Noname manuscript No.
(will be inserted by the editor)

TRVote: A New, Trustworthy and Robust Electronic Voting System

Fatih Tiryakioğlu · Mehmet Sabir Kiraz · Fatih Birinci

Received: date / Accepted: date

Abstract We propose a new Direct-Recording Electronic (DRE)-based voting system that we call TRVote. The reliability of TRVote is ensured during the vote generation phase where the random challenges are generated by the voters instead of utilizing the random number generator of the machine. Namely, the challenges of voters are utilized to prevent and detect a malicious behavior of a corrupted voting machine. Due to the unpredictability of the challenges, the voting machine cannot cheat voters without being detected. TRVote provides two kinds of verification; "cast-as-intended" is ensured during the vote generation phase and "recorded-as-cast" is ensured through a secure Web Bulletin Board (WBB). More concretely, voters can verify that their votes are cast as intended via a zero-knowledge proof on a printed receipt using QR codes. After the election, the central server broadcasts all receipts in a secure WBB where the voters (or, perhaps proxies) can check whether their receipts appear correctly. In order to implement the voting process, the proposed voting machine has simple components such as mechanical switches, a touchscreen, and a printer. In this system, each candidate is represented by a mechanical switch which is equipped within the voting machine. The machine has a flexible structure in the sense that the mechanical switches can be placed and removed as plug-ins depending on the number of candidates which allows to support arbitrary number of candidates. We show that the proposed system is robust and guarantees privacy of voters. We further analyze that it is universally verifiable and secure against coercion.

Keywords Electronic Voting · DRE-Based Systems · Security · Privacy

F. Tiryakioğlu
TUBITAK BILGEM, Gebze, Turkey.
Tel.: +90-262-6481719
Fax: +90-262-6481100
E-mail: fatih.tiryakioglu@tubitak.gov.tr

M.S. Kiraz
Cyber Technology Institute, De Montfort University, Leicester, UK.
E-mail: mehmet.kiraz@dmu.ac.uk

F. Birinci
TUBITAK BILGEM, Gebze, Turkey.
Tel.: +90-262-6481700
Fax: +90-262-6481100
E-mail: fatih.birinci@tubitak.gov.tr

1 Introduction

In conventional election systems voter manually marks the paper ballot, puts it in the voting box, and then the ballot is manually counted by the election officials. Despite several security and usability issues, the conventional mechanisms are believed to be mature and robust, and they continue to be used all over the world. With the developments in the computer science and technology, however, electronic devices have become an indispensable part of our life. New electronic voting systems could also benefit from these technological developments in terms of accuracy, convenience, flexibility, accessibility, quick announcement of the results, even mobility and cost. After some field experience as well as academic research, it became clear that it was not so easy to make voting securely with electronic devices. The main concerns are reliability of the system and privacy of the voters. Other security concerns include coercion, vote selling-buying, and incorrect tallying process. Designing a well-defined voting protocol satisfying all these concerns but also being user friendly (i.e. usable) is a challenging task. The main difficulty is basically due to the fact that two conflicting requirements, transparency and anonymity, are demanded at the same time.

In this context, researchers have proposed various systems with different names like "end-to-end verifiable", "receipt-

based", or "universally verifiable" voting systems in the last two decades. "End-to-end verifiable" systems mainly seek to ensure a voter that her vote was "cast-as-intended", "collected-as-cast", and "tallied-as-cast" without disclosing voter-vote connection. In the receipt-based systems, the voter is given a receipt which does not reveal any information about his voting choices, and the data on all receipts is made public after the polls are closed. After the elections, the voter check whether his receipt exists on public board. If the receipt exists, it is guaranteed that it is tallied accurately. If the receipt does not exist, it will be perceived as corruption and voter will use her receipt for objection.

[28] classify voting systems focusing on verifiability as remote and supervised environment systems. According to its classification:

- Code Voting [11], PGD [26], MarkPledge [35], Helios [1] and Civitas [15] are remote environment systems;
- Chaum04 (Visual Crypto) [10], Prêt à Voter [13], Punchscan [21], Scratch & Vote [2], ThreeBallot, VAV, Twin [39], Scantegrity I and II [18, 12] and Bingo [8] are supervised environment systems. VoteBox [42], vVote (a variant of Prêt à Voter) [17], VeriScan [5], Wombat [40], Star-Vote [6] and DRE-i [25] can also be included to supervised environment systems.

Supervised environment systems are also called pollside systems in [7]. Verifiable pollside systems can be roughly categorized as:

- "electronic ballot" systems (Bingo [8], VoteBox [42], DRE-i [25]),
- "ordinary ballot" optical scan systems (VeriScan [5], but its scanner can also print VOIT mark on ballot),
- pre-ballot optical scan systems (Prêt à Voter [13], Punchscan [21], Scratch & Vote [2], Scantegrity I and II [18, 12], ThreeBallot, VAV, Twin [39], vVote [17]) and
- ballot-on-demand optical scan systems (Chaum04 [10], Wombat [40], Star-Vote [6]).

All systems may give receipt for E2E verification. To the best knowledge of us, some parts of ballot form receipt in all pre-ballot optical scan systems.

Although all verifiable pollside systems provide reliability and privacy, it seems difficult to completely prevent coercion. Forced randomization, chain-voting and stimulus attacks are some of coercion attacks that are valid especially on pre-ballot optical scan systems. Dispute resolution is also an important feature to provide. When a voter claims something to be wrong during voting, system should detect and prove which one, voting machine or voter, is unfair.

In TRvote, we propose a verifiable pollside "electronic ballot" voting machine and system that have also coercion resistance and dispute resolution features. At the heart of our machine lies a novel "cast-as-intended" method that makes the DRE-style machine "software independent". In this way, a possible malicious code attempting to change intended vote can be detected. The voting machine has also mechanical switches that only voter can alter and when a voter objects the voting process in right or unfair manner, the switches are used to resolve dispute.

1.1 Organization

The rest of the paper is organized as follows. Related works about pollside verifiable systems are discussed in the following section. In Section 3, the necessary preliminaries and underlying cryptographic mechanisms are given. In Section 4, the components of the system are explained in details. In Section 5, our new voting system is proposed. Section 6 gives the analysis of the security of the system. Finally, Section 7 concludes the paper.

2 Previous Work

There are mainly two approaches of pollside systems to provide cast-as-intended (CAI) verification: multi layer/sheet ballot and cast-or-challange/spoil/audit approaches. One exception to this is Bingo [8] where random numbers generated and committed prior to voting are used during voting .

In multi layer/sheet ballot approaches, voter intention is hided mostly destroying a layer/sheet of a marked ballot that shows voter intention. In Chaum04 [10], Prêt à Voter [13], Punchscan [21] and Scratch & Vote [2] this is the case. In ThreeBallot and VAV [39], vote is composed of three ballots and all three together compose a ballot. In Scantegrity II [12], some verification codes on the ballot are hided by invisible ink, and marking a candidate uncovers the code on somehow a second layer.

Cast-or-challenge/spoil/audit approaches use Benaloh's solution to ballot casting assurance problem [4]. In this approach, voters are able to perform trial voting as well as real voting. Because it is not certain it would be a trial or real voting till the end of voting procedure, encryption of "intended candidate" is "enforced" to the voting machine. The secret used to calculate ciphertext has to be revealed if it was trial voting. This results in revealing also plaintext of it. Voting machine has to prove to the voter that revealed plaintext is the same as the voter's intention. In Votebox [42], an "electronic ballot" system, voting machine broadcasts encrypted voting intention over a network as a commitment. After that, voter determines to cast the vote or challenge the voting machine via pressing a physical button. If challenge occurs, the voting machine broadcasts also the secret used in the encryption. A person following the broadcast verifies the encryption by using the secret, and the voter can learn the result of challenge from the verifier by contacting her via for

example a web site. VeriScan [5], an ”ordinary ballot” system, is somehow Votebox’s optical paper version: a voter marks ordinary optical ballot and feeds it into a scanner that has printing receipt capability. Ciphertext and secret are not broadcasted but printed on a receipt. The same physical button is used for cast or challenge decision. Because there is an ordinary optical paper here, the scanner returns the ballot to the voter with VOID print on it (on challenge) or drops it into a box automatically (on cast). Receipt contains ciphertext and a signature (on cast) or ciphertext, verifiable decryption data (i.e. secret) and a signature (on challenge). All receipts as well as the challenged ballots’ plaintexts are posted on a bulletin board. Voters check their receipts for ”cast ballots”, and their receipts and returned ballots for ”challenged ballots”.

In Wombat [40], a ”ballot-on-demand” system and very similar to Veriscan [5], voter selects voting choices with DRE-style interface instead of ordinary ballot, and choice is printed both in plaintext and encrypted form as detachable parts. Cast-or-challenge decision is performed by DRE-style interface instead of physical button. If the decision is ”cast”, it is printed ”Cast” on the bottom of the ballot. If the decision is ”audit”, audit information is printed on the bottom of the ballot. Plaintext part of the ballot to be cast is detached and dropped into a ballot box. The ballot’s encrypted part is scanned to post on a bulletin board and it is taken home by the voter. If it is audit ballot, its audit information has to be consistent with the plaintext part of it. Any inconsistency proves the voting machine to be cheating.

Star-Vote [6], a ”ballot-on-demand” system, is similar to Wombat [40]. In Star-Vote, there is not a single voting machine but ”a controller, voting terminals and a ballot box” on an internal network. Controller gives 5-digit authorization code to the voter, so that the voter can vote on any of the voting terminals. There is not a button (physical or DRE-style) for cast-or-challenge decision here, and no decision mark is printed on the ballot: voter gets the ballot which has detachable plaintext and encrypted parts (rather the hash of the encryption). The voter deposits plaintext part of the ballot into a box in case of cast, or keep it as a whole in case of challenge. Cast or challenged ballot’s serial number is communicated to the controller to record which ballots to tabulate. Cast ballot’s receipt (i.e. encrypted part of the ballot) and challenged ballot’s both plaintext and receipt are taken home by the voter, and the voter verifies that they all are posted on a bulletin board following the polls are closed. Additionally it has to be proved on the bulletin board that when all challenged ballot’s encrypted parts are decrypted, the results are the same as the plaintext parts of the ballots.

DRE-i [24], an “electronic ballot” system like Votebox [42], does not need public and shared private keys for encrypting votes and decrypting final tally, respectively. DRE-i publishes commitment data for every potential ballot before the election and the secrets related to commitment data is stored securely in the voting machines. Voter selects her voting choice from DRE interface, the voting machine calculates ciphertext by using secret, prints it on the receipt, and signs the receipt. Cast-or-challenge decision is performed by interface. If the voter selects cast, voting is finished. If the voter selects challenge (i.e. audit), voting machine prints voting choice and the ciphertext of the unvoted choice. Because of mathematical relation between ciphertexts, one can easily calculate which ciphertext corresponds to which voting choice. If the first printed ciphertext is not the voting choice, it is revealed that the voting machine has cheated. An internet voting system (i.e not pollside) build on DRE-i protocol is also used Newcastle University [23]. Other versions of the method have since been developed: DRE-ip doesn’t publish any commitment data before the election and ballot encryption is performed during voting [43], and DRE-Borda, adopted from DRE-ip, supports ranked-choice voting [3].

Pre-ballot systems (i.e. multi layer/sheet ballot approaches except Chaum04 [10]) inherent chain of custody issues. Authority can leak ballot form information (i.e. the random numbers used for ballot generation), the contents of generated ballots can also be leaked during distribution and storage [41]. Two pre-ballot systems, which have been used in real elections locally, stand out: Prêt à Voter [13] and Scantegrity II [12]. Prêt à Voter [13] is vulnerable to chain voting, signature (i.e. İtalian attack) and forced randomization attacks. vVote [17], last variant of Prêt à Voter, is both pre-ballot and ballot-on-demand system and it seems to improve chain of custody issues by using randomness servers. Beside these, it avoids chain voting attack by ignoring the printed ballots that are older than 5 minutes. [33] claims to prove that Scantegrity II [12] has optimal level of coercion resistance.

On the other hand, cast-or-challenge systems can be vulnerable to stimulus attacks defined in [29]. A coercer uses ciphertext as a communication channel to a voter. He directs voter to cast or challenge according to ciphtertext patterns. This attack is generally resolved by hiding ciphertext from voters until cast-or-challenge decision is performed [42]. Votebox [42] addresses this issue by two solution. Firstly, a challenging voter states her intention to a poll worker and the poll worker physically locks the cast button. Secondly, cellular phone use is banned while voting, so that voters cannot access broadcasted ciphertext. In VeriScan [5] and Wombat [40], ciphertext is hidden from a voter by using a shield till the end of her cast-or-challenge decision. Star-Vote’s current version [6] does not present a solution to this attack. It is claimed that it is impossible to prevent stimulus attacks in the age of digital (eyeglasses with embedded video cameras, etc.).

Another issue about cast-or-challenge systems is dispute-resolution. Voter may object to cast or challenge decision rightly or unfairly. In such a scenario voter may be right: A malware affected voting machine changes her intention and encrypts an unintended choice. When the voter challenges voting machine via a physical button or touch screen, voting machine follows the procedures as if the voter selected to cast. Voter objects that, but she cannot prove she challenged the voting machine. Malware affected voting machine may operate normally in test environments and it may realize this attack to a certain extent in real elections. Attacks can also be activated by external signal. In the opposite scenario, an unfair voter may claim that she had challenged the voting machine but the voting machine followed cast procedures. Even this time it cannot be proved that voter did not challenge the voting machine. This all is valid for Votebox [42], VeriScan [5], Wombat [40] and DRE-i [25]. Because challenge or cast decision is not performed by a button or touch screen, this attack is not valid for Star-Vote [6].

It has been claimed that clash attack based on giving the same receipt to multiple voters applies to various preballot and cast-or-challenge systems [34,37]. In this attack, the malwared system can produce new ballots unnoticed.

2.1 Contributions

The main contribution of this paper is to provide a new DRE-based and reliable e-voting system, which we call TRVote. The voting machine has one-way and two-way mechanical switches that make it "software independent" together with a touchscreen and a printer. At the core of our system, a novel vote encryption method ensuring "cast-as-intended" is proposed. More concretely, the voter first randomly pulls up or down a switch designated to the preferred candidate. Pulling up and down corresponds to switching to Red and switching to Blue, respectively. Receiving voting choice and Red or Blue challenge, voting machine should encrypt the intended voting choice with the challenge, not one of the unintended candidates. Once the encrypted vote has been printed on the receipt, the voter assigns Red or Blue challenge for each of the remaining candidates by determining a pattern for all candidate switches. For example, the voter can select the flat pattern. If the voter has pulled up (i.e. switched to Red) the switch for preferred candidate, the flat pattern means that all candidate switches are in the up (i.e. switched to Red) position. If the voter has pulled down the switch (i.e. switched to Blue) for preferred candidate, the flat pattern means that all candidate switches are in the down (i.e. switched to Blue) position. Finally, the voting machine provides zero-knowledge proof that the encrypted vote is one of elements of the voter's challenge set, i.e. (candidate, color) pairs. The proof is printed on the receipt via QR codes. Any party can easily verify the correctness of the proof by devices like smartphones or tablets.

The security and reliability of our system is statistically ensured even if the software running on the machine is corrupted. We show that a voter can immediately notice unexpected activity in the system or a malicious behaviour of a corrupted machine by using her challenges and her receipt. Furthermore, our system is also resistant to vote-selling and coercion attacks because nobody can guarantee the voter votes for a predefined candidate. Because, the vote is printed in encrypted form on receipt and verification of the vote is done via a zero-knowledge proof showing that the encryption is performed honestly (while revealing no information about votes). More concretely, the receipt includes the encryption of vote, the colors (i.e. switch positions) of the candidate switches, zero-knowledge proof showing that the encryption is one element of the (candidate, color) pairs, and the signature of the machine. The votes are verified on a secure WBB (Web Bulletin Board) which assures that the votes are recorded and transmitted to the central server correctly. The tallying process is handled as in usual way with the decryption key which was securely generated and distributed to independent parties.

Because our system is not a preballot system, chain of custody issues are not valid. Our system is not a ranked voting system, so signature attack (i.e. Italian attack) that is valid in ranked voting systems is not possible. Furtermore, it is not vulnerable to chain voting attacks that are specific to ordinary ballot, preballot and ballot-on-demand systems. Our system is also resistant to forced randomization attacks, because a coercer who forces a voter to have a special pattern on her receipt cannot cause a random vote. Clash attack does not seem to be possible because the pattern is not yet clear when encrypted votes are printed on the receipt. A malware machine attempting to print the encrypted vote in the receipt of a previous voter to the voter's receipt is detected if the voter selects a different pattern. Stimulus attack which is possible in our system is resolved by hiding ciphertext from voters until challenge generation (i.e. pattern generation) is performed, like some other cast-or-challenge systems.

Our system has also dispute-resolution feature. When a voter claims something to be wrong during voting, system should detect which one, voting machine or voter, is lying. In cast-or-challenge systems cast or challenge decision can be a source of dispute. Voter may object to cast or challenge decision such that she may claim she challenged the voting machine but the voting machine followed the cast procedures. Our system can resolve such a dispute by using mechanical switches on it.

Last but not least, the new proposed machine has a touchscreen display which is more user-friendly and usable that orient and inform voters about voting process. Furthermore, the voting machine has adjustable plug-ins integration capa-

bilities and forms a flexible structure where candidate and pattern switches can be removed and placed as plug-ins. In this way, the machine can provide to work with sufficiently large candidates. Additionally, voter can object by using her receipt at any step from the initialization of the voting process to the tallying.

3 Preliminaries

In this section we will present the general setup and symbols that are needed for presenting our protocol in the next section.

3.1 Classification of Voting Systems

Definition 1 (Pre-ballot optical scan systems) Pre-ballot or preprepared ballot systems are also called as paper based systems and ballots are specially prepared before elections. Each ballot is unique.

Definition 2 (Ballot-on-demand optical scan systems) Ballot-on-demand or instant ballot based systems ballots are generated during the elections.

Definition 3 (Ordinary ballot optical scan systems) Ordinary ballot optical scan systems ballots are the same, and they are generated like in classical voting systems.

Definition 4 (Electronic ballot systems) Electronic ballot systems ballots are displayed on a screen, and voter selects voting choice from the interface.

For example Punchscan [21] system ballot consists of two separate sheets, Prêt à Voter [13] ballot is split between two halves, and left half of the ballot displays the candidates in a permuted order, while the right half has boxes that are marked to indicate a vote [30], etc. Ballot as a whole shows which row or column of choice corresponds to which candidate. Split or separated part of the ballot does not show the selected candidate without remaining of ballot. Because the ballot itself shows the way the choice goes to candidate, the ballots must be generated secure and accurately. The accuracy of vote counts in Punchscan [21] and Prêt à Voter [13] is dependent on the proper construction of ballots. To deal with this both systems rely on preelection audits of the ballots to ensure the ballots were created correctly [30]. Beside these, all paper based systems (Prêt à Voter [13], Punchscan [21], Scratch & Vote [2], Threeballot [39]) has a threat by mallicious authorities; they can track the ballots handled by them to the voters.

3.2 Threshold Homomorphic Public-Key Cryptosystem

In this part, we briefly describe the underlying cryptographic primitives of the proposed voting system. Let $m \in \mathscr{M}$ denote its plaintext space, $c \in \mathscr{C}$ the ciphertext space, and $r \in \mathscr{R}$ its randomness for a given a public key encryption scheme. Let $c = \mathsf{Enc}_{pk}(m;r)$ denote an encryption of a message m under the public key pk where r is a random value. pk is the public key of election authority which is stored into the application of the voting machine. Let sk be its corresponding private key which decrypts a ciphertext to a message. Note that the private key sk is shared between n independent parties and each of the shares is known only by corresponding party. More concretely, during the key generation process the key pair $(pk,(sk_1,\cdots,sk_n))$ is generated in such a way that each party P_i privately obtains sk_i, where $i = 1,\cdots,n$. Also, decryption can be performed only if at least t of them collude and cooperate during the tallying process.

In the proposed voting scheme, we use the most widely used ElGamal [20] public key cryptosystem which is semantically secure. We also make use of additive homomorphism property of ElGamal cryptosystem for especially zero knowledge proofs. A public key encryption scheme is said to be additively homomorphic if for given $c_i = \mathsf{Enc}(m_i;r_i)$ and $c_j = \mathsf{Enc}(m_j;r_j)$, the equality $c_i c_j = \mathsf{Enc}(m_i + m_j; r_i + r_j)$ holds. As a consequence, it is also true that $\mathsf{Enc}(m;r)^s$ is equal to $\mathsf{Enc}(ms;rs)$ for a known integer s in an additively homomorphic encryption scheme. Another consequence of these properties is the re-randomization of encryption, by observing that $\mathsf{Enc}(m;r)\mathsf{Enc}(0;r')$ is a new encryption whose plaintext is again m (and its randomness is $r+r'$). Re-randomizing and shuffling a list of ciphertexts are known as mixnet used to tally the votes [14,2,31].

3.3 Honest-Verifier Zero-Knowledge Proofs: Σ Proofs

A Σ-protocol for a relation $R = \{(p,w)\}$ is a commit-challenge-response zero-knowledge protocol between a prover and a verifier. Both the prover and the verifier have a common input value p as common input, and the prover has a private input w called "witness", $(p,w) \in R$. A Σ-protocol is a zero knowledge proof of knowledge for relation R satisfying special soundness and special honest-verifier zero-knowledge (see [16] for details). In our proposed system, we will use OR-composition of Schnorr protocol. Namely, there is a pattern of a candidate list and the voting machine generates an OR-proof combining a real run and a simulated run of the protocol to show that the encrypted vote is indeed one of the candidate in the list. Note that this procedure basically prevents invalid votes to be generated by a corrupted machine during the voting process which is critical to ensure the correctness and to eliminate bad consequences on the reputation of the election.

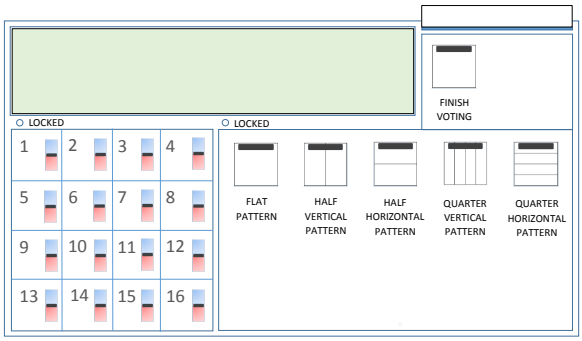

Fig. 1 Proposed adjustable voting machine with 16 candidates.

4 Components of Our System

4.1 Properties of Lockable Mechanical Switches

We are going to design a new type of mechanical switch which we call *Lockable Mechanical Switch* (LMS). It is similar to a fuse switch except that it can be locked. LMSs can be easily plugged and locked into the voting machine depending on the number of candidates.

In the proposed system, there are three types of switches: Candidate, pattern, and Finish Voting switches. The candidate switches are two-way, up (e.g. blue) and down (e.g. red). The pattern and Finish Voting switches are only one-way, down. These switches will be basically used as a physical challenge set of a voter. More concretely, LMS has the following structures:

- There will be k candidate LMSs plugged into the voting machine where k denotes the number of candidates. Each candidate LMS will represent a candidate (denote LMS_i for the i-th candidate $\mathsf{Candidate}_i$).
- The candidate LMSs are lockable in such a way that when a candidate LMS is switched to “up” or “down”, the voting machine can lock and unlock it during the voting process. Up or down switching causes blue or red color to appear in the switch background, respectively.
- There will be n pattern LMSs plugged into the voting machine where n denotes the number of patterns. The pattern LMSs can be switched to only “down”.
- The pattern LMSs are lockable in such a way that when a pattern LMS is switched to “down”, the voting machine can lock and unlock it during the voting process.
- The Finish Voting LMS has the same characteristics as the pattern LMSs.

In Figure 1, we give an example of our adjustable proposed machine for 16 candidate LMSs. There are 5 pattern LMSs.

4.2 Properties of Voting Machine

The proposed voting machine is similar to an ATM machine except that it has mechanical switches instead of keypad. Voter selects voting choice via these switches. A touch screen of the machine interacts with the voter like the ATMs does. The machine can also print receipt just like ATMs. More concretely, it has the following properties:

- Voting machine has a processor running an application handling all the voting procedures.
- The application has also cryptographic capabilities to perform asymmetric encryption and digital signatures with zero-knowledge proofs. The encryptions, proofs of partial knowledge and digital signatures are printed in QR Code, because in this way any party can easily verify the correctness by portable devices like smart phones.
- The voting machine has also a storage unit where all encrypted votes are recorded. It can also store security logs about the entire voting process and key management data of the voting machine (by the authority).
- It has a number of LMSs which can be switched two-way or one-way. Two-way switches are called candidate LMSs and their number is equal to number of candidates. One-way switches are pattern LMSs and Finish Voting LMS. As mentioned earlier, the machine can have more LMSs as plug-ins depending on the number of candidates. We also note that all LMSs can be locked by the voting machine during the voting procedure.
- For usability concerns, the voting machine has also some human-machine interfaces. First of all, voting machine has a touchscreen interface which gives instructions in order to guide the voters. Furthermore, the poll agents can authenticate themselves to the machine by using PIN and/or smart card (by adding a smart card slot) which may depend on an existing secure authentication method.
- A mini printer is embedded to the voting machine and generates a printed receipt for a voter. The receipt consists of an encrypted vote, generated challenge set, proof of knowledge that the encrypted ballot is one of the challenge set, and a signature.
- The machine has also an USB port interface which is only accessible to authorized poll agents for firmware update of the voting machine and export of the encrypted votes. After the polls are closed, the data on the storage unit can be transferred to a USB stick from the voting machine. The final destination of the encrypted votes are sent to voting authority which are later published on a secure WBB.

4.3 Web Bulletin Board & Tallying

WBB is used to verify whether the cast votes have been collected by the central server correctly by simply checking the existence of the receipts. If there are no complaints at this stage then the central server starts the counting process. After decrypting all votes by using the master decryption key which is distributed to several independent organizations or authorities (e.g., political parties, government official, and non-governmental organizations), the counting process is performed as usual in front of the independent auditors using for example mixnets [14,2,31] or homomorphic tallying ([36,2,19]).

5 Our Proposed Voting System

5.1 Informal Description of Our Proposal

The public/private key pair (pk, sk) of the election for encryption/decryption of the votes is generated through a distributed key generation protocol [38]. The pk is loaded to the application prior to the polling. The private key shares sk_i of sk are distributed to n independent parties. The application encrypts the vote using a homomorphic encryption algorithm (e.g., ElGamal) with pk. After the election is over, at least t parties, where t is less than n, gather and decrypt the tallying votes by using their own keys sk_i. Finally, they obtain the final outcome.

The voting process is briefly as follows: the voter switches up or down the LMS representing her preferred candidate. Switching up or down makes blue or red color visible on the switch background, respectively. The voting machine prints the encrypted vote on the receipt. We note that the receipt should be not teared off until the end of the process. Next, the voter selects either Finish Voting switch or one of the pattern LMSs randomly and switches it down. If she selects Finish Voting switch, she tears off the receipt and the voting procedure ends. If she selects one of the patterns, the procedure continues. Receiving the candidate switches' pattern, the voting machine then prints the plain form of the colors (blue or red) with corresponding candidates to the receipt in order. For example, for 4 candidates it can be printing (Candidate_1, Blue), (Candidate_2, Red), (Candidate_3, Blue), (Candidate_4, Red)). It also prints zero-knowledge proof of knowledge that the encrypted vote is one of the elements of the candidates with the corresponding colors. Considering the example, the ciphertext is one of the encryptions of the pairs above. Namely, our system provides a proof of knowledge to each voter (on her receipt) to assure that the submitted vote is correctly received by the voting machine.

As mentioned earlier, switching candidate and pattern LMS basically generates random challenges in order to prevent any malicious behavior of the voting machine. Since these challenges cannot be predicted in advance, the voting machine will not be able to attack the system without being detected with certain probability. We note that the proofs should be checked by the voter after the election, e.g., by simply using a software application on smart phone.

We are now ready to present our e-voting system. The significant phases of our protocol are as follows:

1. Identification and Authentication
2. First Challenge Generation of the Vote Casting Procedure and Encrypted Vote Generation
3. Finishing or Final Challenge Generation of the Vote Casting Procedure
4. Pattern, Proof and Signature Generation
5. Pattern Verification
6. Verification of the Proof
7. Inspection from the Bulletin Board
8. Vote Tallying

5.2 Identification and Authentication

Authentication is performed by the physical process used by the jurisdiction as in the conventional paper-based voting. In other words, a voter has to authenticate himself to the poll agent/voting authority in the polling station before he starts the voting procedure.

5.3 First Challenge Generation of the Vote Casting Procedure and Encrypted Vote Generation

Voter alone is allowed to access the voting machine. k candidate LMSs (k is the number of candidates) of the voting machine is active. Voter is now ready to cast her real intention from the voting machine (see Figure 2). The procedures are as follows.

1. The voter switches randomly up or down the LMS representing her preferred candidate.
2. The voting machine displays the chosen candidate represented by the LMS. For example as shown in Figure 3, the voter casts her vote for candidate Party 4 with switching down (or switching to Blue). Then, the voting machine will ask to cast.
   (a) If the voter touches the Cast button, the machine locks all candidate LMSs and encrypts the vote v where v denotes one of the candidate (and also its colored challenge $c \in \{0,1\}$ where 0 denotes the Red switch and 1 denotes the Blue switch) and prints it on the receipt, denoted as $E(v||c,r) = (g^r, g^{v||c}h^r)$ where r is the randomness (see Figure 4).
   (b) If the voter touches the Cancel button, the overall process will be stopped. In this case, voting process restarts.

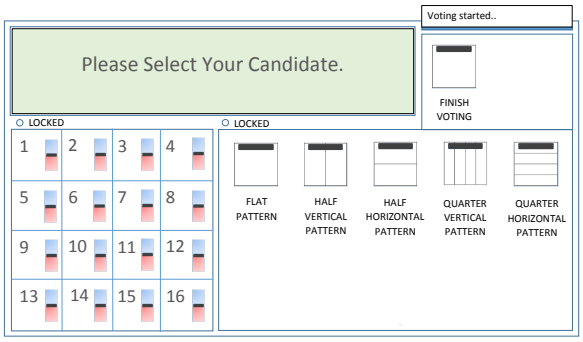

Fig. 2 Start casting your vote: Switch randomly up or down the LMS representing your preferred candidate.

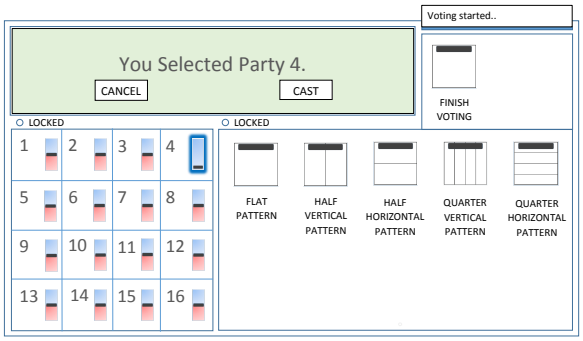

Fig. 3 Verify your vote and press the Cast button to continue on the screen.

5.4 Finishing or Final Challenge Generation of the Vote Casting Procedure

Having the encrypted vote on the receipt, voting procedure can be finished or remaining challenges can now be generated by switching a pattern LMS (see Figure 5). More concretely,

1. The voter decides not to challenge the voting machine and simply switches Finish Voting switch. Finishing the voting procedure in this step provides ease of use for the voter. As the voting machine does not know who will challenge it prior to this step and it has already printed the encrypted vote on the receipt, this shortcut step does not result any weakness. In fact, not every voter has to challenge the voting machine. To detect a possible malicious behavior, it is enough for a small number of voter to challenge the voting machine. Once Finish Voting is switched, Finish Voting and all pattern LMSs are locked. The printer of the voting machine prints Finish Voting

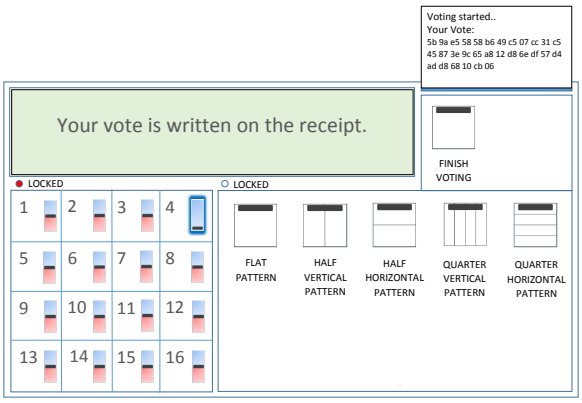

Fig. 4 Encrypted vote is printed on the receipt.

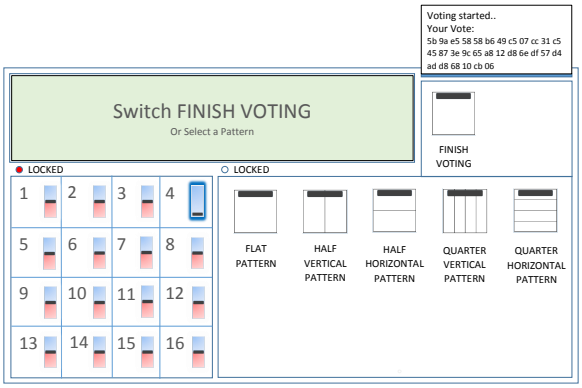

Fig. 5 Finish voting or challenge the voting machine by selecting a pattern.

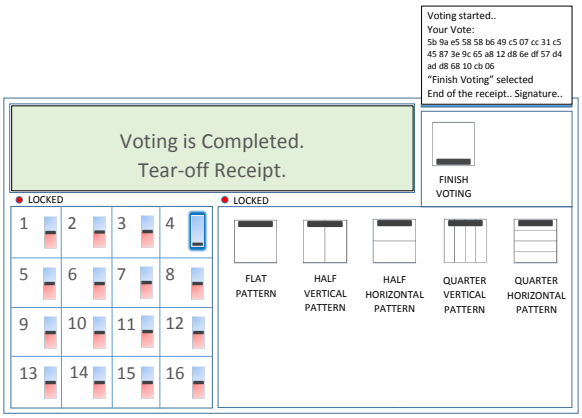

Fig. 6 Voting process is completed without challenging voting machine.

selected on the receipt (See Figure 6). All the information on the receipt is signed by the voting machine for later assurance that the receipt is offical. The vote will be stored to the database (DB) of the voting machine. The voter tears off the receipt. The voter takes the receipt and the voting process is completed. After the voter tears off the receipt, the machine unlocks all candidate and pattern LMSs, including Finish Voting LMS. When the voter leaves the device, she is free to restore the pulled switches, to pull switches of her choice, or to do nothing. The voting machine is now ready for the next voter. All the voter has to do is look for her receipt on the bulletin board as described in Section 5.8.

2. The voter switches one of the pattern LMSs which means the voter generates the challenge set of the remaining candidates on the candidate LMSs of the voting machine. For example as shown in Figure 7, the voter selects Quarter Vertical Pattern which means one column is Blue and one column is Red and so on. As the voter switches Candidate 4 to Blue, the candidates in the column where Candidate 4 is located are considered to have been switched to Blue and the colors of the candidates in the other columns are formed accordingly. If the voter had chosen Flat Pattern, all candidates would have been considered to have switched to the same color. In other words, since Candidate 4 was switched to Blue, the color of all candidates would be considered switched to Blue. Optionally, red and blue LEDs can be placed around the candidate LMSs. If the device illuminates the appropriate one of the LEDs placed around them in this step, it can visually show the voter how the pattern is formed. Once a pattern LMS is switched, all pattern and Finish Voting LMSs are locked (see Figure 7).

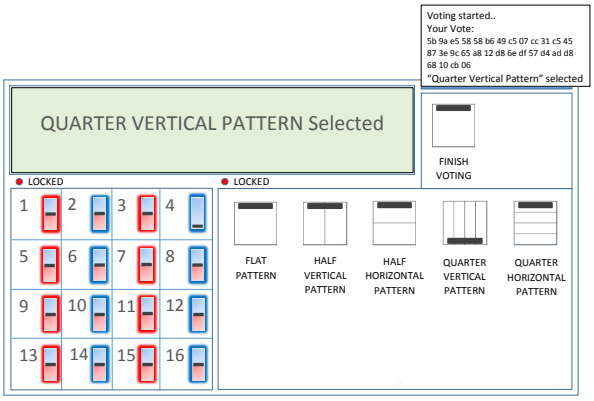

Fig. 7 Voting machine is challenged by selecting Quarter Vertical Pattern.

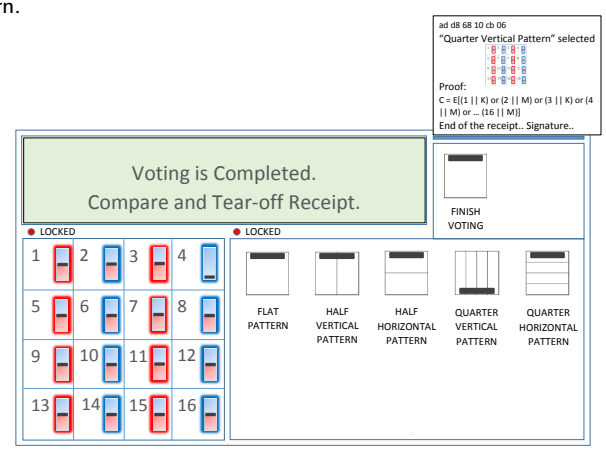

Fig. 8 Voting process is completed successfully. Verify the pattern on the receipt and take it.

5.5 Pattern, Proof and Signature Generation

1. The printer of the voting machine starts processing once one of the pattern LMSs is switched down. Namely, the printer will print the plain form of the challenge-pattern of all candidate LMSs to the receipt.
2. The voting machine generates a proof and prints it on the receipt (see Figure 8). The proof of knowledge shows that the encrypted vote is one of the challenge set on the receipt. At the end of the proof the voter is assured that the machine encrypted one of the pattern list (e.g., $\{(\text{Party } 1, \text{Red}), \cdots, (\text{Party } 16, \text{Blue})\}$). All the information on the receipt is signed by the voting machine for later assurance that the receipt is official. The vote will be stored to the database (DB) of the voting machine.

5.6 Pattern Verification

The voter now verifies that the colors on the candidate LMSs and on the receipt are exactly the same. Here, there are two possible scenarios:

1. If the voter does not observe any mismatch then he confirms the casting process by tearing off the receipt. Once the voter tears off the receipt, then she cannot claim any mismatch between LMSs and the receipt pattern. Thus, the voter will take the receipt and the voting process will be completed. The receipt is illustrated as in Figure 9. After the voter tears off the receipt, the machine unlocks all candidate and pattern LMSs, including Finish Voting LMS. When the voter leaves the device, she is free to restore the pulled switches, to pull the switches of her choice, or to do nothing. The voting machine is now ready for the next voter.
2. If there is indeed a mismatch, the voter does not tear off the receipt and asks poll agent. The poll agent checks

whether the colors of all candidate LMSs are indeed correctly printed on the receipt. If it is not the case, the poll agent will seal and remove the voting machine from use (i.e. malware affected voting machine). The voter may accidentally or maliciously claim that there is a mismatch. In this case, she does not tear off the receipt and asks the poll agent. The poll agent checks again and verifies whether the colors of candidate LMSs are the same as on the receipt. Since there is no mismatch, the dispute will be resolved and the voting procedure will continue (i.e. malicious or mistaken voter). Unfortunately, the poll agent will see the voter's voting choice in this case. Therefore, the voter should be careful in objection as its consequences are serious. She should only object when there is really a mismatch.

5.7 Verification of the Proof

1. Each voter can verify the proof on the receipt via some mathematical tools after casting the vote (e.g., running an open-source verification application on a smart phone which reads the proof from the receipt using QR codes). Note that voter may not want to verify but can also delegate the proof verification to the political parties (since it does not reveal any information about the votes except verifying about the correctness). This proof procedure can be also delegated to some third parties. If a proof of a receipt cannot be verified, the voter objects to the voting authority with the signed receipt.

5.8 Inspection from the Bulletin Board

1. Once election is finished, poll agents will sign all the receipts (qualified signature). Central administration of the High Election Board (HEB) has corresponding public keys in order to verify that the votes are received from the local poll agents.
2. Web bulletin board WBB is the last control point of our protocol. The DRE database which contains all receipt information is posted in the WBB after the polls are closed. WBB is publicly readable but nobody can modify the content of it. Voter verifies that her receipt is found among the receipts in the bulletin board. For the sake of finding receipt easily, voter can filter bulletin board based on DRE voting machine.

5.9 Vote Tallying

When the homomorphic tallying is applied, the encrypted votes are combined into a single encrypted tally which can be processed as in the existing schemes [36, 2, 19]. Only the final encrypted tally should be decrypted with the shareholders' (i.e., independent authorities) private keys corresponding to the public key in the DRE voting machine (by the underlying threshold encryption scheme). Note that efficient *mix-net* procedures can also be applied to break any correlation between voters and their votes [27, 22, 32, 9]. During the mix-net procedure, the shareholders individually randomize the encryptions using a reencryption mechanism by the underlying homomorphic properties, shuffle the reencrypted results and then prove that the input ciphertexts contain a shuffle of the output results. Finally, they cooperate to decrypt the incoming encrypted votes by each computing the partial decryption privately (with the zero-knowledge proofs).

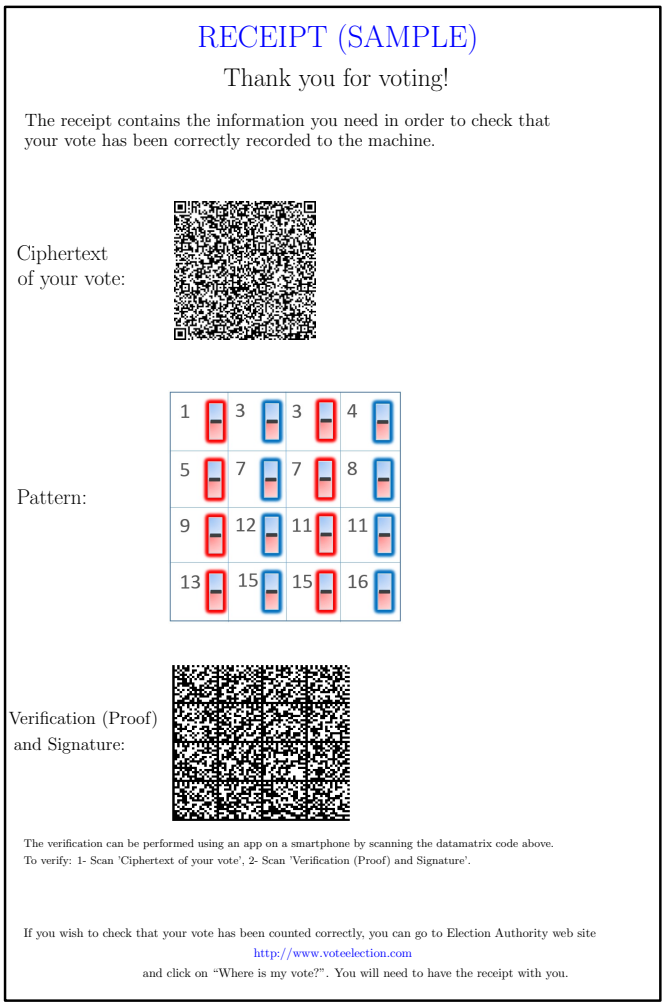

Fig. 9 A sample receipt.

6 Security Analysis

Now we are ready to show the security of our system. We assume that all the participating parties can be categorized as a threat, i.e., voters, voting machines, poll agents, and the

election authority. We show that a malicious party cannot do any malicious behavior without being detected. The main security requirements of e-voting protocols are privacy, verifiability, uncoercibility, receipt-freeness and accuracy. We note that there are also other security requirements of e-voting system to be assured like authentication and eligibility. They are independent procedures and are assumed to be performed as in the classical system, therefore they will be omitted in this work. Note also that each voter is authorized for one voting session at a time as in the classical paper-based system.

6.1 Correctness and Privacy

The number of the patterns on the machine is crucial for correctness and security of the system. If the number of patterns is less than it should be then the machine can predict the colors of some candidate LMSs which results in compromising the correctness. Therefore, in the next theorem we show that number of the patterns must be defined before the election according to the number of the candidates.

Theorem 1 *To prevent a malicious behavior of a corrupted voting machine, the patterns should be determined in such a way that when the voter selects her choice and its color, the color of each of the remaining candidates should not be predictable.*

Proof In our system, a voter switches up or down LMS representing her preferred candidate. Voting machine prints the encrypted vote on the receipt and then voter selects a pattern for remaining LMSs if she wants to challenge the voting machine. Switching a pattern results in assigning a color for each candidate LMS, and the (candidate, color) pairs are printed on receipt. OR-proof guarantees that encrypted vote is one of the (candidate, color) pairs. Thus, if a malicious voting machine encrypts an unintended candidate and cannot correctly guess the color of it, the voter can easily detect this by verifying the receipt.

The proof is constructed for a voting machine with 16 candidates (as in the Figure 1) and can be extended easily. If four patterns had been used in the voting machine instead of five patterns, the machine would easily fool the voter. Namely, if one pattern is absent in the voting machine then the voting machine can guess the color of one of the remaining candidates once the voter selects her voting choice and its color. For example, if the voting machine does not have Quarter Horizontal Pattern and voter selects Candidate 4 with Blue color as in Figure 3, no matter which pattern the voter chooses, Candidate 8's color will be Blue. In this case, the machine can generate an encrypted vote for Candidate 8 instead of Candidate 4 without being detected.

If five patterns are used on the machine as in the Figure 3, when the voter selects her voting choice, the color of each of the remaining candidates can be Blue or Red depending on the pattern selection (i.e. it is not certain). This statistically prevents malicious machines from cheating. However, the machine may attempt to cheat by taking risks. For example, voter selects Candidate 4 with Blue color as in Figure 3, but the machine generates and prints encrypted vote for Candidate 8 with Blue color. In this case, the probability of cheating detection is $\frac{1}{5}$ (i.e. cheating is detected only when the voter chooses Quarter Horizontal Pattern out of five patterns that assigns Red color to Candidate 8).

Table 1 Possibilities of detecting cheating machine when voter selects candidate 4.

1 $\frac{2}{5}$	2 $\frac{1}{5}$	3 $\frac{1}{5}$	4 -
5 $\frac{2}{5}$	6 $\frac{2}{5}$	7 $\frac{2}{5}$	8 $\frac{1}{5}$
9 $\frac{2}{5}$	10 $\frac{2}{5}$	11 $\frac{2}{5}$	12 $\frac{1}{5}$
13 $\frac{1}{5}$	14 $\frac{2}{5}$	15 $\frac{2}{5}$	16 $\frac{2}{5}$

When a voter selects Candidate 4 with Blue color as in Figure 3, the probabilities of detecting fraud (when the machine casts the vote for another candidate) are shown in Table 1 (the probability of being detected when the machine generates vote for Candidate 1 is $\frac{2}{5}$; for Candidate 2 is $\frac{1}{5}$; etc.). Note that the machine selects Red color for Candidate 5, 9, 13, 14, and 15 to decrease detection probability.

As shown in Table 1, the probability of detecting a cheating is $\frac{1}{5}$ or $\frac{2}{5}$. If we consider the minimum probability as $\frac{1}{5}$, then the probability that a malicious machine cannot be detected even at the 10th challenge is 10%, 1.1% at the 20th challenge, 0.12% at the 30th challenge.

The likelihood of selecting patterns may differ from each other (i.e., people may select some patterns more commonly). Accordingly, the above rates becomes 34%, 12% and 4.2% respectively if the probability of selecting the pattern that can detect cheating is considered to be 10%. This rate decreases exponentially as the number of challenges to a malicious machine increases.

Finally, the above proof applies to the machine supporting up to 16 candidates. If a machine supports fewer candidates, the number of patterns required will be less. For example, the number of patterns required on a machine supporting up to 4 candidates is 3.

Theorem 2 *The privacy of a voter is guaranteed.*

Proof Because identification and authentication are independently performed from the stage of casting votes, the voting process does not expose any information about identity of the voter. Secondly, no authority can obtain the private key of the election because it is securely shared between independent parties (using underlying threshold encryption scheme). To ensure complete anonymity, mixnet based tallying can be used where the votes are processed by a re-

randomization (also known as re-encryption) and a publicly verifiable mixnet. If the votes are anonymized securely by preventing any cheating behavior through mix-nets, then the independent parties, who hold the secret shared keys sk_i, cooperate to decrypt all ciphertexts. The final outcome is the list of all votes in plain form.

Theorem 3 *A malicious voting machine cannot fool the voter. Similarly, a malicious voter cannot fool the voting machine.*

Proof A malicious voting machine cannot simply start and end the voting process by itself because the final verification is performed manually by tearing the receipt off the printer to confirm the voting process has finished successfully. More concretely, the separation of the receipt from the printer system means that everything is run correctly, and the voter can stop the process at any time and can put an alarm until the voter tears the receipt off the printer.

In the case of an honest voting machine a malicious voter cannot fool the system or put an alarm because the receipt is only shown to the voter and is not separated from the printer. If the voter puts alarm before the tearing the receipt off the machine then the poll agent can see the receipt to verify whether the voter is indeed right. Otherwise, tearing the receipt off the printer prevents a malicious voter to put a wrong alarm.

6.2 Coercion, Vote-Selling, and Receipt-Freeness

We illustrated that the voter can verify her vote at all steps. In the proposed system anyone can check list of eligible voters and the signatures of the voting machine via QR codes (using OR-proofs). Since correctness of all processes can be investigated the proposed system satisfies the universal verifiability.

Theorem 4 *The proposed voting system is resistant to vote-selling and coercion.*

Proof Receipt-freeness ensures that voters cannot prove their election preference to a vote buyer. In our protocol, the vote is printed as encrypted form in the receipt and nobody can get any information from voter's receipt about the choice. More concretely, printed receipts leak no information about the identity of voters and their choices. Note that a receipt is composed of four parts: (1) an encrypted vote, (2) the colors of candidate LMSs (i.e. pattern), (3) OR-proof to verify the correctness of the encryption, and (4) Signature of the machine to all the data on the receipt. The voter can only verify the pattern on the receipt (by comparing it with the pattern on the machine) while she is at poll-site. Furthermore, anybody who verifies the proofs via QR codes can only learn whether the encrypted vote is the one of the pairs from pattern (i.e., the colors of the candidates). Therefore, nobody, even the voter, can learn additional information about the vote after the voting process has ended. Thus, vote-selling and coercion are not probable, and the proposed scheme has the property of receipt-freeness.

Remark 1 Although we allow only one vote one may argue that the proposed voting system is subject to the Italian attack. Note that the Italian attack considers the following scenario. Some coercers may force voters to cast a specific and unique order of candidates on the machine that could be uniquely identified with each other. Although the vote is privately cast during the voting process, the pattern of the votes could be revealed after the elections via a secure WBB, and the coercers can check the specific order whether the pattern exist or not. We would like to highlight that our system is robust against the Italian attack since the same limited patterns can be used to vote for different candidates. Besides, neither at the WBB nor on the receipt the information about the intention of the voter is shown as plain form. Thus, the pattern does not guarantee that a specific vote has been cast.

7 Conclusion

We proposed a new and secure DRE-based voting system (what we call TRVote). TRVote consists lockable mechanical switches (LMS), a touchscreen, and a printer which are widely used in a vending or an ATM machine. Furthermore, candidate LMSs can be placed and removed as plug-ins in the voting machine, which allows machine to support any desired number of candidates. TRVote assumes that the hardware and the software of the voting machine are assumed to be malicious. Our system is interesting in the sense that the voters are involved in order to challenge the voting machines. Namely, voters can independently challenge the voting machine and can verify the correctness of the votes using a printed receipt. We show that our proposal preserves security and privacy since no party including the manufacturer of the voting machine will be able to fool voters without being detected. The proposed system is also shown to be universally verifiable, secure against coercion and vote-selling.

References

1. Ben Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.
2. Ben Adida and Ronald L. Rivest. Scratch & vote: Self-contained paper-based cryptographic voting. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, pages 29–40. ACM, 2006.
3. Samiran Bag, Muhammad Ajmal Azad, and Feng Hao. E2e verifiable borda count voting system without tallying authorities. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–9, 2019.
4. Josh Benaloh. Simple verifiable elections. *EVT*, 6:5–5, 2006.

5. Josh Benaloh. Administrative and public verifiability: can we have both? *EVT*, 8:1–10, 2008.
6. Josh Benaloh, Michael D. Byrne, Bryce Eakin, Philip T. Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark, Dan S. Wallach, Gail Fisher, Julian Montoya, Michelle Parker, and Michael Winn. Star-vote: A secure, transparent, auditable, and reliable voting system. In *2013 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '13, Washington, D.C., USA, August 12-13, 2013*, 2013.
7. Josh Benaloh, Ronald Rivest, Peter YA Ryan, Philip Stark, Vanessa Teague, and Poorvi Vora. End-to-end verifiability. *arXiv preprint arXiv:1504.03778*, 2015.
8. Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In *International Conference on E-Voting and Identity*, pages 111–124. Springer, 2007.
9. Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. *Public-Key Cryptography – PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 – March 1, 2013. Proceedings*, chapter Verifiable Elections That Scale for Free, pages 479–496. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
10. D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *Security Privacy, IEEE*, 2(1):38–47, Jan 2004.
11. David Chaum. Surevote: technical overview. In *Proceedings of the Workshop on Trustworthy Elections, WOTE*, 2001.
12. David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L Rivest, Peter YA Ryan, Emily Shen, and Alan T Sherman. Scantegrity ii: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes. *EVT*, 8:1–13, 2008.
13. David Chaum, PeterY.A. Ryan, and Steve Schneider. A practical voter-verifiable election scheme. In SabrinadeCapitani di Vimercati, Paul Syverson, and Dieter Gollmann, editors, *Computer Security – ESORICS 2005*, volume 3679 of *Lecture Notes in Computer Science*, pages 118–139. Springer Berlin Heidelberg, 2005.
14. David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981.
15. Michael R Clarkson, Stephen Chong, and Andrew C Myers. Civitas: Toward a secure voting system. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 354–368. IEEE, 2008.
16. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '94, pages 174–187. Springer-Verlag, 1994.
17. Chris Culnane, Peter Y. A. Ryan, Steve A. Schneider, and Vanessa Teague. vvote: A verifiable voting system. *ACM Trans. Inf. Syst. Secur.*, 18(1):3, 2015.
18. Chaum D., Essex A., Carback R., Clark J., Popoveniuc Stefan, Sherman A., and Vora P. Scantegrity: End-to-end voter-verifiable optical- scan voting. *Security Privacy, IEEE*, 6(3):40–46, May 2008.
19. Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a university president using open-audit voting: Analysis of real-world use of helios. In *2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '09, Montreal, Canada, August 10-11, 2009*, 2009.
20. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *IEEE*, 4-31, pages 469–472. Transactions on Information Theory., 1985.
21. Kevin Fisher, Richard Carback, and Alan T Sherman. Punchscan: Introduction and system definition of a high-integrity election system. In *Proceedings of Workshop on Trustworthy Elections*, pages 19–29, 2006.
22. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. *Journal of Cryptology*, 23(4):546–579, 2010.
23. Feng Hao, Dylan Clarke, Brian Randell, and Siamak F Shahandashti. Verifiable classroom voting in practice. *IEEE Security & Privacy*, 16(1):72–81, 2018.
24. Feng Hao, Matthew N Kreeger, Brian Randell, Dylan Clarke, Siamak F Shahandashti, and Peter Hyun-Jeen Lee. Every vote counts: Ensuring integrity in large-scale electronic voting. In *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14)*, 2014.
25. Feng Hao and Matthew Nicolas Kreeger. Every vote counts: Ensuring integrity in large-scale dre-based electronic voting. *IACR Cryptology ePrint Archive*, 2010:452, 2010.
26. James Heather, Peter YA Ryan, and Vanessa Teague. Pretty good democracy for more expressive voting schemes. In *European Symposium on Research in Computer Security*, pages 405–423. Springer, 2010.
27. Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, Berkeley, CA, USA, 2002. USENIX Association.
28. Hugo Jonker, Sjouke Mauw, and Jun Pang. Privacy and verifiability in voting systems: Methods, developments and trends. *Computer Science Review*, 10:1–30, 2013.
29. J Kelsey, Andrew Regenscheid, T Moran, and D Chaum. Hacking paper: Some random attacks on paper-based e2e systems. *Frontiers of Electronic Voting*, 2007.
30. John Kelsey, Andrew Regenscheid, Tal Moran, and David Chaum. Attacking paper-based e2e voting systems. In *Towards Trustworthy Elections*, pages 370–387, Berlin, Heidelberg, 2010. Springer-Verlag.
31. Shahram Khazaei, Björn Terelius, and Douglas Wikström. Cryptanalysis of a universally verifiable efficient re-encryption mixnet. In *Proceedings of the 2012 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'12, pages 7–7. USENIX Association, 2012.
32. Shahram Khazaei, Björn Terelius, and Douglas Wikström. Cryptanalysis of a universally verifiable efficient re-encryption mixnet. In *Proceedings of the 2012 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'12, Berkeley, CA, USA, 2012.
33. Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Proving coercion-resistance of scantegrity ii. In *International Conference on Information and Communications Security*, pages 281–295. Springer, 2010.
34. Ralf Kusters, Tomasz Truderung, and Andreas Vogt. Clash attacks on the verifiability of e-voting systems. In *2012 IEEE Symposium on Security and Privacy*, pages 395–409. IEEE, 2012.
35. C Andrew Neff. Practical high certainty intent verification for encrypted votes, 2004.
36. Kun Peng, Riza Aditya, Colin Boyd, Ed Dawson, and Byoungcheon Lee. Multiplicative homomorphic e-voting. In *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, pages 61–72, 2004.
37. Olivier Pereira and Dan S Wallach. Clash attacks and the star-vote system. In *International Joint Conference on Electronic Voting*, pages 228–247. Springer, 2017.
38. Shafi Goldwasser Ran Canetti. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *LNCS*, 1592, pages 90–106. Springer-Verlag., 1999.
39. Ronald L Rivest and Warren D Smith. Three voting protocols: Threeballot, vav, and twin. *USENIX/ACCURATE Electronic Voting Technology (EVT 2007)*, 2007.
40. Alon Rosen, A Ta-shma, B Riva, and JY Ben-Nun. Wombat voting system, 2011.

41. Peter YA Ryan and Steve A Schneider. Prêt à voter with re-encryption mixes. In *European Symposium on Research in Computer Security*, pages 313–326. Springer, 2006.
42. Daniel Sandler, Kyle Derr, and Dan S Wallach. Votebox: A tamper-evident, verifiable electronic voting system. In *USENIX Security Symposium*, volume 4, page 87, 2008.
43. Siamak F Shahandashti and Feng Hao. Dre-ip: a verifiable e-voting scheme without tallying authorities. In *European Symposium on Research in Computer Security*, pages 223–240. Springer, 2016.