Journal of Digital Science

ISSN 2686-8296

Volume 1 Issue 1 December 2019

CONTENTS

Secure i-Voting Scheme with Blockchain Technology and Blind Signature

Mahmoud Al-Rawy[1] and Atilla Elci [2]

[1] Ark IT, Tirana, Albania
[2] Electrical Electronics Department, Aksaray University, Aksaray, Turkey

https://doi.org/10.33847/2686-8296.1.1_1

Received 25.09.2019/Revised 30.10.2019/Accepted 11.12.2019/Published 22.12.2019

Abstract. In the last four years, blockchain technology affected largely all aspects of our lives. Blockchain started to launch a new technological revolution of storing digital transactions over the Internet, verifying the authenticity, licensing and providing the highest degree of security and encryption. Blockchain usage started with digital currency then its implementation extended to many industries such as voting, health records, copywriters, real estates and so on. However, it is time to upgrade the election scenario from practicing paper-based elections to use modern technologies in order to facilitate our lives. The fact that the blockchain technology has demonstrated almost infinite immutability and high resistance against hacking, lends credit to employ it in securing election data from fraud by saving every single piece of data, record or transaction with unchangeable history. In this paper, we propose and test implement a robust online voting system based on blockchain in order to prevent election forgery and ease the voting process for citizens. The essence of our research lies in abandoning alterable traditional databases and replacing them with two private blockchains that use the peer-to-peer network. Along with the blockchains, we utilized blind signature to maintain vote/voter privacy in order to safeguard voter eligibility validation against manipulation and forgery. Lastly, we discuss a threat model, and suggest solutions overcome it; we also suggest a solution to identity impersonation and vote-selling problems.

Keywords: Blockchain, Internet Voting, Vote/Voter privacy, Blind Signatures, Public-Private Key algorithm anonymization.

1. Introduction

For hundreds of years, traditional elections based on the principle of accepting a ballot paper at a specific polling station have been practiced. The cost of this process, of each ballot paper, trustee, preparation of a polling station, and other high costs not to mention the time spent, difficulties faced by disabled people, repeatedly encountered problems due to fraud, manipulation of results and influencing voters are all factors to consider against traditional/paper-based elections. History is littered with examples of elections being manipulated to influence their outcome; scammers and rulers have developed means of manipulating votes to achieve personal agendas, which is considered a violation of the core principle of democracy.

With the advent of the ever-expanding Internet, many researchers have devoted effort to find the easiest, economical and most importantly a secure way to achieve a fair online election by using an i-Voting system aimed to overcome all problems faced by the traditional elections. The i-Voting system, alternately called the Internet Voting, may be defined as an election system constructed on cryptographic techniques allowing voters to cast their ballots for their favorite candidates and transmitting their votes over the Internet from anywhere in the World while the voting and tally processes run entirely anonymously. In 1981, David Chaum introduced the first electronic voting system [25], where he used public-key cryptography to maintain solid anonymity of voters/votes as well as utilizing Blind

Signature to ensure disconnectivity between voters and their ballots. Since then, the evolving of cryptography inspired several academicians to show interests in Internet voting [20, 21, 26 – 29]. Yet, due to the fact that Internet voting systems run over the Internet, significant challenges such as voters' authenticity and eligibility, ballot privacy, process completion, the immutability of the results and fairness have been obstacles for decades.

Despite the concerns mentioned above, Estonia introduced for the first time in the history online voting system to be the first country in the world to put in place an Internet voting. In the most recent online elections, over 30% of the Estonian participators cast their ballots online [1, 5]. Similarly, over 280,000 votes were submitted through iVote in New South Wales, and 70,090 Norwegian votes submitted online in 2013. Security vulnerabilities in voters' client devices, systems servers, and voter authentication process have been demonstrated by security analysis on these systems [12, 13].

In 2008, Nakamoto introduced his by-now famous white paper [3], which revolutionized the world of cybersecurity. He combined some encryption algorithms in a brilliant way to obtain immutability of data and provide sufficient protection to transfer and store the data in a distributed ledger with the well-known technology called "Blockchain." Blockchain technology, initially known as the cryptocurrency transaction log, helps keep data resistant to tampering with ever-growing linked data ledgers and allows secure exchange transactions of valuables such as votes, funds, stocks or data access rights.

Achieving fair election without running the risk of rigging and manipulation of the results was and still difficult [11]. However, with the emergence of the blockchain which provided many possibilities that inspired researchers to investigate and reach to the proved conviction through their research to suggested that blockchain is a suitable base for Internet voting, besides, it could have the prospect to make e-voting more acceptable and reliable in the society [4].

Fundamentally, blockchain uses data distribution principle (i.e., distributed replication or decentralization). Therefore, it operates as an electronic transaction log-cum-record system that allows all parties to track information through a secure network without requiring verification by a central authority.

Eventually, the most important advantages of using blockchain-based i-Voting has become the following:

i) Anonymity
ii) Accuracy
iii) High performance, particularly against Denial of Service (DoS) and Distributed DoS Attacks
iv) Strong integrity (immutability) by replicating many copies of the same identical data over many nodes, but none is the golden copy.

It is fundamental that a proposed e-voting scheme contributes to preventing any violation of voter-ballot anonymity, and the absence of any possibility to manipulate the results. The main challenges usually i-Voting systems have faced are highlighted below:

Vote Privacy. Voter's choices must be anonymous (secret ballot). Non-observance of the secrecy of the ballot leads to attempts to influence the voter by either intimidation or through potential vote-buying.

Identity Theft (ineligibility). Impersonation and multiple vote casting were real issues in the past with traditional-based voting systems. Deceptive voters used to register themselves multiple times or manipulate their eligibility of voting. In Australia, 217 ineligible voters cast votes in 1996 elections, and in 2010 elections, a family cast more than 150 votes by impersonating others. The security analysis of the Estonian Internet Voting System demonstrated that attackers can plant malware in the voter's client and read the National-ID card's PIN, then impersonate an eligible voter to cast an unqualified vote as they desire. In fact, due to the lack of biometric devices (online) electronic voting machines can allow identity theft. A biometric device would not be available in every home on the polling day, and, supplying such equipment for each house is very expensive and unpractical. Therefore, there must be a solution to overcome this issue.

Immutability. The biggest election concerns lie in the immunity of the electoral system from manipulation and counterfeiting [17].

In this paper, we first illustrate the problem and then propose the solution approach. In section two, we will compare traditional databases and blockchain. In section three and four, we will describe our scheme in further detail in term of authentication and voting processes.

2. Problem and Solution Approach

In online voting systems, privacy must be preserved to dispel the election concerns highlighted above. This paper proposes a novel i-Voting architecture to properly facilitate digitalized elections maintaining vote and voter privacy while preventing fraud through employing firstly the blockchain technology to ensure result immutability, secondly, through the RSA public-private key algorithm (PKI) [2] to prevent identity theft, lastly, we will be using the Blind Signature algorithm to allow only eligible voters to cast a ballot.

The proposed voting scheme uses the blockchain technology to store the cast votes and electronic IDs, thus it can act as an immutable database. As usual, the current Web-based system uses HyperText Transfer Protocol Secure (HTTPS) yet there is the need for a further configuration affected by the system administrators to secure the connection between servers and client’s computer, therefore preventing devastating attacks such as Logjam and FREAK attacks. Disabling the support for TLS (Transport Layer Security) export cipher suites and using a 2048-bit Diffie-Hellman group also disable other cipher suites that are known to be insecure, thus enable forward secrecy to obtain the desired server immunity [14, 15].

Moreover, it is necessary to relinquish the use of third-party servers that used to read and verify votes by voters. Studies proved that not doing so opens the door to different opportunities for privacy violations as in the cases of elections in Australia and some other countries. So, any kind of reading the vote or overriding it in our scheme is prevented just as in traditional paper-based voting [16].

The major challenge in this work is allowing people to cast their votes right from their home, without the need of going to the polling stations. Usually, digital voting systems rely on the use of standalone electronic voting machines (EVM) which also perform user (/voter) verification as well as the entire election process. In this proposal, we give up on needing such machines and allow the voters to use just an Internet browser to cast ballots and votes.

Our design of the blockchain-based digital voting system is explained below in terms of the processes involved, such as, voter logging in, digital ID creation, re-logging, Electoral Authority preparation, vote casting, and vote tallying; but first we introduce the differences between traditional databases and blockchain to better highlight the advantages of the latter.

3. Structural Differences between Traditional Database and Blockchain

In this section, we will review the main reasons behind adopting blockchain over the traditional database in our voting system architecture. In fact, a blockchain is a form of a distributed database, both used for data storage, but there are many fundamental differences in their structures. The traditional database has a central authority (Administrator) to maintain, control, distribute read/write privileges to other authorities, also, it often uses the client-server network. Blockchain is a digital ledger that receives, encrypts, distributes and stores data without requiring a trusted third party (i.e. Disintermediation), such as Bitcoin, considering ledger crypto-currency. Blockchain storage method produces a linear series of blocks, where each is connected to the previous one, that provides high data immutability and confidentiality. Let us consider the superior features of blockchain with respect to serving as a base for data.

3.1. Immutability of Blockchain

In contrast to centralized databases, data stored in the blockchain is not erasable and almost impossible to be modified, unlike the traditional databases which perform Create, Read, Update and Delete operations (C.R.U.D. Operations) according to the given user permission by the administrator. A blockchain is a secure, distributed, and immutable database shared by all parties (nodes) in a distributed network. A blockchain block contains transactions stored and linked to each other through the so-called Merkle root, for example, see Fig. 1; and, each block is connected to the previous block’s "hash" to form an interconnected chain.

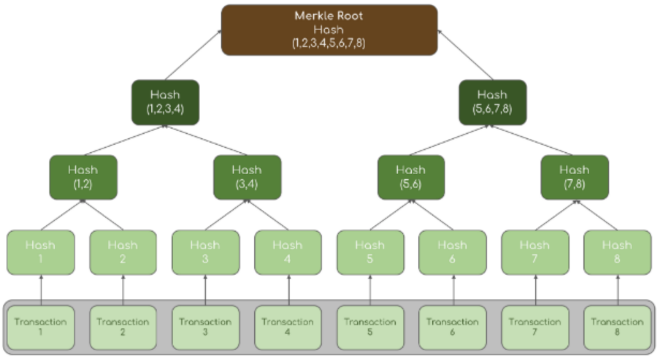

Fig. 1. Merkle root tree example.

A Merkle root is the hash of all the hashes of all the transactions contained in a block; it provides the ultimate encryption for the transactions tree included in a block. This means that any change in any transaction within the block leads to not only a change in the Merkle root hash but so too the block hash will change, thus breaking the chain; as the result, the immutability provided by the blockchain is almost infinite.

1. **Performance**

Traditional databases are very fast compared to the blockchain for the latter uses the cryptography to link its blocks, also employs consensus principle that provides full-distribution by allowing a majority of peers agree on the outcome of transactions in order to accept it into the chain. Every node (aka, computer) in a blockchain network contains a copy of the full data ledger, so any node experiencing failure will be shut out and the network completes its work with the remaining nodes. In the case of distributed databases, the trust factor between the parties must be significantly guaranteed to preserve the integrity of data. Blockchain is quite the opposite operationally; it repeals the trust factor to allow an independent transfer of transactions, even though transacting parties are anonymous. In conclusion, traditional databases are fast but not fully distributed. As for permissionless blockchains that use the principle of consensus, which makes it slow; however, permissioned blockchains are relatively fast, while not exactly fully-distributed.

2. **Decentralization Aspect to Secure Data:**

Instead of using a single repository of data to upload to a server, blockchain distributes data across the entire network to be stored in every node’s digital ledger.

Thus, data loss is almost reduced to zero, because if one or more nodes go down, the data will not be affected, unless the entire node network fails, and this is of an extremely low probability. This in its entirety, cutting out the middle-man, the central authority and need to trust a third-party in processing data create a distributed immutable ledger of data records, as shown in Fig.2.

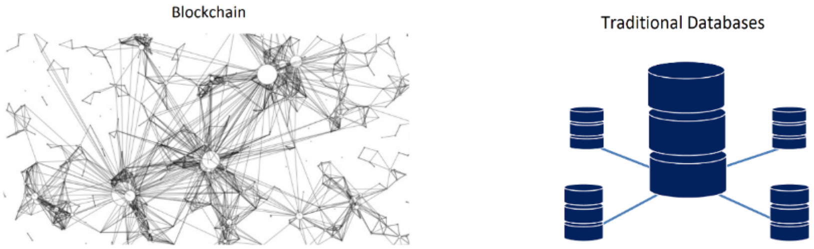

Fig. 2. Blockchain structure vs traditional database structure.

4. Authentication Structure of the i-Voting System

In this section, we will propose a voter/user authentication process applied before casting a ballot in i-Voting System. The user authentication process requires that voters must establish and sign an electronic identity after secure logging into the system, as described below.

1. The Access Authorization

Let us assume that the country where a digital voting based election will be held has its own e-government system where every single citizen is recorded and has his/her own individual private e-number. Therefore, it is possible to make the very first voter's authentication stage via TLS client authentication [19] starting with asking the voter full name (F-Name), National ID number (NIDN), and e-number. Subsequently, a session token is computed as follows which will be used to identify that user then on:

$$\mathbf{Hash} = \{\mathrm{eNumber} + (\mathrm{F} - \mathrm{Name}) + \mathrm{NIDN}\}_{\mathrm{Enc}} \qquad (1)$$

Web applications that use cookies are known to be vulnerable to XSS (Cross-Site Scripting) Attacks [18], so voter's privacy might be violated by unauthorized accesses; thus, we strongly recommend cutting out the use cookies and replace it with session tokens. Additionally, using HTTPS protocol is insufficient for an attacker requesting a page with HTTP gets served user cookies in the unconcealed form. Overcoming this gap requires sending system pages only in response to HTTPS calls before starting the session.

2. Digital ID Card Creation

As the voters would identify themselves in the login step, they must have first created an electronic identification (eID), consisting of an ID and digital signature. "eID" is an electronic identification solution of citizen [7]. The idea behind creating a digital ID card is to prove voter identity while using it in the voting processes and to be verified by Electoral Authority's Representative (EAR, more on this below).

In the public-private key algorithm, the user needs a set of two complementary keys: one of them is published for use by the public, and the other is kept secret by the individual for personal use. If a piece of data is encrypted by using the public key, only the person who has the second prime of the key (private key) will be able to decode it. On the other hand, if the person uses his private key to encrypt a piece of data, anyone who has the public key can decrypt it. This process also makes sure the signer (encryptor) is the owner of that key pair. Usage of the public-private key algorithm provides high protection of the data from being modified as well as authenticating the private key owner.

After a voter logs into the system, a public-private key pair will be automatically generated. The public key will be stored in the system, open to the public. The second key (a private one) will be conveyed to the user, kept hidden and can be used only by the user. Indeed, the private key must not fall into the hands of others, because it might be used to impersonate the voter or used for signature purposes elsewhere. After the creation of the key pair, some information about the person will be requested to be matched with the stored information in the system's database, in order to verify the voter's credibility. The requested information consists of the following fields:

1. Voter's National Identification Number and its expiration date (NIDED).
2. Images of the voter, captured right from the system using the device's webcam: one of the voter himself/herself and two others of both sides of the national identification card (IMGs).
3. Voter's full-name and phone number (V-NO).

The voter's personal identification card number is requested for the second time for confirmation purpose. In order not to depend on the availability of usable biometric devices (fingerprint, eye print), it is necessary to invoke another solution, such as these required images can be photographed, encrypted and stored in the database to prove that the voter has created own electronic identity.

Voter's phone number will be used in processing the time-based one-time password. It will be used while logging into the system, also in every process after that including casting a vote in order to verify the voter's identity. Finally, all the requested data fields are combined and encrypted with the voter's public key to obtain a verifiable eID hash as follows:

$$\mathbf{eID} = \{(\mathrm{F}-\mathrm{Name}) + \mathrm{NIDN} + \mathrm{NIDED} + \mathrm{IMGs} + (\mathrm{V}-\mathrm{NO})\}_{\text{Voter's Public Key}} \quad (2)$$

Eventually, the voter will create a unique signature by using his private key, allowing others to check the validity of the signature using his public key for the purpose of verification [8]. The voter, in order to prove ownership of the private key without requiring to reveal it, will use the signature to sign the ballot.

$$\mathbf{Signature} = (\mathrm{eID})_{\text{Voter's Private Key}} \quad (3)$$

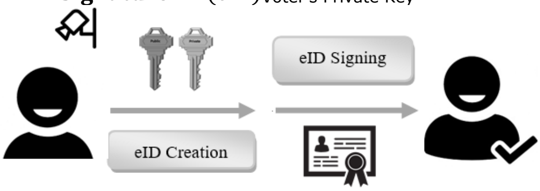

Fig. 3. Voter authentication structure.

3. Re-Login

It should be possible to keep the election system running live and open to the public some number of days before the election in order to allow voters to login and establish their eIDs. In this way, the EAR will have enough time to correct voter data if there will be any mistakes. Meanwhile, voters will get familiar with the election system and check whether someone else used their data to create fake eID; if so, they can inform the EAR to receive the necessary assistance. Even after establishing the eID, attackers who hack the eID of an eligible voter and change the phone number to redirect the privileges to his favor will be exposed. The moment that voters try to log into the system for the second time, their eID will be corrupted, due to the fact that his data has been changed, thus they can easily report to EAR immediately. As the voter successfully completes establishing his/her eID, he can log out of the system, and re-login during the polling day in order to vote.

Protecting voter data from being used by others must be given serious consideration; time-based one-time password algorithm [9] may be used to preclude this issue. In the time-based passwords, time synchronization is very important, a secret key and a timestamp will be defined, and everything will be synchronized via a standard protocol such as network time protocol. Once the password token will be created, it can be used in a limited duration just to ensure that the voter is using his own data in the login process, or the ongoing operation is within his knowledge because he is the one who receives the verification code on his phone.

Once a voter logs into the system (for the second time) in order to vote, the system will check whether his electronic identity has been successfully established or not. In case that everything has been done in proper order, the voter will receive a time-based verification code on his phone. The code could last for, say, two minutes. Also, the voter must confirm his identity using his own private key.

4. Summary

So far, we have described the use of the eID and electronic signature instead of relying on the database management system's self-created IDs to increase the e-voting system's security level. Now, let us summarize our accomplishments of this approach in this section as follows:

Obligating the voters to capture photos of themselves and their ID card is part of the system's processes. These images allow the EAR to verify whether a voter has established own eID or there are impersonations, especially impersonating a deceased person. Those images will be converted to binary format, encrypted and stored in the database. In addition, it will be part of the eID hash, so it will be impossible to decrypt, change or manipulate images.

- Nowadays almost every individual has his own cell phone, which gives us the possibility to use it to secure the voters' data from being exposed. Otherwise, supposing that we did not use the time-based verification code method, and someone's private key got exposed anyone who has the private key and the individual's information can re-login into the system and cast a vote as he desires. That is why the verification code is used to add another level of protection. Even though an individual's information and the private key could fall into other's hand, they will not be able to open the system unless the verification code can be received in time.

5. Voting, Verification and Tallying Processes

Basically, in any election employing 'secret ballot-open counting', the following points must be considered: (1) Vote by secret ballot: a vote's originator must be unknown. More clearly, the voter's ballot shall not be violated by anyone anyhow, which means it must be completely concealed whether it is directed to a candidate or it is null. (2) The voters must be marked in the electoral registry that they voted to prevent duplicate voting or cheating and to allow only the legitimate voters to cast a ballot. (3) Open counting: the outcome of the election should be determined by an open verifiable counting of the votes.

Our design is unlike the Bitcoin [19], which uses a single public blockchain; the proposed scheme uses two private/permissioned blockchains. The reason behind preferring permissioned blockchains is twofold: firstly, it prevents unauthorized nodes/parties from joining the network, which in return prevents Sybel Attack; secondly, it provides significant resistance against today's security problems by using robust cryptography features and limited access to the ledger, without affecting the transparency aspect of blockchain technology.

I-Voting Scheme employs two blockchains. Let us call the first blockchain as BL-v, the voters' identities and the second one as BL-b, the encrypted ballots. The purpose of using permissioned blockchain is to limit the parties who can read the information, also, restricting the nodes and separating it in different locations around the country. Now let us dive into the vote casting process as the following:

- **Addresses.** Every transaction requires candidates' new addresses in order to keep the vote-voter anonymity every single time voter cast a vote [10]. The recipient to receive the ballot credit will specify the generated address.

Sending a checksum with an address permits to verify that address was not manipulated by preventing any possibility of a Men-In-The-Middle Attack via making the communications between voter and candidate going through a secure channel using a handshake protocol (see Fig. 4).

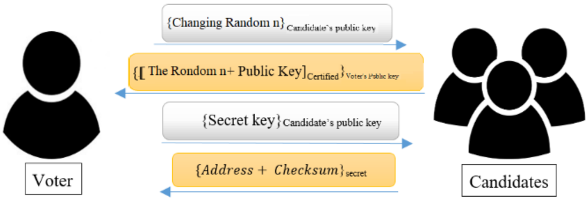
{Changing Random n} Candidate`s public key
{[The Rondom n+ Public Key]Certified } Voter's Public key
{Secret key} Candidate's public key
{Address + Checksum} secret
Voter
Candidates

Fig. 4. Voter-candidate authenticated communication.

- **Transactions and Blocks**. Each voter has the right to cast only one countable vote. Once the voting starts, voters have to sign their ballots from the EAR anonymously using the blind signature in order to verify their eligibility. The blind signature scheme is described as follows:

EAR owns a signing function S'_{EA}. The corresponding publically-known inverse S_{EA} satisfies $S_{EA}(S'_{EA}(s)) = s$, but gives no clue about S'_{EA}. To obtain EAR's signature of the transaction (t) without revealing it, voter will depend on a computing function C_{Voter} and its inverse C'_{Voter}, both of which belong to him/her only, and satisfy the condition that $C'_{Voter}(S'_{EA}(C_{Voter}(s))) = S'_{EA}(s)$ while C_{Voter} and S'_{EA} give no clue about (s). The signing scheme is presented as follows:

1. Voter sends $C_{Voter}(t)$ to the EAR.
2. EAR receives $C_{Voter}(t)$, checks the eligibility and signs it using S'_{EA} to obtain $S'_{EA}(C_{Voter}(t))$, then sends S'_{EA} $(C_{Voter}(t))$ back to the voter.
3. The voter uses C'_{Voter} to obtain S'_{EA} (t) according to $C'_{Voter}(S'_{EA}(C_{Voter}(t))) = S'_{EA}(t)$.

The steps above demonstrate how voters could obtain S'_{EA} for the transaction ID (TxID), without revealing the transaction. After a voter signs the ballot, the voter will specify a candidate/party and cast the vote. Every voter has one value credit (Signed Ballot) to be spent once by giving it to a specific candidate. Now, BL-v transaction containing the fields shown in Table 1 is created and submitted to the transaction pool.

Table 1. BL-v transaction content.

Block element	Dummy example	Description
TxID	f4184fc596403b9e17e450rd63	Formed by encrypting the transaction data twice with the SHA-256 algorithm
Version	1.0	Block version
Size	8 bytes	Block Size
Timestamp	1565259135	Epoch Unix Time Stamp
Prev-Out	J9f74g4h4566f8kl9h985025r6	Presents whether the voter has voted before or not
Out	{Recipient Address} Recipient Public Key	Specified candidate's address
S_{EA}	[Transaction Signature] EAR	Contains the S'_{EA}(TxID)

The BL-v transaction is created by including Transaction version, Transaction size, Prev-out, and the Out. The Out contains the recipient wallet address encrypted with the recipient's public key. However, the recipient will prove that he is the owner of the private key; only then, he gets the vote. After all, TxID established by including all the transaction stuff together, encrypted with SHA-256 twice.

As for the BL-i transaction, it will include (Prev-out, and the Out), as shown in Table 1. The Out contains the recipient wallet address encrypted with voter's public key to conceal it.

Table 2. BL-i transaction content.

Block element	Dummy example	Description
TxID	8h5d8i905r4de34y7i8v4z33c5	Formed by encrypting the transaction data twice with the SHA-256 algorithm
Version	1.0	Block version
Size	8 bytes	Block Size
Timestamp	1565259135	Epoch Unix Time Stamp
Prev-Out	5tf43w6i80hf56831s3vs3st67	Presents whether the voter has voted before or not
Out	{Recipient Address} Voter's Public Key	Specified candidate's address
S_{EA}	[Transaction Signature] Voter's Private Key	Contains the S'EA(TxID)

Eventually, the transactions will be posted with the one-value credit to the transaction pool waiting to be validated and appended to the blocks. Let us summarize this stage as follows:

1. A voter's first cast transaction will include no Prev-Out of the BL-i transaction and false value in Prev-Out of the BL-v transaction. After that, if there is an overwriting vote, it will consist of the previous TxID in the BL-i transaction as well as in BL-v transaction block.
2. BL-v does not have any clue leading to the voter.
3. BL-v must include only signed transaction by EAR.

- **Vote Tallying**. Vote tallying will be one of the responsibilities of the EAR for that constituency. EARs are (virtual agent) persons or institutions permissioned by the network and directed by the state to perform audit and certification of the votes/voters. An EAR could be assigned for every polling station or even one per district.

Candidates/parties will get a summary ballot tallying just the votes received; furthermore, the summary will be fully vetted by the EARs. That is how the vote tallying will happen automatically. EAR will verify both of the blockchains, also counting or listing all voters thus verifying who did cast his vote and who did not.

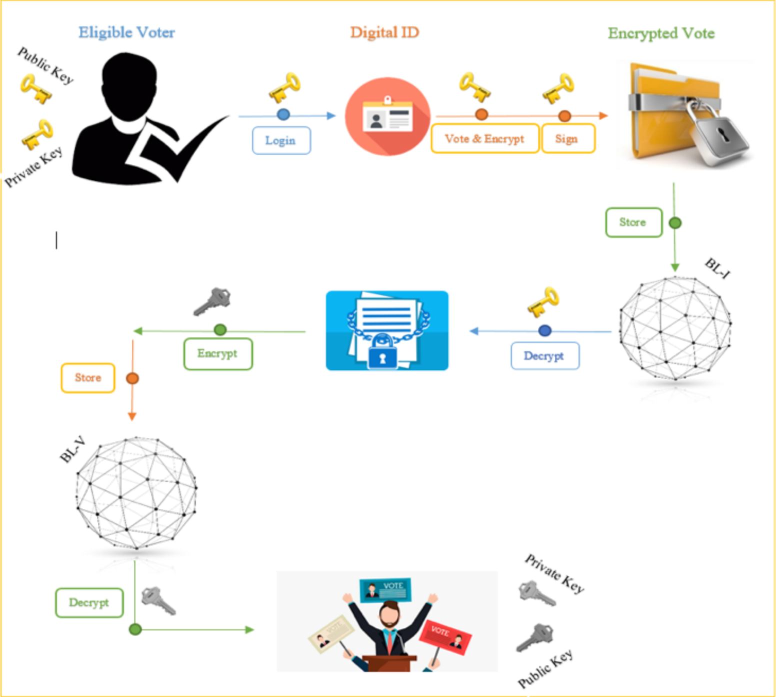

Fig. 5. Secure i-Voting architecture.

6. Threat Model

Maintaining sufficient security of the e-voting paradigm requires overcoming many difficult problems in cybersecurity, especially, with existing technology, a small lapse can affect the election result's integrity. Undesirable experiences in different places around the World (New South Wales, Estonia, India, etc.) in real-world have come up in online voting examples which demonstrated many security issues. There are many kinds of attackers (foreign countries, deceptive voters, funded criminals, etc.) who demonstrated realistic threats; many of these threats must be taken as an example while designing or setting online voting. Let us identify some potential attacks and any exposures that might exist to highlight the way to avoid and overcome them.

1. Cross-Site Scripting (XSS)

An XSS vulnerability is a type of injection, simply attacker who can exploit an XSS attack could gain the ability to act like the victim. Moreover, both the voter and the vulnerable system often will not be aware of the attack. Using the well-known practices below all together considered as a great way to defeat the majority of XSS vulnerabilities [23]. Let us outline these methods below:

Escaping. Escaping data means ensuring that the data is secure before rendering it to the end-user. So escaping user input and key characters prevent received data from being interpreted in any malicious way.

Validating Input. Input validation prevents voters from inserting any special characters, instead of rejecting the request.

Sanitizing. Assuring the input data will not do any harm to users and database by cleaning the data from any potentially harmful markup, moreover, changing untrusted user inputs to an acceptable format.

2. FREAK Attack

The SSL/TLS vulnerability (Factoring Attack on RSA-Export Keys) may allow attackers to decrypt secure communications between vulnerable clients and servers (intercept HTTPS connections) and force them to use older and weaker encryption, also known as the export-grade key or 512-bit RSA keys [24]. Let us highlight the necessary precautions against the FREAK attack by:

- Disable the support for TLS export cipher suites.
- Disable the support for insecure known ciphers (not only RSA export ciphers),
- Disable support for ciphers with 40- and 56-bit encryption,
- Enable forward secrecy.

3. Malware

Malicious attackers who tend to manipulate systems to have more access by promoting their privileges can install Trojans or backdoors by using pre-existing botnets and target a specific country or region [22]. In this case, it would be easy for them to fully control data of the infected voters' computers.

Voters must be very careful and familiar with the computer protection instructions to be able to protect their own computer (or smart terminal device such as a cellphone, tablet, so on). Some of the common critical steps to protecting voters' computers are listed below:

- Installing a firewall
- Installing security software from a reliable company.
- Setting the operating system and the web browser to update automatically.
- Making sure that the web browser's security setting is high enough to detect unauthorized downloads.

7. Conclusion

This paper is an extended version of the older version of A Design for Blockchain-Based Digital Voting System [30]. The paper proposes a novel e-voting architecture to properly facilitate digitalized elections maintaining vote and voter privacy while preventing fraud. The key processes of the architecture, namely, voter ID creation, voting, vote verification and vote tallying are introduced in explaining the logical design of the digital voting system i-Voting. The key technologies used in affecting such results are the distributed ledger of Bitcoin, namely the blockchain technology and RSA public-private key system. As well as in this new extended version we utilized blind signature in authorizing eligible votes and preventing multi-vote cast by a voter. Moreover, solutions to overcoming many server- and client-side attacks faced by earlier elections were indicated.

Fundamentally, the blockchain technology's ledger decentralization and blocks serialization provide great tools against result manipulation. Therefore, using the proposed blockchain-based architecture leads to achieving sound and fair election. Let us not forget the issue of impersonation. By deliberately opening the system sometime before the election, voters will be allowed to establish their identity. If there is any impersonation, it will be discovered. In addition, an eID's legitimate owner only will be able to use it due to the employed time-critical one-time password algorithm.

Furthermore, an effective approach proposed to address the issue of vote-selling by making vote vendors unreliable is one of the most important solutions to this structure. Several ways and possibilities allow a candidate/party to buy a voter's ballot. Firstly, by taking all login information of the voter and logging into the system in the legitimate voter's place during the polling day and casting a vote. Secondly, asking a voter to attend a specific place and make him cast his vote for them remotely. Thirdly, a voter can be asked to video himself his vote during the voting process. Our proposed approach allows voters to cast a vote unlimited times, whereas only the first vote cast will be accepted with subsequent ones disregarded due to the fact that the structure of the blockchain in itself prevents double voting.

It should be noted that the proposed scheme permits absentee ballot due to its web-based nature allowing remote vote casting; however, mail-in ballot (postal vote) and proxy voting are not permitted. The identified approach involves tradeoffs and may not be suitable for all. Some citizens, especially older ones, may not be able to cope with the complexity of digital voting: eID creation, PKI use, or even using a browser. It can be solved by posting guidance videos or preparing help centers to guide voters who have difficulty following the voting procedures. As a stopgap measure, postal and proxy vote may be authorized in advance in certain exceptional cases, such as inability and incapacity to reach the polling station.

References

1. Brightwell, I., Cucurull, J., Galindo, D., Guasch, S.: An overview of the iVote 2015 voting system, New South Wales Electoral Commission, Australia, Scytl Secure Electronic Voting, Spain (2015)
2. Rivest, R., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, February (1978)
3. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf, (2008)
4. Glaser, F.: Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. Hawaii International Conference on System Sciences, Goethe University Frankfurt, Hawaii, (2017)
5. Kizhakkedathil, N.: A Study Into The Prospects Of Implementing End-To-End Verifiability In Estonia Voting. Tallinn University Of Technology, Faculty of Information Technology, Department of Computer Science, Tallinn (2016)
6. Zyskind, G., Nathan, O., Pantland, A.: Decentralizing Privacy: Using Blockchain to Protect Personal Data. IEEE CS Security and Privacy Workshops, (2015)
7. Lyon, D.: National IDs in a Global World: Surveillance, Security, and Citizenship. Case Western Reserve Journal of International Law Cleveland, Ohio, vol. 44, pp. 607–623, (2010)
8. Johnson, D., Menezes, A.: The Elliptic Curve Digital Signature Algorithm (ECDSA). Technical Report CORR 99-34, Dept. of C&O, University of Waterloo, Canada (1999).
9. M'Raihi, D., Machani, S., Pei, M., Rydell, J.: TOTP: Time-Based One-Time Password Algorithm, Internet Engineering Task Force (IETF), (2011)
10. Dunphy, P., Adleman, L.: A First Look at Identity Management Schemes on the Blockchain. IEEE, VASCO Data Security, (2018)
11. Hastings ,N., Peralta, R., Popoveniuc, S., Regenscheid A.: Security considerations for remote electronic UOCAVA voting. National Institute of Standards and Technology, NISTIR 7770, Feb (2011)
12. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J. J.: Security Analysis of the Estonian Internet Voting System, University of Michigan , Open

Rights Group, ACM New York (2014)
13. Halderman, J. A., Teague, V.: The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election, University of Michigan, University of Melbourne, arXiv:1504.05646v2 [cs.CR] Jun (2015)
14. Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thome, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Beeguelin, S., Zimmermann, P.: Imperfect forward secrecy: How Diffie-Hellman fails in practice, May (2015)
15. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman J. A.: Tracking the FREAK attack, https://freakattack.com/
16. McKay, R.: Flaws in iVote's re-vote process which attempts to defeat coercers, http://www.bigpulse.com/governmentelections#changevoteaw
17. Jones,D. W., Simons, B.: Broken Ballots: Will Your Vote Count?, Stanford University Center for the Study of Language and Information, California (2012)
18. Cross-Site Scripting, http://shiflett.org/articles/cross-site-scripting
19. Parsovs, A.: Practical issues with TLS client certificate authentication, University of Tartu, Software Technology and Applications Competence Center, Estonia (2014).
20. Moura, T., Gomes, A.: Blockchain voting and its effects on election transparency and voter confidence, Proceedings of the 18th Annual International Conference on Digital Government Research, ACM, pp. 574–575, USA(2017)
21. McCorry, P., Shahandashti, S. F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy, in International Conference on Financial Cryptography and Data Security. Springer, pp. 357–375, (2017)
22. Danchev, D.: Study finds the average price for renting a botnet, ZDNet, May (2010), http://www.zdnet.com/blog/security/study-finds-theaverage-price-for-renting-a-otnet/6528.
23. Vonnegut, S.: Preventing XSS: 3 Ways to Keep Cross-Site Scripting Out of Your Apps, Oct (2017), http://www.zdnet.com/blog/security/study-finds-theaverage-price-for-renting-a-otnet/6528
24. Vonnegut, M.: FREAK Attack: What You Need to Know, March (2015), http://www.zdnet.com/blog/security/study-finds-theaverage-price-for-renting-a-otnet/6528
25. Chaum, D. L.: Untraceable electronic mail, return addresses and digital pseudonyms, technical note programming techniques and data structures, Advances in Information Security, 7, 211-219 (1981)
26. Czepluch, J. S., Lollike, N. Z., and Malone, S. O.: The use of block chain technology in different application domains, IT University of Copenhagen, Copenhagen, (2015).
27. Jason, P. C., and Yuichi, K.: E-voting system based on the bitcoin protocol and blind signatures, E-voting system based on the bitcoin protocol and blind signatures, 10, 1, 14-22 (2017).
28. Bartolucci, S., Bernat, P., and Joseph, D.: SHARVOT: secret SHARe-based voting on the blockchain, 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, Gothenburg, 30-34 (2018).
29. Ayed, B. A.: A conceptual secure blockchain-based electronic voting System, International Journal of Network Security and Its Applications (IJNSA), 9, 3, 1-9 (2017).
30. Al-Rawy M., Elci A. (2019) A Design for Blockchain-Based Digital Voting System. In: Antipova T., Rocha A. (eds) Digital Science. DSIC18 2018. Advances in Intelligent Systems and Computing, vol 850. Springer, Cham, pp. 397-407.

Aims and Objectives

Published online by ICS two times a year, Journal of Digital Science (JDS) is an international peer-reviewed journal which aims at the latest ideas, innovations, trends, experiences and concerns in the field of digital science covering all areas of the scholarly literature of the sciences, social sciences. The main topics currently covered include: Digital Communications and Network; Digital Economics, Education, Engineering, Finance, Health Care.

The main goal of the journal is the effective dissemination of original incites/results generated by the human brain and presented/reflected in articles using modern information/digital technology.

Editorial Board

Editors

Contact information

Journal URL: https://ics.events/journal-of-digital-science/

Email: conf@ics.events