An Ethereum Based e-Voting System

Achilleas Spanos and Ioanna Kantzavelou(✉)

Department of Informatics and Computer Engineering, University of West Attica, Athens, Greece
{cs161048,ikantz}@uniwa.gr

Abstract. Developing an electronic voting system that would replace the old, traditional electing procedures has been a concern of many researchers for years. Blockchain technology could provide some guarantees for voting platforms, such as transparency, immutability, and confidentiality. In most research works, secure and reliable electronic voting systems are required to address known security, anonymity, and fraud issues. This paper presents a secure decentralized electronic voting system, named the EtherVote, which is based on the Ethereum Blockchain network focusing on eligible citizens' identification. The EtherVote is a serverless e-voting model, thus improving security, privacy, and election costs. An effective method for voter registration and identification to enhance security is proposed. Among the main properties the EtherVote holds are storing encrypted votes, efficiency in handling elections with numerous participants, and simplicity. The system is tested and evaluated, vulnerabilities and possible attacks are exposed, and a discussion examines opportunities for enhancing the proposed e-voting system.

Keywords: EtherVote · Blockchain · e-Voting · Ethereum · Smart Contract · Metamask

1 Introduction

There is a number of supportive reasons to replace in-person voting procedures or other special voting facilities, such as absentee voting, voting in a foreign country, early voting, or proxy voting, with electronic voting (e-voting). Through e-voting, equal voting rights could be provided to citizens facing access problems, such as people with disabilities and individuals in distant areas. Nevertheless maintaining and storing the votes in a database, which would be managed by an organization, incorporates the risks of running into over-authority and manipulated details, limiting fundamental fairness, privacy, anonymity, and transparency in the voting process. Central authorities could delete or modify votes. Even if the authority is trustworthy, an attacker could gain access to the database and modify or change votes and personal data. At the same time, old-fashioned paper voting is very complicated to verify and audit for a citizen who lacks control over the voting system. Blockchain is a new and promising technology that could be

L. A. Maglaras and C. Douligeris (Eds.): WiCON 2023, LNICST 527, pp. 202–223, 2024.
https://doi.org/10.1007/978-3-031-58053-6_14

used to address these problems towards e-voting solutions. The full potential of this technology has not yet been fully unveiled.

This promising technology, could face the primary challenges associated with a voting process, ensuring integrity and security of votes. By adhering to the blockchain's structural and operational principles, and given that votes as well as the authentication data are stored to new blocks referencing the preceding ones, an immutable ledger is established. Within this ledger, once information is recorded, any attempt to alter it would disrupt the interlinked relationships between blocks, rendering such modifications impossible. Thus, the technologies provided by blockchain, when combined with a robust encryption algorithm to safeguard votes and prioritize vote anonymity, along with a secure citizen identification system, have the potential to fundamentally transform the voting process in every country. At the same time, any citizen will be able to verify his vote, as opposed to the traditional way of voting, by getting the transaction hash value. This hash value serves as a verification mechanism for transactions within the blockchain, in our context, representing the act of casting a vote in our system.

In this research work, we propose an electronic voting system, the EtherVote, based entirely on the Ethereum blockchain using smart contracts. Since user identification is a major problem in such systems, especially when no database or classic server side is used, we will focus on adding or combining some authentication factors to validate eligible citizens with the right to vote, before and during the voting procedure. On platforms that are expected to serve any kind of social group, of all ages, ease of use is a critical factor. The Ethereum network is public and can be easily accessed by Metamask, - a self-custodial wallet used to safely access blockchain applications - whose addresses will be matched with citizens. That makes the user authentication, the voting procedure, and generally any interaction with the blockchain very easy to use. However, due to the public nature of the Ethereum Network, it becomes imperative to encrypt all the data stored within it. This encryption process is accomplished with a choice of an encryption algorithm that aligns with our specific requirements, including considerations such as the need for decryption capabilities or otherwise.

The specific contributions of this paper are as follows,

1. Introduced innovation with a decentralized e-voting system that goes beyond singular focal points, offering a comprehensive and secure solution spanning every stage of the voting process.
2. Serverless e-voting model, relying solely on Ethereum and smart contracts.
3. Effective method for voter registration and identification, leveraging advanced techniques and cryptographic functions for enhanced security.
4. Encrypted storage of all votes, maintaining the anonymity of each vote.
5. Efficiently handle a large volume of votes, ensuring its practicality for real-world elections with significant voter participation.
6. Simplicity and ease of use ensured for individuals across all social groups.

In the subsequent sections, this paper unfolds a comprehensive exploration of an innovative blockchain-based voting system. The discussion commences in

Sect. 2 with a review of related work, delving into existing systems that laid the groundwork for our approach. Moving forward in Sect. 3, the paper provides a detailed examination of the system's operation, specifying its architecture and mechanisms to provide a good understanding of its functionality. Subsequently, the focus shifts to the practical aspects of system implementation, presenting results and insights derived from the entire voting process. The examination extends to test cases in Sect. 4, offering a rigorous evaluation of the system's performance. Possible threats and existing vulnerabilities are examined in Sect. 5. In the last section, the paper discusses and addresses limitations and problems encountered, and provides ideas for further work.

2 Related Work

Many proposals and research papers have been developed for blockchain voting systems. A thorough survey [1] that provides a complete comparison between the very recent Blockchain-based methods adopted by electronic voting systems, establishes the state of the art and exposes the achievements of such efforts. Authors have gathered and compared all the techniques adopted by related systems in many areas such as cryptography, citizen identification, resistance to attacks etc. Using these outcomes as a point of start, we propose and implement a fully functional and comprehensive system designed to provide a significant advancement to the existing e-voting systems and proposals.

Below, we will delve into research papers and system proposals, pinpointing instances where techniques were not only adopted but also meticulously refined to suit the intricacies of the proposed system.

According to research [1], the only method for verifying voter identity that hinges solely on the utilization of a phone number and consequently by employing SMS is delineated in paper [5]. Access to the voting process is exclusively granted to individuals who hold an active phone subscription.

i-Voting: Estonia is the first country to introduce online voting in national elections since 2005 [2], using an electronic ID chip [3]. This identity generated SHA1/SHA2 signatures and was used to identify citizens. The voter would have to download the app, authenticate, and then the voting process would follow. The vote is encrypted with the elections public key, and the user's private key. Then the vote is stored on a server controlled by the Estonian government [4].

Unique Identifier: Many proposals have emerged for the Indian electoral system, leveraging the UIDAI Aadhaar, a unique identifier assigned to each registered Indian citizen. Such approaches are inspired from the Estonian system, wherein the Aadhaar identifier represents a private key, in conjunction with the election's public key, to generate a digital signature for voting purposes [6]. Another research publication from 2020 introduces an innovative blockchain-based voting system, merging the Aadhaar number with biometric authentication. To participate, voters must preregister using a virtual ID obtained by UIDAI. Asymetric encryption is used to verify votes. Voters fingerprint is converted to the digital signature enhancing the security of the entire voting process [7].

Multi-factor Authentication Voting: This model proposed in the research paper [8] uses a database housing voters identity, along with the phone number and other personal information. To participate, each voter is required to complete a registration process and establish a unique personal identification number (PIN). Subsequently, all eligible voters authenticate themselves, with their ID number and PIN, after witch the voting process takes place, by entering a one-time password (OTP), generated during the authentication phase. The voting records are ultimately securely stored within the blockchain.

In the research [9], researchers proposed an election system that work as follows. To participate, each voter is required to complete a registration process by furnishing their ID number and other pertinent personal information, which are stored within a newly created block. This system integrates each registered voter into an electoral list. Subsequently, the election process takes place, with the prompt display of the election results immediately upon the conclusion of the voting process.

Indonesian researchers have put forward a blockchain-based voting system that leverages the use of **Metamask** for voter registration. Every user possessing a valid address must be registered by the administrator, thereby designating their Metamask address as eligible for voting. The counting of the votes, is automatically conducted by the smart contract that orchestrates the entire electoral process [10]. In research paper [11], a system incorporating Metamask is also put forward. The paramount aspect of this research endeavor lies in the distribution of Metamask addresses, to voters, with a sufficient amount of Ethers, which will later be employed within the electoral procedure for authenticating themselves.

Save system, is an electronic voting system proposal for university elections. In this system, every voter is identified by the election authorities. During this identification process every voter receives a random magnetic card, that contains a 13-digit number. By the use of that card, each voter is identified as a valid voter [12].

Reasearchers proposed an Ethereum-based electronic voting system, that uses a blockchain-based Interplanetary File System (IPFS) storage method. Proposes a new way to guarantee confidentiality based on a database. Each voter is required to register by establishing a unique member ID and password, subsequently utilized to generate a new address. Addresses and member ID values undergo encryptionsusing the AES algorithm and before being stored to the database. On the contrary, votes are stored in the Ethereum blockchain, taking advantage of the distributed edger [13].

Numerous research papers and models of e-voting systems have been proposed and are worth acknowledging. Several of them use biometric identification, such as fingerprint and eye recognition. Despite the extensive body of research, proposed systems invariably exhibit shortcomings in one domain or another. The fulfillment of the requirements for secure citizen identification, preservation of voting confidentiality, ease of use, and accessibility across all societal segments remains a formidable challenge.

The purpose of this research work is the creation of an electoral system, with an emphasis on the secure, low cost and fast identification of citizens, which will be carried out with the help of Metamask and the use of multi-factor authentication on the Ethereum public network.

Our work builds on the methodologies proposed in the systems above mainly in the field of identification. The combination, and variation of these authentication methods, while introducing an additional, unique identifier like the Metamask address to citizens, could potentially offer the safest, quickest, and most comprehensive authentication method for a voting platform, all without relying on any central authority database. In our paper we will elucidate the rationales underlying the enhanced nature of our system compared with already mentioned and existing systems in various facets encompassing the prevoting phase, the process of identification, the conduct of the voting procedure, vote tally, the preservation of vote integrity, and the safeguarding of citizens' fundamental rights.

We present a comprehensive report of the security measures and future pitfalls inherent in the proposed model. The biggest concern in such proposals, lies in the public infrastructure of the Ethereum Blockchain and thus in addressing personal data leakage.

3 Proposed System

3.1 Functionality

In the proposed system, the EtherVote, the sole service provided by the electoral authority is to record the list of eligible citizens on the blockchain. The process of citizen identification, the conduct of the voting procedure, the storage of votes as well as the vote tally, will be exclusively centered on data storage and the invocation of functions from our smart contract. Every transaction and interaction with the smart contract, whether for identification or for voting is public and can be easily traced, while keeping all personal data private by utilizing suitable cryptographic techniques. Systems that use the blockchain as a database, inherit the immutability and therefore modification or deletion of information is impossible. Each voting citizen will be assigned a Metamask address - with a necessary number of ethers, during the entire election procedure. The electoral procedure is divided into four phases, as described in the sequel and illustrated in Fig. 1.

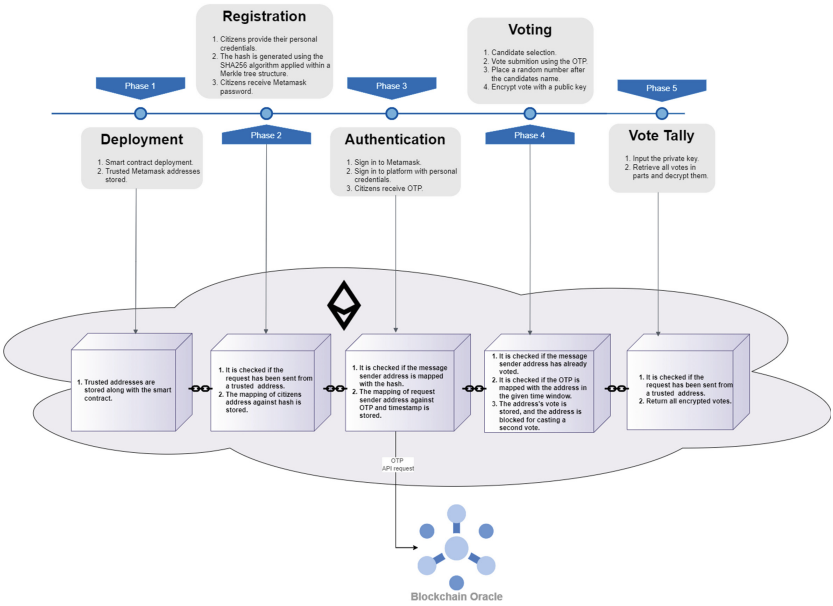

Fig. 1. Voting phases

Upon familiarizing ourselves with the overarching structure of the system through a comprehensive diagram depicting its four primary phases, we shall acquire an in-depth comprehension of the specific functionalities that each phase encompasses. To achieve this profound understanding, we will delve deeper to understand the functions that each phase performs by elucidating the intricate functions and operations inherent to them. This will be followed by a cross functional diagram governing each phase.

This detailed approach, ensures that we have garnered a comprehensive overview of all processes unfolding within each phase of the system. In this way, we will be able to have a better understanding during implementation, but more importantly in the security and vulnerability analysis that we will see in the next chapters.

1. The **first phase** (Fig. 2) consists of writing and storing the smart contract on the Ethereum blockchain. Initially, Metamask addresses are created, which will be considered 'trusted' and will belong to the electoral authorities. Through these addresses, the smart contract will be created, the results of the elections will be taken, but also the sensitive personal data of citizens with the right to vote will be stored in the blockchain, aiming on using these personal data on authentication.
2. In the **second phase** (Fig. 3), every eligible citizen is asked to register on the platform. In order to register, he must either attend or contact the authorities. The authorities must be connected to one of the 'trusted' addresses (presented

in **phase1**), and after creating a new Metamask account, with which the citizen will be assigned and identified, is registered on the platform by linking the newly created address with the citizen's personal information, such as ID number, first name, last name and phone number. Although the variables that will store the information are private, since the Ethereum network is public, anyone with a copy of the blockchain will be able to retrieve this private information. To address this risk, personal information are combined, encrypted with the cryptographic algorithm SHA256, and matched as a key-value pair with the address assigned to each citizen. Upon completion of the second phase, each citizen receives the password for the Metamask account with which they have been matched.

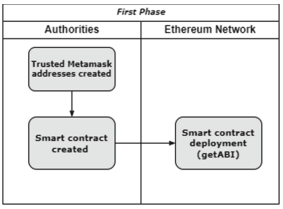

Fig. 2. Phase 1

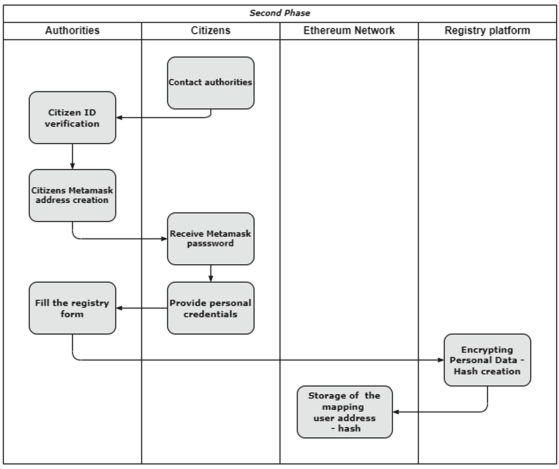

Fig. 3. Phase 2

3. The **third phase** (Fig. 4) is the process of identifying voters on election day. Once they enter the voting platform, they must log in to their Metamask account by entering the password they received on the day of registration, thus creating the first identification parameter. Then as a second identification parameter, they will have to connect to the platform, entering their personal information, with which the specific address has been assigned. Similar to stage 2, the data is combined and encrypted with SHA256, followed by a check to match the transaction's sending address with the hash. If the second authentication stage is successfully completed, a unique code (OTP) will be generated. This code is stored in the Blockchain, with the time of its creation, and is matched with the address corresponding to the current voter, as a key-value pair, just as the match was made with the personal information. Saving the time the OTP was created is to have a window of time to use it. Finally, each voter receives this unique code via SMS to the mobile phone they have registered.

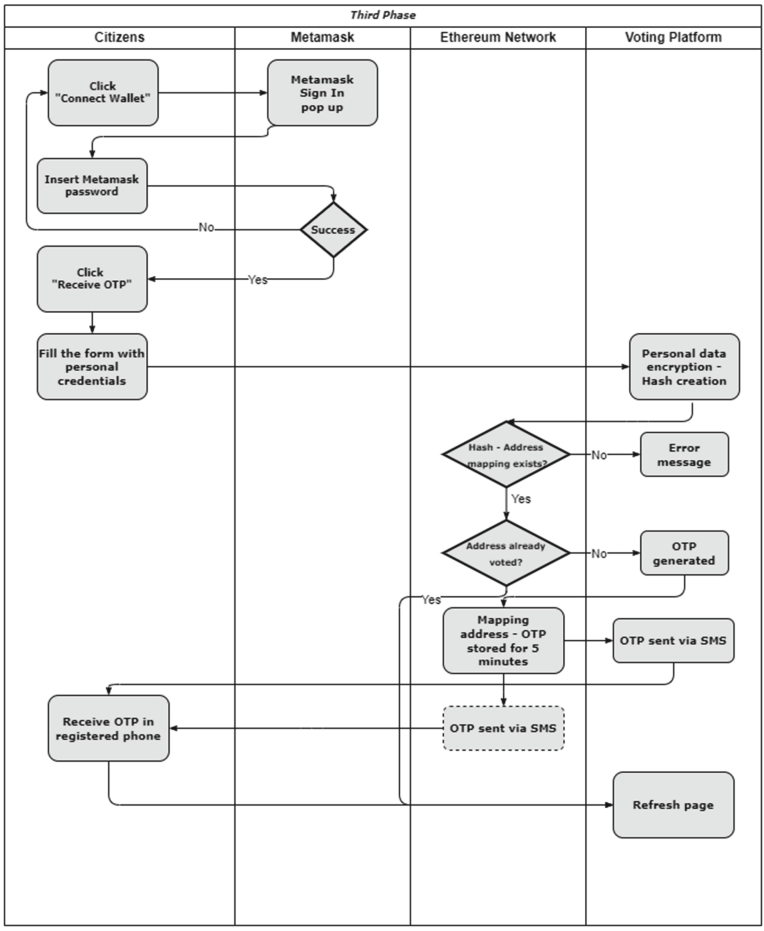

Fig. 4. Phase 3

4. **Phase 4** (Fig. 5) concerns the voting process. Each voter having performed the identification, and having received the OTP on the mobile, is invited to vote, choosing from a list of candidates. After the candidate is selected, the OTP received during identification is requested in order to accept the vote. In the case that the OTP is incorrect, or the time limit has expired, he must receive a new OTP, performing the second stage of identification. In the opposite case, the transaction is done, a random number is appended to the candidate, and the vote is encrypted with a public key and RSA encryption algorithm. This encrypted vote is stored on the blockchain and the s voter's address state is modified to a locked status, ensuring that only one vote can be added, thereby preventing multiple votes. If the Metamask address, associated with the voter, has already been used to cast a vote, during the second

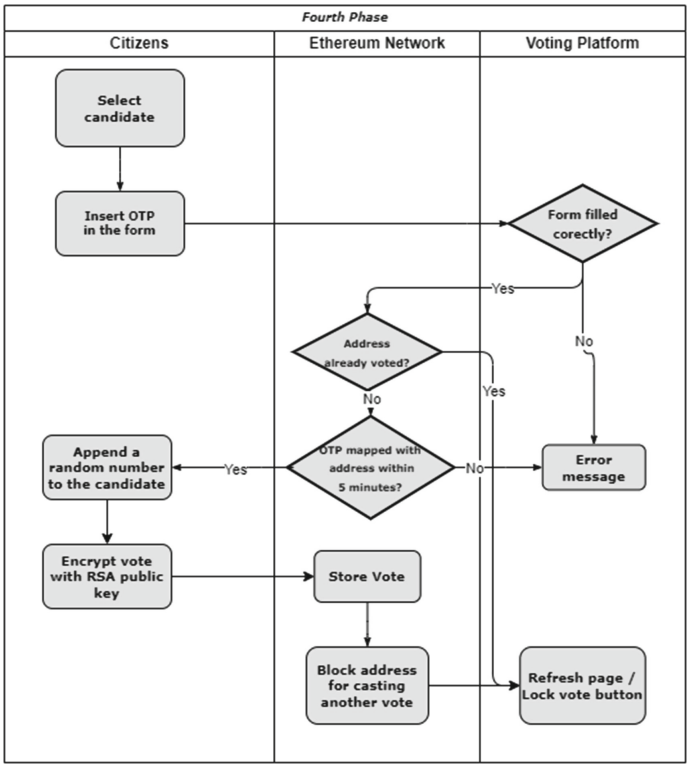

Fig. 5. Phase 4

stage of identification, no OTP will be generated or sent. This eliminates the possibility of submitting a second vote.

5. Finally, **phase 5** (Fig. 6) concerns the vote tally. Government officials must authenticate themselves through a trusted Metamask address, as only requests originating from such trusted addresses will be able to retrieve the encrypted votes. To start the process, they must initially complete the form by providing the RSA private key necessary for decrypting the received encrypted votes. Subsequently, our smart contract will return the votes in batches, organized into teams of 10,000 votes per call, to ensure compliance with the gas limit (running **out of gass**). Once each vote is successfully decrypted, a validation process will ensue to verify whether any instance of a candidate's name is included within the decrypted text. If a name of a candidate is detected, votes associated with the specific candidate will increment by one, repeating this process until all votes have been decrypted and added.

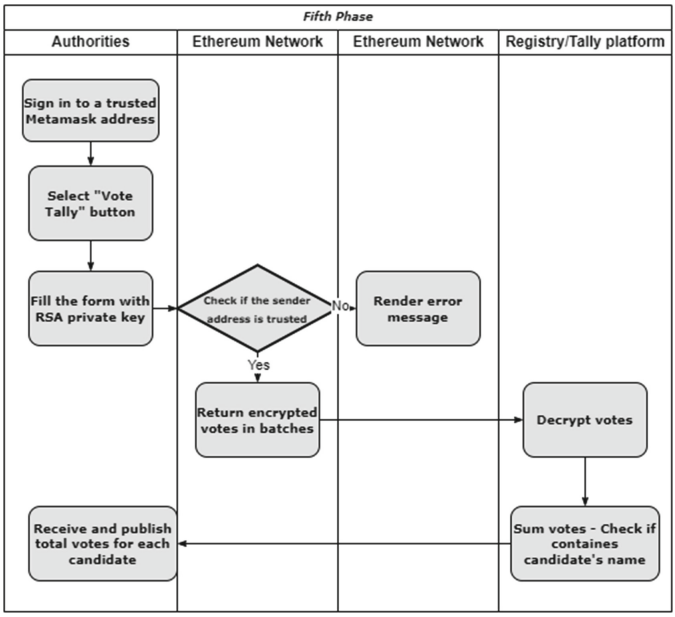

Fig. 6. Phase 5

Through the utilization of blockchain technology, every transaction is subjected to encryption, thereby safeguarding the sanctity of personal data. This approach not only ensures the accurate registration of each voter but also establishes a strict prohibition against multiple voting instances. Our primary concern revolves around the integrity of the vote itself, rather than the identity of the voter who casts it.

Within this framework, any address maintained within the blockchain that has not previously exercised its voting privilege is granted the sole entitlement to do so. This approach diligently upholds the fundamental principle of "one-man-one-vote," a cornerstone of fair and equitable electoral processes. By maintaining this level of transparency and security, we bolster the trust and integrity of the entire voting system while preserving the anonymity and privacy of individual voters.

3.2 Implementation

This section elaborates on the construction and evaluation of the entire system, which was implemented and tested on the Ethereum test network. By doing so, we acquired the essential outcomes required to assess both the security of our system and its functionality.

In the development of our system, we employed ReactJS for the frontend, while exclusively relying on Solidity for the backend. This strategic decision

aligns with our initial objectives, which aimed to eliminate the use of any other backend or database beyond our smart contract and the Ethereum blockchain.

Below, we will outline the comprehensive process that must be meticulously followed to successfully conclude the voting procedure. This process commences from the inception, which is the deployment of the smart contract, and culminates with the reception of the voting results, as depicted in Fig. 7.

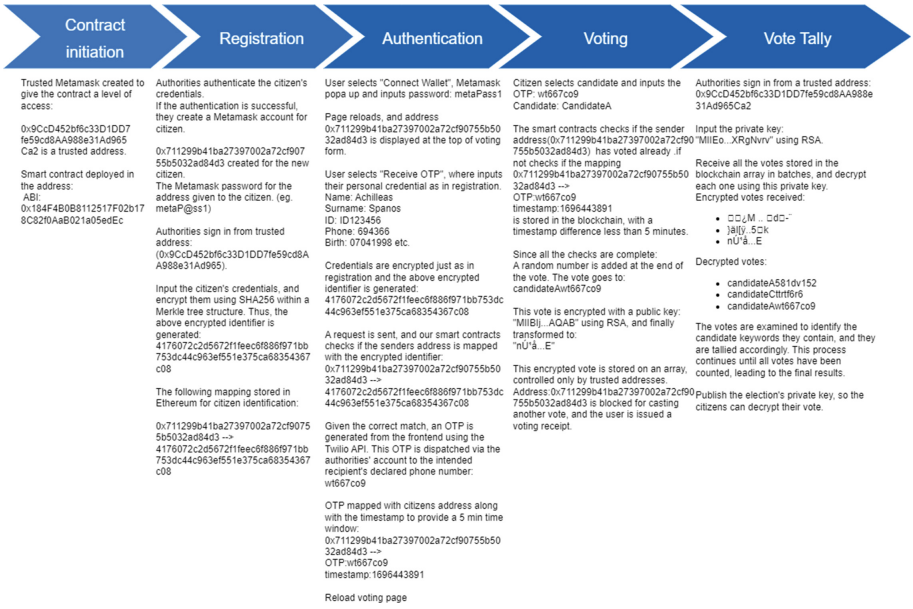

Fig. 7. Implementation through Phases

Contract Initialization: Trusted Metamask addresses are generated, allocated to regulatory authorities, and employed for the regulated invocation of specific smart contract functions. This mechanism ensures that the necessary voting operations maintain a precisely controlled and requisite level of access. *"0x9CcD452bf 6c33D1DD7fe59cd8AA988e31Ad965Ca2" added in contract as trusted.*

Following the creation of a smart contract, we retrieve the contract's Application Binary Interface (ABI) and utilize it within our front-end interface to invoke the contract's functions. *Contract ABI: "0x184F4B0B8112517F02b178C82f0AaB 021a05edEc"*

Citizen Registration: The authorities authenticate each citizen's identity and verify it's voting eligibility before enrolling them on the voters' list. Following the successful authentication, a Metamask account is instantiated, and the corresponding password is imparted to the respective citizen.

Address: *"0x711299b41ba27397002a72cf90755b5032ad84d3"*
Password: *metaP@ss1*

The credentials are disseminated to the citizen, following which the authorities sign into a trusted address to initiate the voter registration process on the blockchain. The newly generated address, along with the citizen's credentials, is incorporated into the registration form by the governing authorities. Subsequently, these credentials undergo encryption utilizing the SHA256 algorithm within a Merkle tree structure. The resultant root hash, used as an user authentication hash, is then recorded in the blockchain, associating it with the voter's listing.

User authentication hash: *"4176072c2d5672f1feec6f886f971bb753dc44c963ef551e375ca68354367c08"*

This mapping will enable users to authenticate themselves during the elections without the necessity of recalling any password; solely their personal credentials will be required.

Authentication: To successfully complete this phase, it will be essential to employ functional components from both the user interface and the smart contract. It's crucial to bear in mind that this process encompasses multi-factor identification, and the successful outcome of this authentication enables the storage of the vote.

The multi-factor authentication process of our system includes the following subprocesses in the following order:

- Upon selecting the 'Connect Wallet' button, the Metamask is triggered, prompting the citizen to sign in to their registered account using the confidentially provided password.
  Enters password: *metaP@ss1*
- Subsequently, the user will be prompted to click on the "Receive OTP" button. Upon doing so, a window will emerge, prompting the voter to input their personal information, mirroring the procedure followed by authorities during the registration process. This user-provided data will be subjected to encryption, and an identification hash corresponding to this encrypted information will be generated using the same method as in registration.
  User authentication hash: *"4176072c2d5672f1feec6f886f971bb753dc44c963ef551e375ca68354367c08"*
  The authentication process will subsequently validate whether the address linked to the sender of the request ("msg.sender") corresponds to the received authentication hash.
- Assuming successful verification in the preceding two checks, an OTP (One-Time Password), crucial for enabling a successful vote, is generated and provided to each citizen. This OTP will be securely recorded on the blockchain, mapped with the citizens address, along with the timestamp. This method allows for a 5-minute window during which the OTP can be used.

Mapping stored: *"0x711299b41ba27397002a72cf90755b 5032ad84d3" − > OTP: "wt667co9" timestamp: "1696443891"*

Subsequently, the OTP is dispatched to the user's phone through the government's official account using the Twillio API, or any messaging API.

Upon successfully completing these four identification parameters (included the identification checks during the registration process), citizens are prepared and authorized to cast their votes.

Voting: The sole action required from the voter, is to choose a candidate and input the OTP into the designated form.
OTP: *wt667co9*, Candidate: *CandidateA*

Following, our smart contract confirms the association of the OTP with the sender's address (which represents the citizen) within the designated time limit by scrutinizing the timestamp disparity. Additionally, it ensures that this address has not yet participated in the voting process, initiating the process of encrypting and securely storing the vote in the blockchain commences. Before being stored on the blockchain, all votes are subjected to encryption using the RSA algorithm with a common public key. To prevent the generation of identical encrypted strings for votes pertaining to the same candidate, a random number or the OTP utilized for the vote will be appended to the candidate's name. This approach ensures that each encrypted vote not only remains unique but also carries the candidate's name, enabling the counting process through decryption of the vote.
Final candidate format: *candidateAwt667co9*
RSA public key: *MIIBIj...AQAB*
Vote stored: *nÚ¹å...E*

This vote is stored in an array, and the address's status is updated, to prevent casting another vote.

Vote Tally: Authorities gain access to the votes by inputting the private key into a designated form. To manage the decryption efficiently and prevent gas limit issues, votes are received in batches. Once received, they undergo decryption.

A comprehensive tally is then conducted, cross-referencing the decrypted votes to identify the included candidate selections. Finally, the election results are compiled and printed for public disclosure.

Encrypted votes:	Decrypted votes:
1. ¿M .. d-¨	1. candidateA581dv152
2. äI⌋ÿ..5k	2. candidateCttrtf6r6
3. nÚ¹å...E	3. candidateAwt667co9

After the voting results publication, the private key may be published, so each citizen may decrypt its vote.

4 Testing and Evaluation

4.1 Test Cases

By conducting test cases, we not only assess the system's functionality but also gain a comprehensive understanding of the array of controls and security parameters it encompasses. The initial section exclusively pertains to the initial two phases, encompassing all pre-election procedures, and is illustrated in Table 1.

Table 1. Pre-elections test cases

Pre-elections dApp test cases			
ID	Test case description	Expected result	Result
1	Safe/authorized Metamask account created	Authorized address created and stored	Pass
2	Smart contract deployment	Get smart contract's abi	Pass
3	Authorized Metamask address connected to dApp for citizen registration	Metamask account connected with Ethereum network	Pass
4	Create voter's Metamask address	Password and address generated	Pass
5	SHA256 encryption of voters personal data	User authentication hash generated	Pass
6	Register voter using the authorized Metamask address	Authentication hash mapped with address	Pass
7	Registering a voter with incomplete form details	Render error message	Pass
8	Register voter using unauthorized Metamask address	Personal data encrypted but not stored/Error	Pass

Table 2 below showcases the test cases for one of the pivotal phases we have explored, focusing on user identification during the third phase of the system. These test cases are instrumental in verifying the robustness and effectiveness of our user identification protocols in this critical phase.

Table 2. Authentication test cases

Voter authentication test cases			
ID	Test case description	Expected result	Result
1	Click on "Connect Wallet" to connect dApp with Metamask	Metamask triggers	Pass
2	Connect dApp with Metamask	Displayed address upon successful Metamask connection	Pass
3	Click on "Receive OTP" dApp connected to Metamask Authentication hash matches with address Account has not voted	OTP generated, mapped for 5 min and sent via SMS	Pass
4	Click on "Receive OTP" dApp connected to Metamask Authentication hash does not exist (wrong credentials) Account has not voted	User not found Error	Pass
5	Click on "Receive OTP" dApp connected to Metamask Authentication hash exists but does not match with address Account has not voted	User not found Error	Pass
6	Click on "Receive OTP" dApp connected **not** to Metamask	Error	Pass
7	Click on "Receive OTP" dApp connected to Metamask Authentication hash matches with address Account has already voted	Refresh page	Pass

In Table 3 encompasses all the test cases related to the voting process. Having a good understanding of this phase's functions will greatly aid in our complete understanding of the subsequent phase, which deals with the vote tally procedure.

Table 3. Voting process test cases

Voting test cases			
ID	Test case description	Expected result	Result
1	dApp connected to Metamask Address has not voted OTP exists and matches with address OTP has not been expired	Random number appended to candidate Vote encrypted Vote stored Adddress blocked for voting again	Pass
2	dApp connected to Metamask Address has not voted OTP exists and matches with address OTP has been expired	OTP error	Pass
3	dApp connected to Metamask Address has not voted OTP exists but not matched with address OTP has not been expired	Voting Error	Pass
4	dApp connected to Metamask Address has not voted OTP not matched with address	OTP error	Pass
5	dApp connected to Metamask Vote with address that has already voted	Blocked by dApp and smart contract function	Pass
6	Vote while dApp is **not** connected with Metamask	Error	Pass

Finally, here are the test cases concerning the vote count, as we can see in Table 4.

Table 4. Vote tally test cases

Vote tally test cases			
ID	Test case description	Expected result	Result
1	dApp is connected with Metamask via trusted address Select "Vote tally" Insert correct private key	All votes are received in batches Votes are decrypted and summed up	Pass
2	dApp is connected with Metamask via trusted address Select "Vote tally" Insert **incorrect** private key	Vote tally error	Pass
3	dApp is **not** connected with Metamask or connected with a not trusted address	Error	Pass
4	Receive votes before the end of the elections	Error	Pass

We are now fully equipped to comprehend both the system's operation and the entirety of the controls implemented throughout the voting process.

5 Security Analysis

In this section, a critical exploration of potential vulnerabilities will take place. We will delve into an examination of various threats, that would oversee the principles of voting procedure, but possible attacks such as MITM as well.

5.1 Potential Threats in Voting Principles

The reliability of the voting system is imperative to ensure the democratic process and foster public trust in political governance. Within this subsection, we will scrutinize the extent to which a blockchain-based voting application aligns with the principles of democratic voting, such as the recognition of human dignity, the principle of 'one person - one vote,' and the transparency of the process.

Double Voting Threat: By associating citizens with their Metamask addresses and locking these addresses upon casting a vote, the submission of a second vote by any citizen is effectively prevented. This correlation, coupled with the existence of a one-time-password generated through the input of personal data known only to each citizen, renders the 'theft' of a vote impossible. Additionally, the transmission of this password via SMS to the mobile phone registered on the day of enrollment necessitates the physical presence of the citizen or continuous communication with the individual who funded the vote purchase. Due to these stringent security parameters, the alteration or coercion of a vote can only occur through two specific avenues.

1. The first scenario could only unfold if a citizen were to sell their Metamask password received during registration, along with all their personal data (ID number, mobile, etc.), an action that would likely be vehemently rejected by any vigilant citizen. Furthermore, on the day of voting, both the citizen and the potential 'attacker' would need to maintain constant communication for the timely transmission of the unique code enabling the vote. This process makes it exceedingly challenging to execute successfully.
2. The second case involves the physical presence of an individual next to the citizen called to vote. This physical presence allows the accompanying person to manipulate the voter, as there is no provision for monitoring the voter to ensure they are alone during the voting process.

Vote Anonymity Threat: Our system satisfies the principle of election integrity through the association of citizens with Metamask addresses and the encryption-based control of personal data. The use of the SHA256 cryptographic algorithm is ideal for our case, given its resistance to decryption. Thus, all personal data remains secure, allowing for verification through the matching of generated hashes. Simultaneously, the system employs a unique function storing all the encrypted votes, without the origin of each vote. Consequently, tracing the source of each vote, combined with hash decryption, becomes impossible, meeting the principle of voting confidentiality.

Even in the retrieval of receipts from the blockchain, facilitated by each transaction originating from a specific address and the encryption of personal data, identifying the sender's details for each vote remains an insurmountable challenge. The robust security measures, including address-specific transaction origins and data encryption, ensure that the privacy and anonymity of the voters are rigorously maintained throughout the entire process.

Vote Validity Threat. This principle is satisfied by the implemented system. Through multiple checks in both the user interface and our smart contract, the submission of an invalid vote is prevented. To store a vote, a unique code must be matched with the Metamask address, indicating the successful completion of the verification stage. Otherwise, vote storage is not possible. Additionally, due to the electronic nature of the voting system and the necessity to select a candidate from a predetermined list to cast a vote, storing an 'invalid' vote is impossible. Furthermore, the existence of a list of valid voters helps avoid numerous errors that could occur during the voting or vote counting process. By combining all the methods employed by the implemented system, storing an invalid vote (a vote that should not be counted) is rendered impossible.

Through electronic voting, this principle can be maintained, alleviating many problems associated with traditional voting methods. Controls for storing a vote can easily be modified by any system to align with its specific needs.

5.2 Potential Threats and Resistance to Attacks

Attacks in Blockchain Network: Like any other technology, Blockchain has its drawbacks despite its transparent and immutable digital ledger. Various types of security threats make Blockchain networks vulnerable.

51% Attack

A successful 51% attack would be catastrophic for our system, as it could alter the results of elections or impede the entire electoral process. However, such an attack is challenging and demanding on a network with a significant participation rate, accompanied by substantial costs.

Beyond the cost, a group attempting to attack the network through a 51% attack must not only control 51% of the network nodes but also introduce the modified blocks into the blockchain in a very expensive timeframe. Even if they hold 51% of the network's hashing power, they may be unable to keep up with the block creation rate or introduce their chain before valid new blocks are created by the real blockchain network.

After Ethereum's transition to proof-of-stake, a 51% attack on the Ethereum blockchain became even more expensive. To execute such an attack, a user or group would need to possess 51% of the staked ETH in the network.

Protection against this type of attack is one of the main reasons for choosing the Ethereum network to implement the voting system. Large cryptocurrencies are unlikely to suffer from 51% attacks due to the prohibitive cost of acquiring such significant hashing power and the fact that they make it impossible to

introduce a modified blockchain. Therefore, the 51% attack threat primarily affects cryptocurrencies with lower participation and hashing power.

DOS Attack
Due to its digital nature, the blockchain is susceptible to attacks and exploitation. DOS and DDoS attacks on a blockchain focus on the protocol level, with the most significant threat being transaction flooding. Traditional DDoS attacks can be executed against a blockchain to slow down its operations.

In the case of a DDoS attack, some nodes may be temporarily disabled. The distributed nature of the blockchain network ensures that transactions can continue even if some nodes go offline. However, this does not imply that blockchain networks are fully resistant to DDoS attacks.

DDoS attacks are considered "weapons of mass destruction" on the Internet. It is more challenging to defend against these attacks, and currently, there are no precautions that any organization can apply to be 100% secure. The greater the computational power, the higher the chances of facing a blockchain DDoS attack. This is another reason for choosing the Ethereum network to implement the voting system, as it offers a broad range of security against such massive DDoS attacks.

Sybil Attack
In a voting system, a Sybil attack can be employed to influence the outcome of the vote. The attacker can use multiple identities to cast multiple votes, thereby affecting the voting results. Additionally, the attacker may spread false votes in the system, causing confusion and uncertainty among genuine voters.

The system we have created is entirely resistant to any form of Sybil attack. This is possible due to the multifactor authentication methodology required for the voting process. With a list of valid accounts (voters) and the necessary authentication steps for storing a vote, attempting to cast multiple votes from one account is not feasible. Moreover, the use of the Ethereum network prevents a significant increase in blockchain traffic, preventing Sybil attacks on a node.

By taking appropriate measures to create a list of valid voters and continuously monitoring the smart contract each time someone interacts with it, in combination with the use of Ethereum and corresponding protocols, the system implemented in this thesis is considered absolutely secure against Sybil attacks. Unauthorized users are not allowed to interact with the smart contract, ensuring the system's security.

Operational Attacks on the Voting Process: We will analyze the security of the system against attacks that could harm its operational processes.

MITM Attack
Information gathered during an attack could be used for various purposes, including identity theft or voter manipulation.

There are three types of attacks:

- IP Spoofing: An attacker disguises themselves as an application by changing the packet headers to an IP address. Users attempting to access a URL may be redirected to the attacker's website.
- ARP Spoofing: The attacker's MAC address is linked to a user's IP address on a local network. This causes all messages to be transmitted to the attacker.
- DNS Spoofing: An attack on a DNS server with the goal of changing the addresses of a website. Users trying to enter a website might be directed to the attacker's site.

To successfully execute a Man-in-the-Middle (MITM) attack, the attacker needs to decrypt the received data. This decryption might involve various techniques, such as SSL stripping, downgrading an HTTPS connection to HTTP, bypassing TLS identity checks sent from the application to the user.

To prevent such attacks on our system, several measures should be taken. The most crucial is the establishment of a secure connection. Besides end-to-end encryption, policies like HTTP, HSTS could be configured to enforce the use of SSL/TLS security on multiple subdomains. This enhances the website and web application's security against protocol downgrade attacks and cookie tampering attempts.

Due to the use of Metamask and the security methods it provides, even in the scenario of an attacker conducting ARP spoofing and downgrading to HTTP, the server would not function from the user's side, as SSL encryption is required for the proper operation of Metamask.

Moreover, citizens should be educated by authorities during their registration in the system to protect themselves from phishing emails that might lead them to fraudulent websites or prompt them to download malicious code to their devices.

Exploitation of Smart Contracts
In smart contract exploitation attacks, an attacker takes advantage of vulnerabilities or weaknesses in the contract code. Here are some common types of smart contract exploitation attacks, along with how they are mitigated in our system:

1. **Reentrancy Attacks**: This attack occurs when a smart contract is repeatedly called by another contract during a transaction, with the goal of extracting resources before the original transaction completes.
   In our system, sensitive personal data variables and functions are defined as private. Therefore, they cannot be called by other smart contracts. Additionally, public functions include identity checks, preventing unwanted addresses from interacting.
2. **Front-running Attacks**: Front-running attacks happen when an individual anticipates the execution of a smart contract and executes it first, before the original transaction is completed. This can lead to capital loss or malfunction of the contract.
   Our system is resistant to front-running attacks due to identity verification measures. To vote, users must be authenticated and linked to a One-Time

Password (OTP). If someone attempts to vote before the initial voting, their vote won't be confirmed without prior identity verification and OTP assignment.

3. **Timestamp Attacks**: Timestamp attacks occur when an attacker creates a transaction and changes its timestamp to achieve a desired state in the contract affected by the transaction.
   The system is protected against timestamp attacks through timestamp checks implemented in the smart contract. Authorities also collect votes at the exact moment the voting deadline expires, preventing manipulation of results through altered timestamps.
4. **Variable Overflow/Underflow Attacks**: these attacks occur when the prices of a variable in a smart contract exceed their range limits, causing a variable in a smart contract to go out of range, causing unpredictable behavior of the contract and possibly loss of capital. In order to enhance the robustness of the smart contract and minimize the risks associated with variable overflow and underflow, we opted for the integration of mappings alongside the dedicated vote storage table. This strategic choice provides a secure and automated mechanism within the Ethereum environment, effectively managing data storage and mitigating potential issues related to exceeding predefined range limits.

6 Discussion and Conclusion

The implemented system has some weaknesses, the resolution of which could make it ideal and ready for use even at the level of national elections.

The first weakness that needs to be resolved concerns the sending of the one-time password via SMS. It is not possible to send SMS, but also in general to call APIs from the smart contract. Furthermore, due to the storage of a substantial volume of votes, the necessity to retrieve them in batches can result in considerable time consumption during the vote tallying process, potentially leading to delays.

Oracles could be a solution to these problems, so we could call APIs from the smart contract, and handle SMS or API communications. Oracles are third-party services that act as intermediaries between smart contracts and external systems. They can provide smart contracts with access to external data and services, including the ability to make HTTP requests. Moreover, enhancing more suitable cryptographic algorithm to align with the specific security requirements of voting could further enhance the safety, integrity, and the speed of receiving the voting data and results.

The implemented system serves as a fully integrated and functional model, paving the way for the successful development and deployment of voting systems for a wide range of applications. By implementing an innovative user identification system that effectively fulfills the requirements of voting without endangering personal data, and coupled with robust vote encryption, the system could be deemed deployment-ready. It particularly suits applications like university

elections, which require fewer resources. Additionally, its adaptable technology and generic framework make it a solid foundation for building even more secure systems in the future.

With the continuous advancements in Solidity and blockchain technologies, the prospect of an even more enhanced e-voting application is on the horizon. Nevertheless, through the encryption of API keys and the exclusive handling of messages by the smart contract, the proposed application can be considered secure and suitable for use in national elections, providing a robust foundation for future improvements in the field.

References

1. Garg, K., Saraswat, P., Bisht, S., Aggarwal, S., Krishna Kothuri, S., Gupta, S.: A comparitive analysis on e-voting system using blockchain. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (2019)
2. Ehin, P., Solvak, M., Willemson, J., Vinkel, P.: Internet voting in Estonia 2005–2019: evidence from eleven elections. Gov. Inf. Q. **39**(4), 101718 (2022)
3. Baltic, T.: Estonian electronic ID - card application specification prerequisites to the smart card differentiation to previous version of EstEID card application (2013)
4. Ayed, A.B.: A conceptual secure blockchain-based electronic voting system. Int. J. Netw. Secur. Appl. **9**(3), 1–9 (2017)
5. Khoury, D., et al.: Decentralized voting platform based on ethereum blockchain. In: 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET). IEEE (2018)
6. Anjan, S., Sequeira, J.P.: Blockchain based E-voting system for India using UIDAI's Aadhaar. J. Comput. Sci. Eng. Softw. Test. **5**(3), 26–32 (2019)
7. Roopak, T.M., Sumathi, R.: Electronic voting based on virtual ID of Aadhar using blockchain technology. In: 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE (2020)
8. Abayomi-Zannu, T.P., Odun-Ayo, I.A., Barka, T.F.: A proposed mobile voting framework utilizing blockchain technology and multi-factor authentication. In: Journal of Physics: Conference Series, vol. 1378, no. 3. IOP Publishing (2019)
9. Anwar ul Hassan, C., et al.: A liquid democracy enabled blockchain-based electronic voting system. Sci. Program. **2022**, 1383007 (2022)
10. Pramulia, D., Anggorojati, B.: Implementation and evaluation of blockchain based e-voting system with Ethereum and Metamask. In: 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). IEEE (2020)
11. Vairam, T., Sarathambekai, S., Balaji, R.: Blockchain based voting system in local network. In: 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1. IEEE (2021)
12. Ochoa, X., Peláez, E.: Affordable and secure electronic voting for university elections: the SAVE case study. In: 2017 Fourth International Conference on eDemocracy & eGovernment (ICEDEG). IEEE (2017)
13. Ahn, B.: Implementation and early adoption of an ethereum-based electronic voting system for the prevention of fraudulent voting. Sustainability **14**(5), 2917 (2022)