A secure end-to-end verifiable e-voting system using zero knowledge based blockchain

Somnath Panja, Bimal Kumar Roy

***Abstract*—In this paper, we present a cryptographic technique for an authenticated, end-to-end verifiable and secret ballot election. Voters should receive assurance that their vote is cast as intended, recorded as cast and tallied as recorded. The election system as a whole should ensure that voter coercion is unlikely, even when voters are willing to be influenced. Currently, almost all verifiable e-voting systems require trusted authorities to perform the tallying process. An exception is the DRE-i and DRE-ip system. The DRE-ip system removes the requirement of tallying authorities by encrypting ballot in such a way that the election tally can be publicly verified without decrypting cast ballots. However, the DRE-ip system necessitates a secure bulletin board (BB) for storing the encrypted ballot as without it the integrity of the system may be lost and the result can be compromised without detection during the audit phase. In this paper, we have modified the DRE-ip system so that if any recorded ballot is tampered by an adversary before the tallying phase, it will be detected during the tallying phase. In addition, we have described a method using zero knowledge based public blockchain to store these ballots so that it remains tamper proof. To the best of our knowledge, it is the first end-to-end verifiable Direct-recording electronic (DRE) based e-voting system using blockchain. In our case, we assume that the bulletin board is insecure and an adversary has read and write access to the bulletin board. We have also added a secure biometric with government provided identity card based authentication mechanism for voter authentication. The proposed system is able to encrypt ballot in such a way that the election tally can be publicly verified without decrypting cast ballots maintaining end-to-end verifiability and without requiring the secure bulletin board.**

***Index Terms*—E-voting, Blockchain, Direct Record Electronic, Authentication, Biometric, Zero Knowledge, End-to-End verifiable.**

I. Introduction

ELECTION is a process of establishing the democracy in the country. It is also one of the most challenging task, one whose constraints are remarkably strict. There has been extensive adoption of Direct-recording electronic (DRE) for voting at polling stations around the world. Starting with the seminal work by Chaum, published in IEEE Security & Privacy [1] in 2004, research on end-to-end (E2E) E-voting has become a thriving field. Informally, the notion of being E2E verifiable refers to have two properties: First, each voter is able to verify if their vote has been cast as intended, recorded as cast. Second, anyone can verify if all votes are tallied as recorded. By contrast, in traditional paper-based voting system, a voter can not verify how their vote is recorded and tallied in the voting process. As with traditional elections, voters go to their polling station, prove that they are eligible to cast vote by presenting their identity card. The voter is given a token [2] that allows them to cast vote for their candidates of choice. Thus the system depends on trustworthy individual at the polling stations, thus leading to the introduction of automated paperless secure e-voting system. This paper presents a secure authenticated DRE based E2E verifiable e-voting system without tallying authorities.

Hao et al. proposed a voting system, called DRE-i (DRE with integrity) [15], to achieve E2E verifiability without involving any tallying authorities (TAs). However, the pre-computation strategy requires that the pre-computed data is securely stored and accessed during the voting phase. This introduces the possibility for an adversary to break into the secure storage module and compromise the privacy of all ballots. To overcome this issue, they provided the voting system DRE-ip [16] (DRE-i with enhanced privacy). DRE-ip achieves E2E verifiability without TAs and simultaneously a significant stronger privacy guarantee than DRE-i. However, both DRE-i and DRE-ip systems necessitates the requirement of a secure public bulletin board (BB). If the public BB is not secure, an attacker may change the already recorded ballots on the BB in such a way that it can not be detected during the audit phase and hence compromising integrity of the system. In this paper, we have proposed a solution to remove requirement of the secure BB. Instead of using secure BB, our system uses an insecure BB, a modified the DRE-ip design and a new blockchain technology. This system prevents vote coercion even when the voter is willing to be influenced. For example, a voter may be asked to vote for an adversary's choice of candidate and show the cast ballot receipt to prove their choice of vote. Our system prevents that by encrypting the vote using Cramer-Shoup encryption. The proposed system also provides a verification of voter eligibility using a secure biometric based authentication mechanism.

S. Nakamoto's work on Bitcoin [17], prompted considerable research on blockchain technology. One other blockchain technology that could be used in e-Voting is Ethereum [18]. Here, we are using a modified blockchain technology for the purpose of a secure decentralized storage.

A. Related work

There has been extensive research on e-voting system over the past two decades. Researchers have proposed a number of E2E verifiable schemes and some of these are used in practice. Notable E2E e-voting system include Votegrity [1] (proposed

Somnath Panja and Bimal Kumar Roy are with Applied Statistics Unit, Indian Statistical Institute, Kolkata, India. E-mails: (somn.math2007@gmail.com, bimal@isical.ac.in).

by Chaum), Markpledge [3], Prêt à Voter [4], STAR-Vote [5], Punchscan [6], scratch & vote [7], Scantegrity , Scantegrity II [8], Helios [9], Bingo Voting [10], Wombat [11], DRE-i [15], DRE-ip [16]. A review of these systems can be found in [12]. Many other schemes follow similar approaches, in particular, a variant of Prêt à Voter, vVote, has been used in 2014 state election in Victoria, Australia [13]. Scantegrity [8] was trialled in local elections in Takoma Park, Maryland, USA [14]. Helios [9] was used to elect Universitè catholique de Louvain in 2009 and it has been used in universities and associations (IACR and ACM). Other schemes that have been used in internal university or party elections include Punchscan [6], Bingo Voting [10], Wombat [11] and DRE-i [15]. However, almost all DRE based E2E verifiable systems require a secure bulletin board. Our system relax the requirement of secure BB and provides efficient solution using blockchain. The system maintains security and integrity even if only one node in the blockchain is honest.

B. Our contribution

1) We have introduced a privacy preserving secure biometric and election ID card based authentication mechanism to the DRE-ip e-voting system using a blockchain technology. 2) We have modified the existing DRE-ip scheme so that if a recorded ballot is modified in between the verification done by the voter at the BB during voting phase and before final tally phase, it would be detected during tallying phase. 3) We combined it with a zero-knowledge based blockchain technology to eliminate the need of secure bulletin board. We allow the adversary to have read and write access to the bulletin board. Our system preserves privacy of the voter and integrity of the election process even if only one node in the blockchain is honest.

C. Organization

Section II discusses the integrity problem in DRE-ip system if the BB is not secure; section III provides an overview of the platform; section IV describes the voter registration process; section V presents a private blockchain to store voter registration information; section VI discusses the voter authentication during voting session; section VII presents vote casting, recording and tallying algorithm; section VIII presents a blockchain technology to store recorded ballots and discusses on integrity and privacy of the system; section IX discusses the concluding remarks.

II. The integrity problem in DRE-ip with insecure BB

We assume that the notion of DRE-ip [16] system provided by Hao et al. is known. DRE-ip system necessitates a secure BB. However, here we assume, the BB is insecure and allow the adversary to have read and write access to the BB. The DRE-ip system records each confirmed vote on the BB in tabulated form.

Initial: g_1, g_2	
$j : R_j, Z_j, P_{WF}\{Z_j\}$	confirmed
Final: t, s	

where $Z_j \in \{g_1^{r_j}, g_1^{r_j} g_1\}$. Now if an adversary knows or guesses correctly which candidate the voter voted for, it can change the value of Z_j by interchanging $g_1^{r_j}$ with $g_1^{r_j} g_1$ or vice versa and simultaneously changing the partial tally t accordingly. This change does not contradicts P_{WF} and can't be detected during tallying phase. This is why we have modified the scheme and used blockchain to secure storage of ballots. The updated scheme has been discussed in section VII.

III. Components of the proposed system

1) The proposed system consists of all the devices required for the DRE-ip [16] system. However, in our case, the publicly accessible BB may be insecure i.e. an adversary have read or write access to the BB. So, it requires a DRE machine with a printer attached to it and a public bulleting board to show the recorded ballot in public. The bulletin board can be a publicly accessible web site. 2) Additionally, a finger print scanner with fingerprint pulse at the sensor and a key pad to input voter identification number must be attached to the DRE machine at the polling station which will verify the eligibility of the voter to cast vote and prompt the DRE machine to proceed with the voting process. 3) We maintain two blockchain per each DRE machine with two different consensus and mining process for two chains. The first chain is created during voter registration process by election authorities to securely store voter authentication information that will be matched during voter authentication process. Second chain is created during the voting phase to securely store recorded ballots. The consensus algorithm and the mining process is discussed in the corresponding sections.

IV. Voter registration

In this phase, the voter will provide personal information including voter identification number and a biometric information, fingerprint, to the authenticated officer. The information provided will be verified during the "Voter Authentication" phase of the election. Following steps are involved in the registration phase:

1. The officer takes details of voter like unique voter identification number, name etc.

2. Officer asks the voter to provide fingerprint. Fingerprint scanner with fingerprint pulse at sensor is used to scan the fingerprint.

3. A biometric based encryption algorithm with enhanced privacy and security [25] is used to the transform fingerprint image to feature based encrypted data. This encrypted biometric data along with voter identification number, name etc. are combined in the form (Voter identification number, name, encrypted biometric data, Flag-"Not voted") and stored in a private blockchain discussed in the next section. Here, Flag represents whether the voter has already voted or not.

V. Private blockchain for voter registration information

We use a blockchain combined with Merkle hash tree to store the registration information. This blockchain is different from the public blockchain used to store the encrypted ballot. We use a private blockchain where only authenticated and trusted party can join in the network and will securely delete all information about fingerprint and voter information after the end of election process. In this chain, the mining process can only be done by administrator. During the registration phase, the administrators are the authenticated officers. During voting phase, DRE machine is the administrator. A provisional voter list is being displayed on public based on this Block chain without fingerprint information about two weeks before start of the election. Any error reported by the public may be corrected before start of the voting process. In that case, the administrator updates the blockchain. After modification, the hash value of the last block in the block is shared among the trusted authorities. We construct the blockchain in the following way.

1. We sort all registration information tuples (Voter identification number, name, encrypted biometric data, Flag-”Not voted”) according to the voter identification number.

2. We decide the number of such tuple to be kept in the single block based on the block size and encrypted biometric data size, in our case, 512.

3. We put first 512 tuples in the first block arranging them in Merkle hash tree structure and then mine that block. Then we continue to create the next block using next 512 tuples, mine the block and so on. We add two field, minimum and maximum voter identification, in each of the block header and Merkle tree internal node headers. These are used to facilitate binary search based on that number.

4. During the voter authentication phase, the DRE will change the flag field of the tuple to ”voted” after successful authentication and update the hashes of the blockchain in $O(log_2 n+m)$ time where n is the height of the Merkle tree and m is the number of block after that block in the blockchain.

VI. Voter authentication

At the polling station, the voter has to verify their identity at the voter authentication phase. If he is an authentic voter, he will be allowed to cast the vote. Following steps are involved in this phase:

1. The unique voter identification number is taken from the voter.

2. The DRE search for the voter identification number in its private blockchain and check whether the voter is already voted on not. If it finds a match in its blockchain and she is not already voted, then it proceeds to step 3 otherwise it prompts ”no match found” or ”already voted ” and goes to step 1.

3. The fingerprint of the voter is taken using a fingerprint scanner with fingerprint pulse at the sensor.

4. The verification procedure of the biometric based encryption algorithm [25] is executed and if it succeeds the voter is allowed to vote otherwise not allowed.

VII. Voting and Tallying Phase

In this section we describe the algorithm for voting phase and tallying phase.

A. Notation

We use same notations that are used in the DRE-ip system. We use $P_K\{\lambda\colon \Gamma = \gamma^\lambda\}$ to denote a non-interactive $proofofknowledge$ of a secret λ such that $\Gamma = \gamma^\lambda$ for publicly known Γ and γ. We shorten the notation to $P_K\{\gamma\}$ where context is clear. We use $P_{WF}\{A : X, ..., Y, Z\}$ to denote a $proofofwell-formedness$ of A with respect to $X, ..., Y, Z$. We shorten the notation to $P_{WF}\{A\}$ where context is clear. These notations were introduced by Camenisch and Stadler [20].

B. Cryptographic setup

Our proposed system work over an elliptic curve in an ECDSA like group setting or a DSA like multiplicative cyclic group setting where the decisional Diffie-Hellman (DDH) assumption holds. In particular, we can choose two large primes p and q such that q divides $(p-1)$. Then, we choose the subgroup $\mathbb{G}_q$ of order q of the group $\mathbb{Z}_p^*$ and assume that g is the generator of $\mathbb{G}_q$. q must be greater than the number of voters. The decisional Diffie-Hellman assumption [21] is given below.

Assumption 1. (DDH) The two probability distribution $\{(g^a, g^b, g^{ab})\colon a, b$ are randomly and independently chosen from $\mathbb{Z}_q^*\}$ and $\{(g^a, g^b, g^c)\colon a, b, c$ are randomly and independently chosen from $\mathbb{Z}_q^*\}$ are computationally indistinguishable in the security parameter $n = log(q)$.

In our protocol, we have used non-interactive zero-knowledge proofs based on Fiat-Shamir heuristic [22] and the proofs are in random oracle model [23].

C. Modified DRE-ip system

The system requires a publicly accessible bulletin board (BB). The BB can be secure or insecure. We allow the adversary to have read and write access to the BB. The system incorporates voter initiated auditing to achieve end-to-end verifiability. We assume one DRE machine used to select a candidate in a regional zone. The BB maintains a public blockchain per one DRE machine to store recorded ballot sent by the DRE machine. Whenever the BB receives a recored ballot from DRE machine, it creates a new block and subsequently mine the block in the corresponding blockchain. In subsequent discussion, we’ll show that only BB or DRE can mine a block when it receives a recorded ballot generated by only this particular DRE-ip algorithm. We assume DRE has a write access to BB over an authenticated channel.

Here we describe the case where there are only two candidates i.e. if v_i represents the vote for i-th ballot, we have $v_i \in \{0, 1\}$. A Benaloh-style voter initiated auditing [24] has been incorporated in the DRE-ip system to achieve individual verifiability, i.e. the voter gets option to audit the ballot composed by DRE to ensure that the DRE is preparing the ballot according to her choice. An audited ballot is not used to cast a vote. Therefore, at the end of the voting phase, the total set of ballots $\mathbb{B}$ will be the union of the audited ballots $\mathbb{A}$ and cast ballots $\mathbb{C}$ i.e. $\mathbb{B} = \mathbb{A} \cup \mathbb{C}$.

We use the Cramer-Shoup cryptosystem to encrypt the ballots. The algorithm is given below.

Key Generation Phase:

In this phase, the DRE executes the Cramer-Shoup key generation algorithm to generate keys. The key generation algorithm is executed only once prior to the voting phase.

1. The system generates an efficient description of a cyclic group $\mathbb{G}_q$ of order q with two distinct, random generators g_1, g_2.

2. It chooses five random values (x_1, x_2, y_1, y_2, z) from $\{0, 1, ..., q-1\}$.

3. It computes $c = g_1^{x_1} g_2^{x_2}, d = g_1^{y_1} g_2^{y_2}, h = g_1^z$.

4. The DRE publishes (c, d, h) along with the description of $\mathbb{G}_q, q, g_1, g_2$ as its public key.

The public key along with group descriptions are shared between DRE and BB and it is published on the BB. The DRE securely deletes the secret key, (x_1, x_2, y_1, y_2, z), and the logarithmic relationship between g_1 and g_2.

Initialization:

Initially, $t = 0, s = 0, s_1 = 0, m = 0, n = 1, n_1 = 1$.

Voting Phase:

This phase involves the voter, the DRE and the BB.

1. The voter enters the booth, initiates the voting and keys in her vote $v_i \in \{0, 1\}$.

2. The DRE generates random $r_i \in \mathbb{Z}_q^*$, evaluates

$U_i = g_1^{r_i}, V_i = g_2^{r_i}, E_i = h^{r_i} g_1^{v_i}, \alpha_i = H(U_i, V_i, E_i)$, where $H()$ is a universal one-way hash function (or a collision-resistant cryptographic hash function, which is a stronger requirement)

$W_i = c^{r_i} d^{r_i \alpha_i}$,

$P_{WF}\{E_i : g_1, g_2, c, d, h, U_i, V_i, W_i\} = P_K\{r_i : ((U_i = g_1^{r_i}) \wedge (V_i = g_2^{r_i}) \wedge (E_i = h^{r_i}) \wedge (W_i = (cd^{\alpha_i})^{r_i})) \vee ((U_i = g_1^{r_i}) \wedge (V_i = g_2^{r_i}) \wedge (E_i/g_1 = h^{r_i}) \wedge (W_i = (cd^{\alpha_i})^{r_i}))\}$

$s_1 = s_1 + r_i, n_1 = n_1 U_i, P_K\{s_1 : n_1 = g_1^{s_1}\}$. Here $P_K\{r_i\}$ is a non-interactive zero knowledge proof of knowledge of r_i whereas $P_K\{s_1\}$ is a non-interactive zero knowledge proof of knowledge of sum of all random numbers generated till now i.e. s_1. At this stage, $s_1 = \Sigma_{j \in \mathbb{B}} r_j, n_1 = \Pi_{j \in \mathbb{B}} U_j$

The DRE machine provides a signed receipt including the unique ballot index i and the ballot content $(U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\})$ to the voter.

3. The voter receives the first part of the receipt and choose to either audit the ballot or confirm her vote.

In case of audit:

4. The DRE adds i to $\mathbb{A}$. A signed receipt of the audit, clearly marked as audited, including r_i and v_i is provided to the voter.

5. The voter takes and keeps the receipt, verifies her choice of vote v_i. If the verification succeeds, voting continues to step 1 else the voter should raise a dispute.

6. The DRE merges both parts of the receipt in a single part, $(i\colon \quad (U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (audited, r_i, v_i))$ and creates a block to mine it in the block-chain and also send this transaction to BB.

In Case of confirmation:

4. The DRE adds i to $\mathbb{C}$, updates the tally, the sum and evaluates:

$t = \Sigma_{j \in \mathbb{C}} v_j, m = \Sigma_{j \in \mathbb{C}} r_j \alpha_j, s = \Sigma_{j \in \mathbb{C}} r_j, n = \Pi_{j \in \mathbb{C}} U_j$.

5. It evaluates $P_K\{s : n = g_1^s\}$, the non-interactive zero knowledge proof of knowledge of the partial sum s.

The DRE provides a signed receipt, clearly marked as confirmed, including $P_K\{s\}$ to the voter. Then the DRE securely deletes both r_i and v_i.

6. The voter leaves the booth with her receipts.

7. The DRE merges both parts of the receipt in a single part, $(i\colon (U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (confirmed, P_K\{s\}))$ and creates a block to mine it at its block-chain and also send this transaction to BB.

8. The voter verifies that her receipts match those on the BB.

Verification by blockchain:

This phase involves the DRE, BB and the underlying blockchain.

The blockchain consensus verifies all the zero knowledge proofs in a receipt before adding it to the blockchain. It also verifies consistency of r_i and v_i in case of an audited ballot. The blockchain consensus mechanism is described in detail in the next section. The blockchain drops a receipt if the verification of any of its zero knowledge proofs fails.

Tallying Phase:

This phase involves the DRE, BB, the blockchain and the public.

1. The DRE posts on the BB the final tally t, final sum s and m.

2. The public:

i) verify all the well-formedness proofs on the BB(well-formedness verification).

ii) verify that for all the audited ballots on the BB: the first part of the receipt, $(U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\})$, is consistent with r_i and v_i.

iii) verify that all the following equations hold(tally verification)

$\Pi_{j \in \mathbb{C}} U_j = g_1^s, \Pi_{j \in \mathbb{C}} V_j = g_2^s, \Pi_{j \in \mathbb{C}} E_j = h^s g_1^t, \Pi_{j \in \mathbb{C}} W_j = c^s d^m$

D. Extension to multiple candidate

If there are more than two candidates, say $n (n >= 3)$, we will consider an upper bound, say N, on the number of voters and will encode the vote for j-th candidate as $v_i = N^{j-1}$. The i-th ballot in that case will of the form

$((U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (audited, r_i, v_i))$ in case of audit or

$((U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (confirmed, P_K\{s\}))$ in case of confirmed vote, where $E_i = h^{r_i} g_1^{N^{j-1}}$. The well-formedness proof $P_{WF}\{E_i\}$ will be 1-out-of-n disjunctive proof and can be stated as:

$P_{WF}\{E_i : g_1, g_2, c, d, h, U_i, V_i, W_i\} = P_K\{r_i : \vee_{j=1}^{n} ((U_i = g_1^{r_i}) \wedge (V_i = g_2^{r_i}) \wedge (E_i/g_1^{N^{j-1}} = h^{r_i}) \wedge (W_i = (cd^{\alpha_i})^{r_i}))\}$

VIII. Storing recorded ballot in public blockchain

A public blockchain created to store recored ballots involving the DRE, BB and public nodes. In this section, we describe how we store the public key, (c, d, h), and recored ballots, both audited and confirmed, in the blockchain. The proposed solution avoid double-inclusion of the same ballot that is already included in the blockchain earlier. The network stores transactions by hashing them into the ongoing blockchain of zero-knowledge based consensus algorithm, forming a record that can not be changed without solving the discrete logarithm problem (DLP) multiple times for each transaction. The longest chain serves as the proof of sequence of events witnessed. As long as there is one node that is not cooperating to attack the network, they'll generate longest chain. As in the other blockchain technology, messages are broadcast, and nodes can leave and rejoin the network at will, accepting the longest chain as proof of what happened while they were gone.

A. Transactions

We define a transactions as an audited ballot receipts, $(i\colon\ (U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (audited, r_i, v_i))$, or a confirmed ballot receipt, $(i\colon (U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (confirmed, P_K\{s\}))$. The first transaction is a special transaction involving (c, d, h) along with the description of $\mathbb{G}_q, q, g_1, g_2$. Each transaction bears with the digital signature of the DRE machine as the owner of the transaction. DRE adds these transactions in the blockchain and also sends this transaction to the BB. The BB again transfers the transaction to its peers.

B. Avoiding double-inclusion of a receipt in the blockchain

Before adding a transaction into the blockchain, a node needs to ensure that the transaction is not already added. To accomplish this, we have included a zero knowledge proof, $P_K\{s_1\}$, in each transaction. The consensus algorithm verifies this zero-knowledge proof to ensure that the transaction is not already included in the blockchain. Transactions must be publicly announced and the nodes must agree on a single history of the order in which they were received.

C. Blocks and hashing

In the proposed solution, a block contains only single transaction. As with the other blockchain systems (as used by Bitcoin and related systems), a cryptographic collision resistant hash function is used to take a hash of a block consisting a single transaction and published widely. Each hash include the previous hash in its hash to form a chain with each additional hash reinforcing the previous one.

D. Distributed consensus mechanism and proof-of-work

The proof of work involves creating a transaction or ballot receipt that consists of the required zero-knowledge proofs. The proof-of-work for a transaction is done by our modified DRE-ip algorithm using Cramer-Shoup cryptosystems and non-interactive zero knowledge proofs and can be verified by this blockchain consensus algorithm. Consensus algorithm include all the verification algorithms of the zero-knowledge proofs. To create a valid transaction, an attacker has to solve multiple discrete logarithm problem (DLP) before the blockchain grows up by one transaction i.e. the work required is exponential. It accepts the longest chain with validated blocks as proof of work witnessed till the time.

The proof-of-work is done by the modified DRE-ip algorithm during its voting phase discussed in section *Voting Phase*. Once a transaction satisfies proof-of-work, it cannot be changed without solving the DLP problem. As later blocks are added after it, the work to change the block would involves redoing all the blocks after it.

The administrator must set up parameters of the blockchain. The public key of the voting algorithm, (c, d, h), is required by the consensus algorithm. The administrator must set (c, d, h) as the global configuration parameters.

The consensus mechanism is stated below.

1. The first transaction must be (c, d, h) along with the description of $\mathbb{G}_q, q, g_1, g_2$. The block involving this transaction must be mined by the administrator i.e either the DRE machine or BB. This transaction is a special transaction that does not follow proof-of-work and its verification algorithm. These are the public information that is needed during the verification by public.

2. From the second block onwards, the blockchain maintains following information. Let $\mathbb{B}, \mathbb{A}, \mathbb{C}$ are sets of all ballots, audited ballots and confirmed ballots respectively till the current transaction. Initially, $n = 1, n_1 = 1$.

The second transaction is an audited or a confirmed ballot receipt. It follows the proof-of-work and verification process. The second block involving this second transaction must be mined by the administrator of the blockchain i.e. either DRE or BB.

3. For i-th transaction, $(i\colon (U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (audited, r_i, v_i))$, or $(i\colon (U_i, V_i, E_i, W_i, P_{WF}\{E_i\}, P_K\{s_1\}), (confirmed, P_K\{s\}))$, it evaluates $\alpha_i = H(U_i, V_i, E_i)$, where $H()$ is the same cryptographic hash function used in encryption phase. Then it calculates $\gamma = cd^{\alpha_i}$ and verifies the non-interactive zero-knowledge proof $P_{WF}\{E_i\}$ given $g_1, g_2, c, d, h, U_i, V_i, W_i, \alpha_i$ for that transaction using "Verifier Algorithm 4" described in APPENDIX A. If verification fails it rejects the block containing the transaction.

4. It updates $n_1 = \Pi_{j \in \mathbb{B}} U_j$ and verifies the non-interactive zero-knowledge proof $P_K\{s_1\}$ given g_1, n_1 for that transaction using "Verifier Algorithm 2" described in APPENDIX A. It rejects the block containing the transaction if verification fails.

5. It checks whether the transaction is a audited ballot or confirmed ballot by checking the "audited" or "confirmed" mark in the transaction.

In case of audited:

6. $\alpha_i = H(U_i, V_i, E_i)$ is already evaluated in step 3. It verifies whether $U_i = g_1^{r_i}, V_i = g_2^{r_i}, E_i = h^{r_i} g_1^{v_i}, W_i = c^{r_i} d^{r_i \alpha_i}$, and rejects the block if the verification fails.

In case of confirmed:

7. It updates $n = \Pi_{j\in\mathbb{C}}U_j$ and verifies the non-interactive zero-knowledge proof $P_K\{s\}$ given g_1, n for that transaction using "Verifier Algorithm 2" described in APPENDIX A. It rejects the block containing the transaction if verification fails.

If all the verification successes then the block is added to the blockchain or else rejected.

E. Network

The network is similar to the other blockchain systems like Bitcoin [17].

F. Integrity of the system

The voter initiated auditing performs two checks: first, by observing the first part of the receipt is provided before deciding either to audit or confirm the ballot; second, by checking that all receipts match what is published on the BB. These verification ensure that the votes are cast as intended and recorded as cast. Since, each receipt consist of several zero-knowledge proofs that are to be verified by the blockchain, an attacker must solve three discrete log problems to change a ballot. Also, to change a single ballot, the attacker has to redo the task for all ballots in the blockchain added after that block. It can be checked that if all well-formdness proofs and zero-knowledge proofs are correct and final tally verification succeeds, then the reported tally is the correct tally of all confirmed votes. This ensures that the votes are tallied as recorded. So the system achieves end-to-end (E2E) verifiability.

G. Privacy of the voter

As each vote is encrypted using Cramer-Shoup cryptosystem, the privacy of the voter is kept secret. The DRE must not reveal its partial tally and sum information otherwise the privacy of the ballots cast during the attack period is lost-a loss which is inevitable but the ballots cast outside of the attack period remains private.

IX. Conclusion

In this paper, we have described a way store the voter registration information and encrypted ballot using blockchain. We have shown that the system provides an efficient and practical DRE-based voting solution preserving privacy and secrecy of ballots without secure bulletin board or hardware storage even if the adversary gets temporary access to the DRE machine. The use of this public blockchain ensures the ballot integrity even if only one node in the blockchain network is honest.

Appendix A

In this section, we present non-interactive zero-knowledge proof of knowledge. Similar proofs are also described in DRE-ip [16] system.

Algorithm 1:

A prover with identifier ID generates a proof of knowledge of a secret λ such that $(\Gamma_1 = \gamma_1^\lambda) \wedge ... \wedge (\Gamma_4 = \gamma_4^\lambda)$ for known $ID, \Gamma_1, \gamma_1, ..., \Gamma_4, \gamma_4$.

```
Input: $ID, \Gamma_1, \gamma_1, ..., \Gamma_4, \gamma_4, \lambda$ such that $(\Gamma_1 = \gamma_1^\lambda) \wedge ... \wedge (\Gamma_4 = \gamma_4^\lambda)$
Output: $\eta = P_K\{\lambda\colon (\Gamma_1 = \gamma_1^\lambda) \wedge ... \wedge (\Gamma_4 = \gamma_4^\lambda)\}$
begin
choose random $w \in \mathbb{Z}_q^*$
calculate $t_1 = \gamma_1^w, t_2 = \gamma_2^w, t_3 = \gamma_3^w$ and $t_4 = \gamma_4^w$.
calculate
$c = H(ID, \gamma_1, \Gamma_1, \gamma_2, \Gamma_2, \gamma_3, \Gamma_3, \gamma_4, \Gamma_4, t_1, t_2, t_3, t_4)$
calculate $r = w - c\lambda$
return $\eta = (c, r)$
```

Verifier Algorithm 2:

Verification of proof η generated by *Algorithm 1* given $ID, \Gamma_1, \gamma_1, \Gamma_2, \gamma_2, \Gamma_3, \gamma_3, \Gamma_4, \gamma_4$.

```
Input: $ID, \Gamma_1, \gamma_1, \Gamma_2, \gamma_2, \Gamma_3, \gamma_3, \Gamma_4, \gamma_4, \eta = (c, r)$
Output: successful or failure
begin
calculate
$t_1 = \gamma_1^r\Gamma_1^c, t_2 = \gamma_2^r\Gamma_2^c, t_3 = \gamma_3^r\Gamma_3^c$ and $t_4 = \gamma_4^r\Gamma_4^c$.
calculate
$c' = H(ID, \gamma_1, \Gamma_1, \gamma_2, \Gamma_2, \gamma_3, \Gamma_3, \gamma_4, \Gamma_4, t_1, t_2, t_3, t_4)$
if $c' = c$ then return successful
else return failure
```

Algorithm 3:

A prover with identifier ID generates a proof of knowledge of a secret λ such that either $(\Gamma_1 = \gamma_1^\lambda) \wedge ... \wedge (\Gamma_4 = \gamma_4^\lambda)$ or $(\Gamma_5 = \gamma_5^\lambda) \wedge ... \wedge (\Gamma_8 = \gamma_8^\lambda)$ for known $ID, \Gamma_1, \gamma_1, ..., \Gamma_8, \gamma_8$.

```
Input: $ID, (\Gamma_i, \gamma_i)_{i=1}^8, \lambda$ such that $\Gamma_1 = \gamma_1^\lambda, ..., \Gamma_8 = \gamma_8^\lambda$
Output: $\eta = P_K\{\lambda\colon ((\Gamma_1 = \gamma_1^\lambda) \wedge ... \wedge (\Gamma_4 = \gamma_4^\lambda)) \vee ((\Gamma_5 = \gamma_5^\lambda) \wedge ... \wedge (\Gamma_8 = \gamma_8^\lambda))\}$
begin
choose random $w, r_2, c_2 \in \mathbb{Z}_q^*$
calculate $t_1 = \gamma_1^w, ..., t_4 = \gamma_4^w$ and $t_5 = \gamma_1^{r_2}\Gamma_1^{c_2}, ..., t_8 = \gamma_8^{r_2}\Gamma_1^{c_2}$.
calculate
$c = H(ID, (\gamma_i, \Gamma_i)_{i=1}^8, (t_i)_{i=1}^8)$,$c_1 = c - c_2$
calculate $r_1 = w - c_1\lambda$
return $\eta = (c_1, c_2, r_1, r_2)$
```

Verifier Algorithm 4:

Verification of proof η generated by *Algorithm 3* given $ID, \Gamma_1, \gamma_1, ..., \Gamma_8, \gamma_8$.

```
Input: $ID, (\Gamma_i, \gamma_i)_{i=1}^8, , \eta = (c_1, c_2, r_1, r_2)$
Output: successful or failure
begin
calculate
$t_1 = \gamma_1^{r_1}\Gamma_1^{c_1}, t_2 = \gamma_2^{r_1}\Gamma_2^{c_1}, t_3 = \gamma_3^{r_1}\Gamma_3^{c_1}$ and $t_4 = \gamma_4^{r_1}\Gamma_4^{c_1}$ and
$t_5 = \gamma_5^{r_2}\Gamma_5^{c_2}, t_6 = \gamma_6^{r_2}\Gamma_6^{c_2}, t_7 = \gamma_7^{r_2}\Gamma_7^{c_2}, t_8 = \gamma_8^{r_2}\Gamma_8^{c_2}$.
calculate
$c' = H(ID, (\gamma_i, \Gamma_i)_{i=1}^8, (t_i)_{i=1}^8)$
if $c' = c_1 + c_2$ then return successful
else return failure
```

References

[1] Devid Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE security & privacy*, 2(1):38-47, 2004.

[2] Tadayosh Kohno, Adam Stubblefield, Aviel D. Rubin, Dan S. Wallach. Analysis of an Electronic Voting System. *IEEE Symposium on Security & Privacy, 2004.*

[3] C. A. Neff. Practical high certainty intent verification for encrypted votes, 2004. Available from http://citeseer.ist.psu.

[4] P. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia. Prêt à Voter: a voter-verifiable voting system. *IEEE T. Inf. Foren. Sec.*, 4(4):662-673, Dec 2009.

[5] S. Bell, J. Benaloh, M. D. Byrne, D. DeBeauvoir, B. Eakin, G. Fisher, P. Kortum, N. McBurnett, J. Montoya, M. Parker, O. Pereira, P. B. Stark, D. S. Wallach and M. Winn. STAR-Vote: A secure, transparent, auditable, and reliable voting system. *USENIX Journal of Election Technology & Systems*, 1(1):18-37, 2013.

[6] K. Fisher, R. Carback, and A. T. Sherman. Punchscan: Introduction and system definition of a high-integrity election system. In *Workshop on Trustworthy Elections (WOTE)*, 2006.

[7] B. Adida e R. L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. *WEPS06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, New York, 2006.

[8] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. Rivest, P. Ryan, E. Shen, A. Sherman, and P. Vora. Scantegrity II: End-to-end verifiability by voters of optical scan elections through confirmation codes. *Information Forensics and Security, IEEE Transactions on*, 4(4):611-627, Dec 2009.

[9] B. Adida. Helios: Web-based open-audit voting. In *USENIX Security Symp.*, volume 17, pages 335-348, 2008.

[10] J.-M. Bohli, J. Müller-Quade, and S. Röhrich. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In *E-Voting and Identity*, pages 111-124. Springer, 2007.

[11] J. Ben-Nun, M. Llewellyn, B. Riva, A. Rosen, A. Ta-Shma, and D. Wikström. A new implementation of a dual (paper and cryptographic) voting system. In *EVOTE2012: 5th Intl Conf. on Electronic Voting*, pages 315-329, 2012.

[12] Feng Hao and Peter Y A Ryan (Eds). *Real-world Electronic Voting: Design, Analysis and Deployment*, Series in Security, Privacy and Trust. CRC Press, 2016.

[13] C. Culnane, P. Y. A. Ryan, S. Schneider, and V. Teague. vVote: A verifiable voting system. *ACM Trans. Inf. Syst. Secur.*, 18(1):3:1-3:30, June 2015.

[14] R. Carback, D. Chaum, J. Clark, J. Conway, A. Essex, P. Herrnson, T. Mayberry, S. Popoveniuc, R. Rivest, E. Shen, A. Sherman, and P. Vora. Scantegrity II municipal election at Takoma Park: The first E2E binding governmental election with ballot privacy. *USENIX Security Symp.*, pages 291-306, 2010.

[15] F. Hao, M. N. Kreeger, B. Randell, D. Clarke, S. F. Shahandashti, and P. H.-J. Lee. Every vote counts: Ensuring integrity in large-scale electronic voting. *USENIX Journal of Election Technology & Systems*, 2(3):1-25, 2014.

[16] Siamak F. Shahandashti and Feng Hao. DRE-ip: A Verifiable E-Voting Scheme without Tallying Authorities. The *21st European Symposium on Research in Computer Security (ESORICS)*, 2016.

[17] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[18] G. Wood. Ethereum: a Secure Decentralised Generalised Transaction Ledger. *Sante Publique (Paris)*, vol. 28., no. 3, pp 391397, 2016.

[19] Yehuda Lindell, Anna Lysyanskaya, and Tal Robin. On the composition of authenticated bizantine agreement. In *STOC*, pages 514-523, ACM, 2002.

[20] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Crypto97*, volume 1294 of *LNCS*, pages 410-424. Springer, 1997.

[21] W. Diffie and M. E. Hellman. New directions in cryptography. In *Information Theory, IEEE Transactions on*, 22(6):644-654, Nov 1976.

[22] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Crypto86*, volume 263 of *LNCS*, pages 186-194. Springer, 1987.

[23] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS93*, pages 62-73. ACM, 1993.

[24] J. Benaloh. Ballot casting assurance via voter-initiated poll station auditing. In *USENIX Workshop on Accurate E-Voting Technology (EVT)*, pages 14, 2007.

[25] Ratha, N.K., Connell, J.H., Bolle, R.M. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal 40(3)*, 614634 (2001).