Minimal Anti Collusion Infrastructure (MACI)

A protocol for running private on chain polls

Privacy & Scaling Explorations

July 10, 2024

Contents

1 Abstract

Minimal Anti Collusion Infrastructure (MACI) is a protocol that combines smart contracts and zero-knowledge proofs to enable private on-chain polls. Developed by Privacy & Scaling Explorations, MACI aims to provide a secure and fair approach to voting, particularly in the context of public goods funding and general elections. This paper presents an overview of the MACI system, its objectives, and its potential applications.

2 Introduction

Voting is a fundamental process in democratic decision-making, but ensuring the security, privacy, and fairness of voting systems has been a long-standing challenge. Traditional voting systems often suffer from issues such as voter coercion, vote buying, and lack of transparency. With the advent of blockchain technology, there has been a growing interest in developing secure and transparent voting systems that can address these issues.

The Minimal Anti Collusion Infrastructure (MACI) is a protocol that leverages smart contracts and zero-knowledge proofs to enable private on-chain polls. MACI aims to provide a secure and fair approach to voting, particularly in the context of public goods funding and general elections. The protocol is designed to ensure that votes are submitted privately, processed correctly, and tallied accurately, while preventing collusion and protecting voter privacy.

MACI involves two primary roles: voters and coordinators. Voters must register to participate in polls, and a robust Sybil resistance solution can be employed to ensure that only genuine users can sign up. Votes are submitted on-chain privately, as they are encrypted off-chain by the user. The coordinator is responsible for processing the messages and tallying the votes, with all processing steps being proved using zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs). These proofs are posted and verified on-chain, allowing voters to confirm the validity of the results.

The primary objective of MACI is to become a widely adopted and secure private voting system. In the short term, MACI aims to be used by projects in the public goods funding sector, ensuring that funds are distributed equitably to deserving projects. In the long run, the goal is to see MACI used

for general elections, where the stakes are high and bribery resistance is of utmost importance. Technically, MACI strives to be cost-effective, secure, and easy to integrate.

3 Methodology

MACI's workflow involves two key roles: users (voters) and a trusted coordinator. Users participate in MACI polls by signing up with their MACI public key and then voting on open polls. The coordinator is responsible for deploying the MACI smart contracts, initiating polls, tallying the final results, and finalizing polls by publishing the results on-chain.

The core smart contracts in MACI include:

1. MACI.sol: Responsible for registering user signups by recording the initial public key for each user and allowing the coordinator to deploy polls.

2. Poll.sol: Where users submit their votes. One MACI contract can be used for multiple Poll contracts, allowing users to vote on multiple issues.

3. MessageProcessor.sol and Tally.sol: Used by the coordinator to process user votes and prove on-chain that they correctly tallied each vote.

Each MACI Poll is a state machine with three stages: Open, Closed, and Finalized.

During the Open stage, users can sign up and vote. To sign up, users generate a MACI keypair and send the public key to the MACI smart contract. The criteria for user signups can be customized using a SignUpGatekeeper contract, which can require users to prove ownership of an NFT, receive an attestation on EAS, or pass a proof-of-personhood verification.

To vote, users bundle variables (public key, vote option, vote amount, etc.) into a "command," sign it with their MACI private key, and encrypt the signature and command together to create a "message." Users encrypt their vote using a shared key that only the user and coordinator know, preventing bribers from reading transaction data to see how a user voted.

During the Closed stage, the voting period has ended, and the coordinator must process all messages, tally the results, and publish the proofs on-chain. The coordinator uses the MessageProcessor contract to prove they have correctly decrypted each message and applied them to create an updated state tree. The coordinator processes messages in batches to avoid

exceeding data limits and creates a zk-SNARK proof that the messages and sign-ups have been correctly processed.

Once all messages are processed, the coordinator tallies the votes of the valid messages off-chain and creates a zk-SNARK proof that the total number of votes sum to the given tally result. The coordinator submits a hash of the correct tally results and the zk-SNARK proof to the Tally contract, which is verified by a separate verifier contract.

In the Finalized stage, the final results of the poll can be announced, and anyone can verify that the results have been processed and calculated correctly by the coordinator.

The use of zk-SNARKs ensures that the coordinator's final tally result is valid without revealing individual votes, while the shared key scheme prevents bribers from reading transaction data to determine how users voted.

4 Collusion Resistance & Anti Bribery Properties

A key feature of MACI is its design to provide collusion resistance and prevent bribery in digital voting applications. It achieves this through a combination of techniques, including message encryption and a unique key change mechanism. In MACI, voters are identified by their public key, which is associated with a private key used to sign and submit messages to polls. To prevent collusion and bribery, MACI allows users to change their voting key if it becomes compromised or if they wish to revoke past actions. Messages are processed in reverse order after a poll ends, ensuring that a voter's last legitimate message is counted as valid, while invalidating any subsequent messages signed with the old key, potentially by a briber. Put simply, a user can submit multiple votes where each new vote overwrites the previous votes.

The key change mechanism, combined with message encryption, makes it difficult for bribers to verify whether a voter has complied with their demands since a user could submit a valid vote from any key unknown to the briber. Votes are encrypted off-chain by the user before being submitted on-chain, further enhancing the system's resistance to collusion and bribery attempts. These properties are essential for ensuring the integrity and fairness of digital voting applications, particularly in the context of public goods funding and general elections.

5 Security Considerations

While MACI offers several security features, there are still some trust assumptions to consider.

The coordinator plays a crucial role in the MACI workflow, and a corrupt or inept coordinator could disrupt the voting process in certain ways. For example, a coordinator can decrypt votes, which could be used to publish individual votes or to bribe voters. Additionally, a coordinator can halt a round by never tallying the results or submitting the final proofs, effectively stopping the voting process.

However, there are limits to what a corrupt coordinator can do. A coordinator cannot publish incorrect results by censoring valid votes or creating fraudulent votes. The use of zk-SNARKs ensures that the coordinator must provide valid proofs of correct message processing and vote tallying. Furthermore, a coordinator cannot change the parameters of a poll, such as extending the voting deadline, once the poll has been deployed.

It is important to note that while a corrupt coordinator can stop a vote by never publishing the results, they cannot publish false results. This limitation helps maintain the integrity of the voting process, even in the presence of a malicious coordinator.

To mitigate the risks associated with a corrupt coordinator, MACI could explore ways to distribute the coordinator's responsibilities among multiple parties or implement a system of checks and balances to prevent a single coordinator from having too much power. Additionally, ongoing security audits and improvements to the protocol can help identify and address potential vulnerabilities.

6 Real-world Applications

MACI has the potential to be applied in various real-world scenarios where secure and private voting is crucial. Some of the current and potential applications include:

6.1 Public Goods Funding

MACI is being integrated into funding applications such as clr.fund and Gitcoin (via their Allo stack) to ensure fair and democratic distribution of funds to deserving projects.

6.2 Retroactive Public Funding Rounds (RPGF):

MACI is being used in RPGF with communities like EthMexico and has been discussed with Optimism for their RPGF rounds.

6.3 DAO Governance

Although not yet experimented with, MACI could be used for decentralized autonomous organization (DAO) governance to ensure secure and private voting on important decisions.

6.4 General/Local Elections

In the future, MACI has the potential to be used as a replacement for general or local elections, providing a secure and transparent voting system that protects voter privacy and prevents collusion.

6.5 Feedback Systems

MACI can be applied to any scenario where collusion is a problem, such as feedback systems where money or power is involved.

7 Future Work

The MACI team has identified several areas for future improvements and research:

7.1 Enabling Unconditional Voter Privacy

Currently, if a voter performs a key change, the coordinator could collude with a bad actor to inform them of the key change. To address this, the team aims to remove the link between the original signup key and the key used for voting. Users would sign up to vote via the MACI contract, prove they've passed the entry condition, and then sign up with a new key to polls deployed by the same MACI contract. By using a zk-SNARK circuit to prove knowledge of the preimage of a StateLeaf, voters can anonymously join polls with the new key, ensuring no link to the original key. A nullifier will prevent the same original key from being used to sign up more than once for each new poll. The drawback is an extra step for users to register to individual polls, which the team aims to offset with gasless transactions or moving some logic off-chain.

7.2 Message Structure Optimization

The team plans to replace the Merkle tree used for storing messages with a hash chain. This approach offers several benefits:

7.2.1 Unlimited number of messages

Hash chains do not have a depth limit like Merkle trees, allowing for an unlimited number of messages.

7.2.2 Removal of expensive merge operations

Hashing the previous hash chain with the message is cheaper than inserting into a Merkle tree, and removing the need for the coordinator to perform merge operations on accumulator queues will reduce costs and processing time.

7.2.3 Cheaper message sending

Only one hash is required to update the hash chain, making it cheaper to send messages.

7.2.4 Less constraints on circuits

Simplified logic leads to smaller circuits, as processing messages with chain hashes removes unnecessary inclusion proofs and requires only k hashes for k messages without extra signals.

7.3 Removing the Need for a Trusted Coordinator

Research is being conducted to explore ways to remove the need for a trusted coordinator, increasing the decentralization and security of the system.

7.4 Making the System More Accessible

The team is investigating methods to remove voters' costs, such as signup and vote costs, to make the system more accessible to a wider range of users.

8 Conclusion

MACI is a project that addresses the issue of collusion in voting systems by combining smart contracts and zero-knowledge proofs. The protocol aims to

provide a secure and private way to conduct on-chain polls, with potential applications in public goods funding, DAO governance, and general elections.

MACI's design incorporates several features to ensure voter privacy and prevent collusion, such as the use of zk-SNARKs for proving correct message processing and vote tallying, and the encryption of votes using a shared key known only to the user and the coordinator. However, the system also has some limitations, such as the need for a trusted coordinator and the potential for the coordinator to halt the voting process or decrypt votes.

As MACI continues to evolve and be adopted by more projects, it has the potential to influence the way voting and public goods funding are approached. Further research and development are needed to address the limitations and challenges of the system and to explore ways to enhance its security, privacy, and accessibility.

8.1 Additional Resources

- MACI Documentation
- MACI original post
- Collusion
- clr.fund
- MACI<>Allo