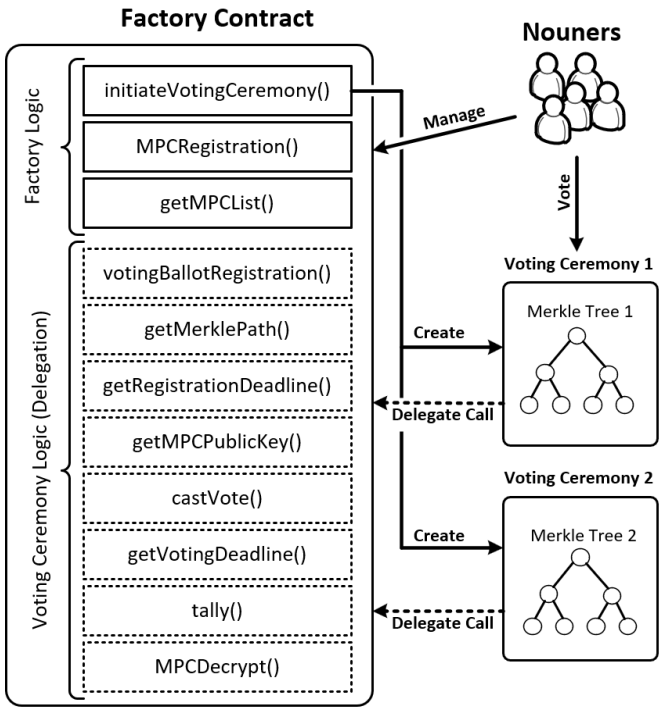

Figure 1: Abstract overview of Smart Contract System

Each voting contract serves as a secure storage location for the Merkle tree, which enables us to ensure the integrity and privacy of the votes. The factory contract provides the ability to initiate a voting contract and offers the option to choose between different lists of MPC's for voting on the proposal. This allows for customization of the voting process to meet the specific needs of that certain proposal.

To enhance the security of the system, we offer the option to use older NFT holders who have already committed to previous voting, ensuring that only trusted members are participating in the voting process. However, if necessary, the system also allows for a completely new registration process to sign up new holders. The registration for the MPC also takes place in the factory contract, ensuring that the process is seamless and secure.

Importantly, our system employs a one-contract-per-voting ceremony design that enhances scalability and ease of management. Additionally, to minimize the computational overhead, the ZKP trusted set-up is performed once in the factory contract and does not need to be repeated for each voting contract. This reduces the time required for setting up each voting ceremony and improves the overall efficiency of the system.

3.2 Optional (Rewarding) Contribution in MPC Ceremony

In many cases, some voters may not be interested in getting involved in creation of the MPC HE public key, but instead they would only want to participate in the voting. Our proposal separates voting registration from MPC pubkey ceremony to provide more comfort and options to choose from for the Nouners community. In Our proposal, eligible voters are not forced to contribute in the MPC ceremony for creation of the shared homomorphic encryption (HE) public key. On the other hand, any Nouner, who wishes to contribute to the security of voting protocol, can participate in the MPC ceremony for creation of the final HE public key, which is used for encryption of each registered voter's vote.

This approach provides Nouns DAO with a unique opportunity to construct an awarding/staking scenario for Nouners that contribute to the security and robustness of voting. Moreover, any Nouner, who wishes to only participate in the voting and does not want to get involved in MPC ceremony, can easily register as a voter-only entity and submit their vote anonymously.

This approach offers following advantages, compared to state-of-the-art private voting schemes:

- Easier UX for Nouners, who would like to only be involved in the voting, not contribution to the MPC-HE ceremony.
- Voting ceremonies can skip MPC-HE step and just delegate to the default (already existing) MPC pubkeys from the Factory contract.
- Nouners, who participate in the MPC-HE ceremony, can be rewarded based on their effort.
- Since contribution to the MPC-HE ceremony is optional and rewarding, there is an option for defining different collaterals and punishments method for Nouners, who fail to accomplish their part before decryption deadline of voting ceremony.

3.3 VOTER REGISTRATION AND BALLOT ISSUANCE

The smart contract has on-chain access to the list of eligible voters (holders of Nouns NFT) and can verify the ownership of the NFT using the ERC-721 standard allowing only eligible voters to participate in the election. Overall, each eligible voter performs following steps for registration:

1. create a random secret number $\boldsymbol{s}$ and calculates its hash $\boldsymbol{\mu} = \boldsymbol{h}(\boldsymbol{s})$. The calculated $\boldsymbol{\mu}$ will act as the unique ballot ID of the voter.

2. [Optional] Generate homomorphic encryption (HE) key pairs in an MPC-manner. Each Nouner, who wishes to participate in creation of the final HE Pubkey (used for encryption of the votes), generates their share of the key as follows:

    - $(s_i, p_i) \leftarrow HEKeyGen()$, where s_i and p_i are private and public key, respectively.

    - Currently, one the most MPC-friendly additive HE schemes is El Gamal, however, other partially HE schemes, such as Paillier can also be used.

3. The voter submits a registration request to the contract with following inputs:

    1. $\boldsymbol{\mu}$ as a nullifier for his/her ballot ID. Can also be a list of nullifiers: $[\mu_1, \mu_2, \dots, \mu_k]$. Note that k represents number of tokens that the voter holds.

    2. An integer number t, indicating the token ID of the Noun NFT that voter holds. Can also be a list of integers $[t_1, t_2, \dots, t_k]$. Note that k represents number of tokens that the voter holds.

    3. [Optional] p_i as their share of the final HE pubkey. The final HE pubkey will be constructed after registration deadline automatically as: $P_{HE} = \prod f(p_i)$, where $f(x)$ is designed depending on the underlying HE scheme (Paillier or El Gamal) used.

4. The contract verifies if the voter holds the given NFT token(s). Then, it stores the given nullifier(s) in its Merkle tree (as a new leaf) and updates its Merkle root accordingly.

5. [Optional] Moreover, the contract multiplies the given p_i to the value of its storage for previously added pubkeys. Finally, the storage will contain $P_{HE} = \prod f(p_i)$.

Once the registration period is over, the smart contract contains all of the eligible voters' nullifiers and the final P_{HE}. The proposed method in this step ensures that only eligible voters can participate in the election and their votes will remain private and anonymous using a homomorphic encryption scheme. Moreover, the final homomorphic key is constructed using a non-interactive MPC protocol on-chain.

3.4 VOTING

Every owner of Nouns NFT who has been registered successfully during the registration phase, can participate in the voting phase by providing a correct spending proof for the specific BallotID that they've stored its nullifier in the contract. In summary, the voter executes following steps to submit a successful vote:

1. The voter encrypts their vote using the final P_{HE} from contract. It is noteworthy that the encryption has an additively homomorphic setting and all of encrypted votes are summed without requiring any decryption.
2. Using the same BallotID that voter wishes to use, they calculate a proof of
3. The voter retrieves the Merkle tree root and the path from the Merkle tree leaf node for their nullifier (BallotID) to the root from the contract.
4. The voter then computes a Merkle proof of inclusion for their nullifier using the path and the root. The Merkle proof is a sequence of hash values that proves that the nullifier is included in the Merkle tree.
5. Moreover, the voter computes a Pedersen commitment to the randomness used to generate the nullifier and the private key used to encrypt the commitment. The Pedersen commitment is a function of two inputs, a commitment value and a random blinding factor, and is computed as: $\boldsymbol{C} = \boldsymbol{r} \times \boldsymbol{G} + \boldsymbol{x} \times \boldsymbol{H}$ where $\boldsymbol{G}$ and $\boldsymbol{H}$ are generators of the elliptic curve used in the setup of the Factory contract, $\boldsymbol{r}$ is the randomness used to generate the nullifier, $\boldsymbol{x}$ is the private key used to encrypt the commitment, and $\boldsymbol{C}$ is the Pedersen commitment.
6. The voter then generates a signature on the Pedersen commitment using their spending key. The spending key is a 256-bit value that is used to prove ownership of the note for voting.
7. Finally, the voter submits the Pedersen commitment, the Merkle proof, the signature, along with their encrypted vote to the contract.

3.5 TALLYING

After reaching the on-chain deadline, the contract will no longer accept vote submissions and the final homomorphically encrypted aggregated votes is ready in the contract's storage. At this stage we offer two types of tallying approaches that can be set during the initialization of the voting ceremony by Nouners:

1. **Off-chain:** In this approach, participants in MPC-HE will be responsible for executing a trustless ceremony that can be easily achieved by an open-source code. Every individual entity will be able to run the MPC-HE ceremony locally to ensure the integrity, correctness and transparency of the final result. The main advantage of this approach is the lower costs. However, there is a computational effort to verify correctness of the final MPC decryption ceremony.

2. **On-chain:** In contract with the previous approach, every MPC participant updates the encrypted final voting result submitting their part of the MPC decryption to the contract. The final result will be visible on-chain after the last MPC participant performs/submits their decryption. The advantage of this approach is the on-chain transparency and state clarity of the MPC ceremony. This results in an easy-to-verify validity check for every entity on the blockchain. Moreover, due to the on-chain track of MPC participants, we can propose a punishment and incentivization method for players. On the other hand, each MPC decryption submission requires a transaction that can

increase overall effort of the contract. Additionally, we can employ account abstraction idea (EIP-4337) in order to further improve the UX of this approach.

4 SUMMARY

In conclusion, the proposed voting protocol combines the security of homomorphic encryption and zero-knowledge proofs with the transparency and audibility of blockchain technology to provide a secure, private, and trustworthy electronic voting system. While the protocol may be complex and require a high degree of technical expertise, its potential benefits in terms of security, flexible UX, privacy, and efficiency make it a compelling option for Nouns DAO. We have listed main Pros and Cons of the proposed voting protocol to the best of our knowledge:

Pros:

1. **Security:** The protocol leverages the security of homomorphic encryption and zero-knowledge proofs to protect against tampering, hacking, double-voting and fraud.
2. **Privacy:** The use of different commitment schemes and zero-knowledge proofs ensures that individual votes are kept private and anonymous, while the use of MPC homomorphic encryption ensures that no one can see/decrypt individual votes.
3. **Transparency:** The use of blockchain and smart contracts ensures that the voting process is transparent and auditable by anyone, providing a high degree of trust in the integrity of every voting ceremony.
4. **Accessibility:** The protocol can be implemented in a decentralized and distributed manner, making it accessible to anyone with an internet connection.
5. **Efficiency:** The protocol can be executed with minimal computational overhead (Most of the computations, such as proof generations, are done off-chain), making it scalable and efficient for large-scale voting ceremonies on blockchain.
6. **Flexible UX:** Unlike state-of-the-art privacy preserving voting protocols, in our proposal we have designed a method, in which eligible voters are not forced to participate in the MPC-HE ceremony, unless they volunteer to. To this end, less advanced Nouners can participate in the voting without the need to get involved in complex MPC ceremonies. On the other hand, any Nouner who wishes to enhance security and integrity of the voting ceremony by participating in the MPC-HE setup, will have the opportunity to be praised (qualitatively or quantitatively) by Nouns DAO for their efforts.
7. **Anonymous Delegation of Voting Power:** The proposed voting protocol provides Nouners with an option to anonymously delegate their voting powers to anyone.
8. **Support of Multisig Contract Wallets:** The proposed protocol supports (and is compatible with) different types of multisig contract wallets, such as Gnosis Safes.
9. **Optimized for Nouns:** The protocol is designed and optimized with Nouns ecosystem in-mind to provide maximum compatibility and flexibility that Nouners would want.

Cons:

1. Complexity: The use of complex cryptographic techniques such as zero-knowledge proofs and homomorphic encryption requires a high degree of technical expertise and may be difficult to implement for non-experts.

2. Computational Cost: The computational overhead required to execute some steps of the protocol may be high, leading to increased costs for running the election.

Overall, the pros of the proposed voting protocol outweigh the cons, and with proper implementation, it has the potential to significantly improve the security, privacy, and trustworthiness to reach a noticeable milestone in Nouns private voting.