TRVote: A New, Trustworthy and Robust Electronic Voting System

Fatih Tiryakioğlu and Mehmet Sabir Kiraz and Fatih Birinci and Mehmet Karahan
{fatih.tiryakioglu, mehmet.kiraz, fatih.birinci, mehmet.karahan}@tubitak.gov.tr

TÜBİTAK BİLGEM, Turkey

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 a transparent coverage, a touchscreen, color recognition boxes and a printer. In this system, each candidate is represented by a color recognition box which is equipped within the voting machine. The machine has a flexible structure in the sense that the candidate boxes 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

1 Introduction

Conventional election systems consist of a voter manually marking the paper ballot, casting it manually to the voting box, and then the ballot being manually counted by the election officials. Despite several security and usability issues, the conventional mechanisms are believed to be mature, robust, and have been used for a long time. With the developments in the computer science and technology,

electronic devices have become an indispensable part of our life. Voting systems also begin to benefit from these technological developments. Electronic voting (e-voting) system is one of the most interesting application of cryptography. As a system, it brings many advantages both for voters and administrations, including speed, accessibility, accuracy, convenience, flexibility, and mobility. Furthermore, voting systems could become more convenient in terms of casting votes, and tallying process. In particular, tallying process could take less time with e-voting and it reduces the risk of human error when compared with traditional paper-based elections. E-voting systems, compared to the traditional ones, could also reduce costs in the long-term period. While e-voting could bring significant advantages it also would raise some new concerns. In particular, it is a well-known fact that the main concern of e-voting is the reliability of the system and privacy of the voters. Other security concerns include coercion, vote selling-buying, and incorrect tallying process. A well-defined voting protocol should satisfy all these requirements, however designing a robust system satisfying all the security requirements is a challenging task. The main difficulty is basically due to the fact that two opposite requirements, transparency and anonymity, are demanded together.

E-voting can be classified into two distinct systems: voting machines and Internet voting. Voting machines are basically specialized or standard equipments placed at polling stations. They typically have an interface for the voters to cast their vote easily and to process the ballots. On the other hand, Internet voting do not need physical polling stations. Namely, voters may cast their votes using their own devices such as computers, smart devices (phones and tablets). In this paper, we focus on voting machines, especially direct recording electronic based voting machines.

Poll-site computerized voting systems have so called Direct-Recording Electronic (DRE) devices situated at polling places, which are similar to ATMs. They allow voters to view ballots on a screen and to cast their ballots directly through the machines by means of buttons or a touch-screen. The votes are recorded directly onto computer memory devices. DRE voting machines have been in use since 1990s all around the world especially developed and developing countries such as USA, Australia, Russia, and Brazil [1]. However, it still keeps its ambiguity in some countries which is implemented, planned to implement or abandoned completely. Unfortunately, new deficiencies of the voting machines make the concept more controversial. Usability is also one of the most crucial aspect in order to increase the reliability and trust level in the society. In addition to these kinds of security requirements, there are also some possible active attacks to the voting machines. For example, an attacker who gets physical access to a machine could install a malicious code. In this way, an attacker could steal votes undetectably, modifying all records, logs, and counters. Mitigating such kinds of security demands and threats require precise election procedures and modifications to the dedicated voting machine's hardware/software.

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

voting systems in the last years. End-to-end verifiable systems mainly seek to ensure a voter that her vote was "counted-as-cast" without disclosing voter-vote connection. In the receipt-based systems, the voter is given a receipt to check whether it is stored to the central voting station after the voting process. The receipt does not reveal any information about the voting choices, and after the polls close all the voter receipts are made public. After the elections, the voters check whether their votes are stored to the central system securely and are guaranteed that they are tallied accurately. If the receipts do not exist in the public records, it will be perceived as corruption and voter will use her receipt for objection.

1.1 Contributions

The main contribution of this paper is to provide a new machine-based and reliable e-voting system, which we call TRVote. The voting machine has coin boxes for each candidate together with a transparent coverage, a touchscreen, Color Recognition Boxes 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 inserts one of Black/White coins into the preferred candidate box. Receiving voting choice and challenge, voting machine should encrypt the intended voting choice with the challenge, not one of the unintended candidates which is proven by the zero-knowledge proof printed on the receipt. Any party can easily verify the correctness of the proof by devices like smartphones or tablets via QR codes. After the encrypted vote is printed on the receipt, the voter challenges the remaining boxes with Black/White coins randomly. Finally, the voting machine proves that the encrypted vote (for the preferred candidate) is one of elements of the voter's challenge set, i.e. (candidate, coin) pairs.

The security and reliability of our system is statistically ensured even if the software running on the machine is corrupted. More concretely, we show that a voter can immediately notice unexpected activity in the system or a malicious behavior of a corrupted machine by using her challenges and her receipt. Furthermore, our system is also resistant to coercion and vote-selling because nobody can guarantee the voter to vote to 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 that the encryption is performed correctly (while revealing no information about votes). More concretely, the receipt only shows the encryption of vote, the pattern of the inserted coins, zero-knowledge proof showing that the encryption is one of the patterns, and the signature of the machine. The votes are verified on a secure Web Bulletin Board (WBB) which assures that the votes are recorded and transmitted to the central server correctly. The tallying process is handled as in usual way where the decryption key is securely generated and distributed to independent parties.

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 capabilities and forms a flexible structure where candidate boxes 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.

1.2 Organization

The rest of the paper is organized as follows. Related works about machine based 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

The idea of using technology in elections dated back to the beginning of 19th century [30]. In US, DRE voting machines have been used since the 20th century. The first DRE voting machine was employed in the polling stations in 1970s [37]. If we look at to the recent history, we see that touch screen DRE machines are used somewhat abundant throughout many countries, especially in US. In 2004, a survey led to the conclusion that in US 30.75% of all registered voters used e-voting systems [10], up from 7.7% in 1996 [16]. DRE voting machines are also used in Brazil, Venezuela, India and the Netherlands [1].

It is hard to ensure the security of dedicated e-voting machines such as Nedap and Diebold. In such schemes, it is crucial to keep the voting machine safely and ensure chain of custody not only during the elections but also during the storage. The 2000 election debacle in Florida with confusing butterfly ballots, dangling chads, and contested recounts motivated researchers to examine secure end-to-end verifiable voting systems. Prêt à Voter [14], Punchscan [23], VoteHere's Mark Pledge [35, 36], Votegrity [13], ThreeBallot [40], Voter Initiated Auditing [8], Scantegrity [19], Scantegrity II [20], STAR-Vote [9], and many more [3,4,7,24,29] can be given as some examples of DRE machines. Most of these systems do not need special equipments providing important security requirements. However, it is still very difficult to construct a reliable system while ensuring security and privacy completely [6, 25, 27, 30, 42, 43]. In general, they provide three kind of verification mechanisms; "cast-as-intended", "recorded-as-cast", and "tallied-as-recorded". Voters can verify that their intentions are recorded correctly via a receipt and the WBB, and observers can verify the tallying, without destroying the security of the election (i.e., anonymity, coercion, and vote selling).

Despite these desired security properties some attacks are listed in literature. Scantegrity is open to the randomization attack, but Scantegrity II seems to eliminate this problem. In [44] and [31] the security of The ThreeBallot Voting System and Prêt à Voter and Punchscan are examined. The ThreeBallot scheme has theoretical importance that it is possible to design a verifiable voting scheme

satisfying privacy without cryptography. With various countermeasures and procedural changes Prêt à Voter and Punchscan are secure against most threats. But, there are some possible attacks on these voting systems in the procedure of preparation of the ballots, in voting procedure and in casting, verification and auditing processes. In our TRVote system, to prevent these kinds of procedural attacks, the receipt, which contains encrypted ballot and list of candidates, is created during voting procedure and it reveals no information about choice of voter.

Verifiable voting systems are studied deeply in the academic literature, but they have not been practiced in governmental elections until 2009. In 2009, Scantegrity II has been used successfully in municipal election of Takoma Park, Maryland [11]. In 2014, vVote, which is a development of Prêt à Voter voting system designed by Culnane et al. [18], has been deployed in state election in Victoria, Australia. Differently from the election of Takoma Park which is a single-candidate selection, in Victoria, voters choose at least five candidates and list them according to the order of preference.

Transparency, verification and usability are the most critical aspects of voting system to be approved by society. In the election of the state of Victoria, voters in London, which use their votes at the Australian High Commission, according to a survey over 75 % agreed that the system was easy to use [41]. Acemyan et al. [2] use a methodology to identify user errors and test the casting and verification steps and observed that a big percentage of participants could not verify their votes. The most frequent reason is that they did not cast a ballot during voting process. According to another analysis on receipt checking [34], the majority of the voters who make verification did not object to the results.

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 Threshold Homomorphic Public-Key Cryptosystem

In this part, we briefly describe the underlying cryptographic primitives of the proposed voting system. Let $m \in \mathcal{M}$ denote its plaintext space, $c \in \mathcal{C}$ the ciphertext space, and $r \in \mathcal{R}$ 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 [22] 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 [5, 15, 32].

3.2 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 [17] 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.

4 Components of Our System

4.1 Properties of Color Recognition Boxes and Colored Coins as Challenges

We are going to design a new type of recognition box which we call *Color Recognition Box* (CRB). It is similar to a card reader box except that it can detect the color of the coins. CRBs can be easily plugged and locked into the voting machine depending on the number of candidates.

In the proposed system, there are only two colored coins, Black and White. These coins will be basically used as a physical challenge set of a voter. More concretely, CRB has the following structures:

- There will be k CRBs plugged into the voting machine where k denotes the number of candidates. Each CRB will represent a candidate (denote CRB_i for the i-th candidate $\mathsf{Candidate}_i$).

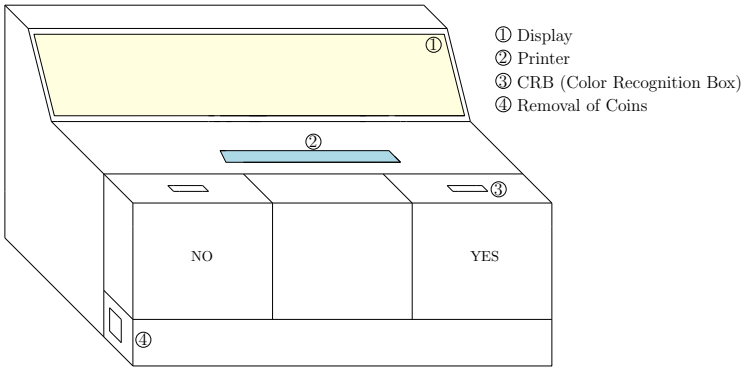

Fig. 1. Proposed adjustable voting machine with 2 candidates (e.g., referendum)

- The CRBs are transparent in such a way that the coins inside the CRB will be visible to the voter during the voting process.
- CRB is designed in such a way that only one coin should be inserted into it and the coin cannot be taken out from the CRB until either the vote has been cast and the voter leaved the machine or the voting process is canceled by the voter.
- Once the vote has been cast, the coins cannot be released from the CRB. Only authorized poll agents are eligible to take the coins out of the voting machine.

In Figure 1, we give an example of our adjustable proposed machine for two CRBs (i.e., candidates).

Remark 1. Although random behavior of a voter is realized by CRBs and colored coins, different mechanical setups can also be implemented. For example, for each candidate two special buttons simulating Black and White coins can also be used. The machine is equipped with a mechanical control in such a way that when the voter generates a challenge (either Black or White), the challenge cannot be changed by the voter or by the machine.

4.2 Properties of Voting Machine

The proposed voting machine is similar to a travel-ticket machine in which you can insert money or a bank/credit card to get a ticket for a journey. Briefly, the machine first asks you about the travel details and subsequently for payment. Next, it stores the money safely, checks whether it is valid and sufficient for the journey, and finally returns the ticket to you with a receipt. Working principle of our dedicated voting machine is very much likely to the travel-ticket, vending, or ATM machine. Namely, our voting machine can detect color of the coins and can keep them safely but in a transparent way so that the voter can see them. More concretely, it has the following properties:

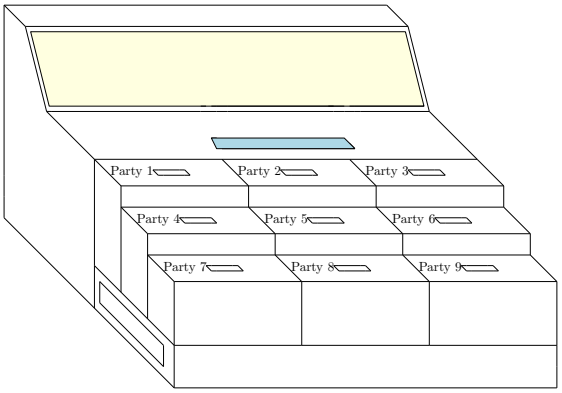

Fig. 2. Proposed adjustable voting machine with 9 candidates.

- 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 CRBs in which the coins are inserted (number of CRBs is equal to number of candidates). As mentioned earlier, the machine can have more CRBs as plug-ins depending on the number of candidates. We also note that coins are kept inside the CRBs until the voting procedure of a voter is finished, and they can only be taken out by the authorized poll agents. This assures that a voter can cast only one vote.
- 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 the update firmware of the voting machine and for the 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.

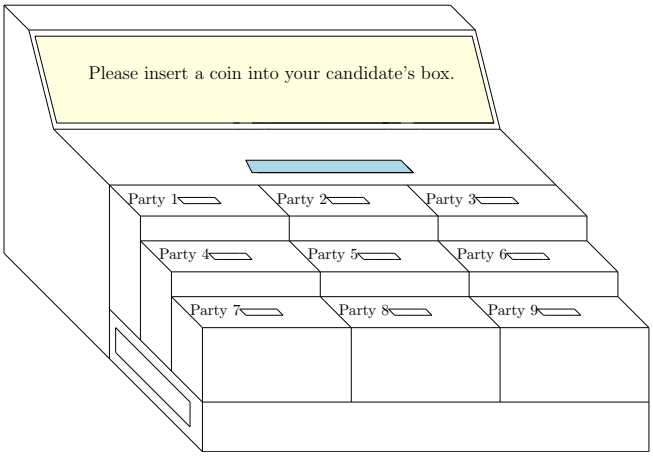

Fig. 3. Start casting your vote: Choose Black or White coin randomly and insert it into your candidate box.

The final destination of the encrypted votes are sent to voting authority which are later published on a secure WBB.

See Figure 2 for an illustration of the proposed voting machine with 9 candidates.

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 [5, 15, 32] or homomorphic tallying ([5, 21, 38]).

5 Our Proposed Voting System

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 [39]. 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.

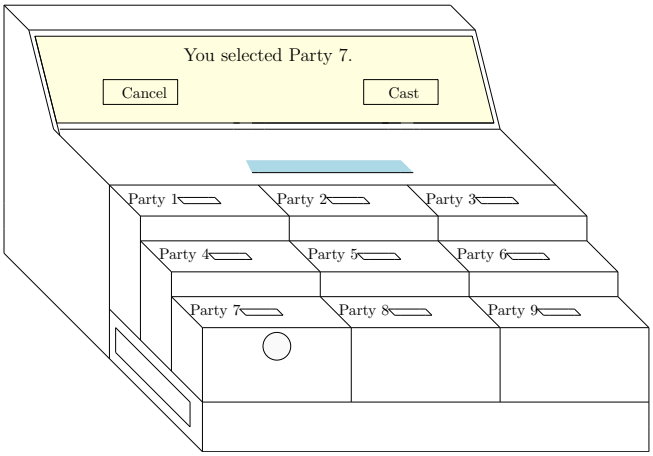

Fig. 4. Verify your vote and press the Cast button to continue on the screen.

The voting process is briefly as follows: the voter inserts a random coin (either Black or White) into the CRB representing her preferred candidate. 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 inserts the other coins (either Black or White) into the rest of the CRBs randomly. The voting machine then prints the plain form of the colored coins with corresponding candidates (to the receipt) in order (e.g., for 4 candidates it prints (Candidate_1, Black), (Candidate_2, White), (Candidate_3, Black), (Candidate_4, White)). It also prints zero-knowledge proof of knowledge that the encrypted vote is one of the pattern of the candidates with the corresponding coins. 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, colored coins basically generate 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 with certain probability without being detected. 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. Identication and Authentication
2. First Challenge Generation of the Vote Casting Procedure and Encrypted Vote Generation
3. Final Challenge Generation of the Vote Casting Procedure
4. Pattern, Proof and Signature Generation
5. Pattern Verification
6. Verification of the Proof and Inspection from the Bulletin Board
7. Vote Tallying

5.1 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.

Once the authentication is successful, the poll agent gives k colored coins to the voter for k candidates where $k \neq 2$, $k \neq 3$, and $\lfloor k/2 \rfloor$ of them are Black and $\lceil k/2 \rceil$ of them are White. If $k = 2$ or $k = 3$ then the voter is given 4 coins where 2 of them are Black and 2 of them are White. This exception is necessary to protect privacy of voters (see Section 6 to see the details.).

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

Voter alone is allowed to access the voting machine. k CRBs (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 as follows.

1. The voter selects a coin randomly (Black or White) and inserts it into her candidate's CRB (see Figure 3). The voter can see but cannot interfere the inserted coin in the CRB.
2. The voting machine senses the coin and its color via the CRB, and displays the chosen candidate represented by the CRB. For example as shown in Figure 4, the voter casts her vote for candidate Party 7 with White coin. Then, the voting machine will ask to cast.
   (a) If the voter touches the Cast button, the machine encrypts the vote v where v denotes one of the candidate (and also its colored challenge $c \in \{0, 1\}$ where 0 denotes the White coin and 1 denotes the Black coin) 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 5).
   (b) If the voter touches the Cancel button, the overall process will be stopped. In this case, the machine returns the coin to the voter and voting process restarts.

5.3 Final Challenge Generation of the Vote Casting Procedure

Having the encrypted vote on the receipt, remaining challenges can now be generated by inserting all other coins into the voting machine in random order. More concretely,

1. At this step, the voter is requested to insert all of the rest of the coins randomly into the other CRBs of voting machine which means the voter generates the challenge set of the remaining candidates on the CRBs of the voting machine (see Figure 6).

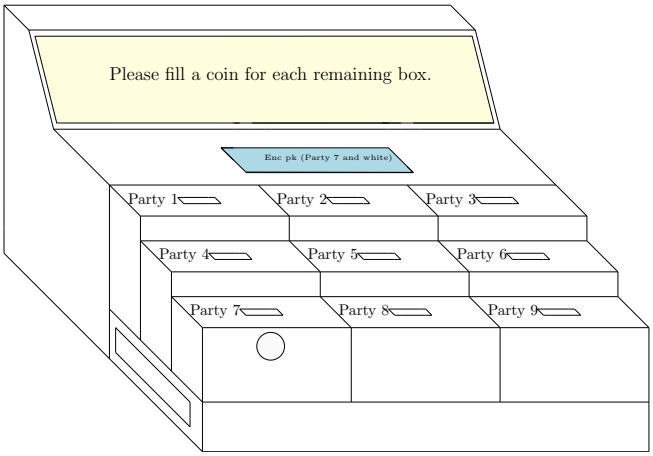

Fig. 5. Challenge the voting machine: Insert Black or White coins into all other candidate boxes in random order.

5.4 Pattern, Proof and Signature Generation

1. The printer of the voting machine starts processing once all the coins are inserted into the machine. Namely, the printer will print the plain form of the challenge-pattern of all inserted coins to the receipt.
2. The voting machine generates a proof and prints it on the receipt. 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., $\{(\text{Party } 1, \mathsf{White}), \cdots, (\text{Party } 9, \mathsf{White})\}$. All the information on the receipt is signed by the voting machine for later assurance that the receipt is official (see Figure 7). The vote will be stored to the database (DB) of the voting machine.

5.5 Pattern Verification

The voter now verifies that the pattern on the CRBs 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 he cannot claim any mismatch between machine CRBs 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 8. After the voter leaves the voting machine, the poll agent comes to the voting machine, authenticates himself in order to take all the coins which can be used by other voters repeatedly (which can be also performed using a mechanical remote controller). The voting machine is now ready for the next voter.

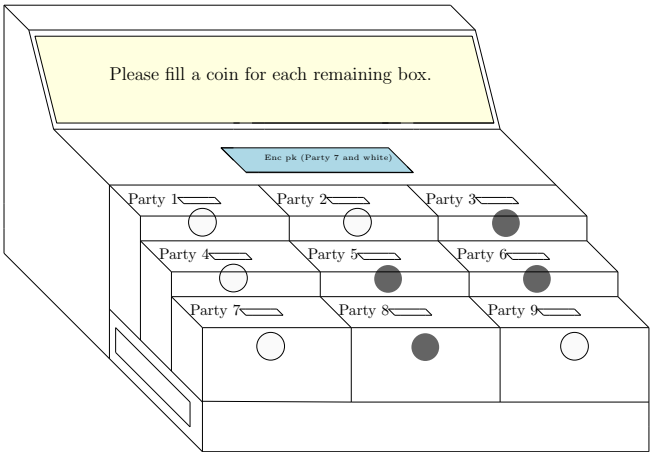

Fig. 6. Inserting Black or White coins into all other candidate boxes is finished.

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 pattern of all the inserted coins is indeed correctly printed on the receipt. If it is not the case, the poll agent will seal and remove the voting machine from use. Note that the voter can mistakenly 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 inserted coin pattern is the same as on the receipt. Since there is no mismatch, the voting procedure will continue. Note that the poll agent will get no information about the vote since all the coins were already inserted into the machine and the pattern is independent from the chosen vote.

5.6 Verification of the Proof and Inspection from the Bulletin Board

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.
2. 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 form the local poll agents.

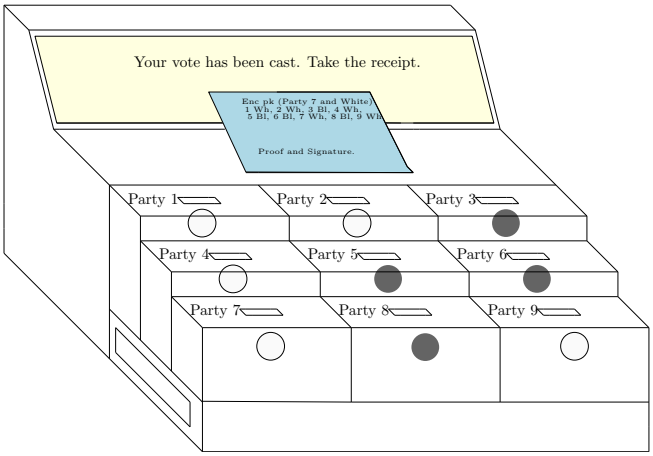

Fig. 7. Voting process is completed successfully. Verify the pattern on the receipt and take it.

3. 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.

5.7 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 [5, 21,38]. 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 [12, 26, 28, 33]. 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).

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,

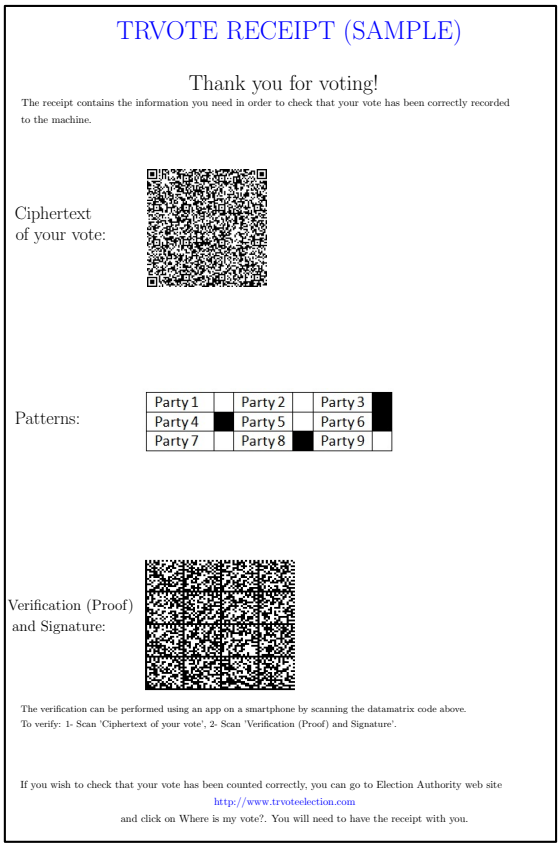

Fig. 8. An example of a receipt.

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. We further highlight that limiting each voter to cast only one ballot is achieved by keeping the coins inside the machine.

6.1 Correctness and Privacy

The number and the colors of the coins given to the voter are crucial for correctness and security of the system. If colored coins, which are equal to the number of candidates, are randomly chosen by the voter then the machine can predict

the order of the coins (namely, colors of coins) in the CRBs which compromises the correctness. Therefore, in the next theorem we show that number and color of the coins 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;*

(i) If there are $k = 2$ candidates then the voter must be given exactly four coins in such a way that two of them are Black *and the other two are* White.
(ii) If there are $k = 3$ candidates then again, the voter must be given exactly four coins in such a way that two of them are Black *and the other two are* White.
(iii) If there are $k \geq 4$ candidates, then the voter must be given exactly k coins in such a way that $\lfloor k/2 \rfloor$ of them are Black *and the other $\lceil k/2 \rceil$ are* White.

Proof. In our system, a voter inserts a random coin into CRB representing her preferred candidate. Voting machine prints the encrypted vote on the receipt and then voter inserts other coins randomly into the rest of the CRBs. OR-proof only guarantees that encrypted vote is one of the vote in the pattern on the receipt. Thus, if a malicious voting machine cannot correctly guess the color of coin, the voter can easily detect by verifying the receipt. The proof is given for each case separately as follows:

(i) For the case of two candidates ($k = 2$): If two coins with the same color are given to the voter, the machine can easily fool the voter. Namely, if two coins, where one is Black and the other is White, are given to the voter then the machine can know the color of the remaining coin once the voter inserts the first one. If three coins are given to the voter (without loss of generality, one is Black and the other two are White), then the machine will know the color of the remaining coins if the voter inserts the Black coin first. On the other hand, if the voter is given two Black and two White coins, the machine cannot predict the color of the second coin since the voter still holds either two Blacks and one White or one Black and two White coins. Therefore, the cheating probability of the machine would be $2/3$ and become $(2/3)^l$ for l voters (e.g., it becomes negligible for $l = 80$.). Hence, a corrupted machine cannot cheat without being detected if two Black coins and two White coins are given to the voter for $k = 2$.
(ii) For the case of three candidates ($k = 3$): Similarly, if three coins are given to the voter (without loss of generality, one is Black and the other two is White) then the machine will know the color of the remaining coins in case the voter inserts the Black coin first to give her vote. Instead of this, if two Black and two White coins are given to the voter then the machine cannot predict the order of the following coins after the voter inserts the first coin. Hence, a malicious machine cannot cheat the voter if two Black coins and two White coins are given to the voter for $k = 3$. Similarly, by using the same arguments in (i), the cheating probability would be again negligible.
(iii) For the case of k candidates ($k \geq 4$): k coins, with $\lfloor k/2 \rfloor$ of them are Black and $\lceil k/2 \rceil$ of them are White, will be sufficient to prevent malicious behavior

of machine. Indeed, assume without loss of generality that voter inserts one of Black coins first. Then, there will be $(\lfloor k/2 \rfloor - 1) \geq 1$ Black coins for final challenge. By using the same arguments in (i), the cheating probability would be again negligible.

□

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

Proof sketch. 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 sketch. 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 (note that the coins in the boxes and the receipt does not reveal the voter's choice). 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 sketch. 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) inserted coin 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 pair from pattern (i.e., the color of the party). 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 2. 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 patterns can be used to vote for different candidates. More concretely, the critical point of our work is that the first coin is the real vote and all others are the fake ones, namely for adding randomization. Neither at the WBB nor on the receipt the information about the first coin is shown as plain form. Thus, the pattern does not guarantee that a specific vote has been cast at the first step.

7 Conclusion

We presented a new and secure DRE-based voting system (what we call TRVote). TRVote consists a transparent coverage, a touchscreen, simple colorimeter (color recognition) and a printer which are widely used in a vending or an ATM machine. Furthermore, candidate boxes 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. The main drawback of the proposed machine is to handle many candidates. In that case, it may not be user-friendly and therefore, it is interesting to propose a more friendly solution to support sufficiently large candidates.

References

1. E-Voting World Map 2015. `http://www.e-voting.cc/wp-content/uploads/2012/03/e-voting_worldmap_2015.pdf`. Accessed: 2016-03-23.
2. Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. From error to error: Why voters could not cast a ballot and verify their vote with helios, prêt à voter, and scantegrity ii. *USENIX Journal of Election Technology and Systems (JETS)*, (2):1–25, 2015.
3. Ben Adida and C. Andrew Neff. Ballot casting assurance. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop*, EVT'06, pages 7–7, Berkeley, CA, USA, 2006. USENIX Association.
4. Ben Adida and C. Andrew Neff. Efficient receipt-free ballot casting resistant to covert channels. In *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'09, pages 11–11. USENIX Association, 2009.
5. 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.
6. J. Bannet, D. W. Price, A. Rudys, J. Singer, and D. S. Wallach. Hack-a-vote: Security issues with electronic voting systems. *IEEE Security Privacy*, 2(1):32–37, Jan 2004.
7. Susan Bell, Josh Benaloh, Michael D. Byrne, Dana Debeauvoir, Bryce Eakin, Philip 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., August 2013. USENIX Association.
8. Josh Benaloh. Ballot casting assurance via voter-initiated poll station auditing. In *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology*, EVT'07, pages 14–14, Berkeley, CA, USA, 2007. USENIX Association.
9. 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.
10. Kimball W. Brace. Overview of voting equipment usage in united states, direct recording electronic (dre) voting, May 5, 2004.
11. Richard Carback, David Chaum, Jeremy Clark, John Conway, Aleksander Essex, Paul S. Herrnson, Travis Mayberry, Stefan Popoveniuc, Ronald L. Rivest, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity II municipal election at takoma park: The first E2E binding governmental election with ballot privacy. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 291–306, 2010.

12. 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.
13. D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *Security Privacy, IEEE*, 2(1):38–47, Jan 2004.
14. 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.
15. David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981.
16. United States Federal Election Committee. Direct recording electronic information.
17. 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.
18. 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.
19. 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.
20. Chaum D., Carback R.T., Clark J., Essex A., Popoveniuc Stefan, Rivest R.L., Ryan P.Y.A., Shen E., Sherman A.T., and Vora P. L. 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.
21. 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.
22. 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.
23. Aleks Essex, Jeremy Clark, Richard Carback, and Stefan Popoveniuc. The punchscan voting system: Vocomp competition submission. In *the First University Voting Systems Competition (VoComp)*. Springer Berlin Heidelberg, 2007.
24. Sarah P. Everett. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection*. PhD thesis, Rice University, HOUSTON, TEXAS, 5 2007.
25. Ryan Gardner, Sujata Garera, and Aviel D. Rubin. On the difficulty of validating voting machine software with software. In *Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology*, EVT'07, pages 11–11, Berkeley, CA, USA, 2007. USENIX Association.
26. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. *Journal of Cryptology*, 23(4):546–579, 2010.
27. Joseph Lorenzo Hall. Transparency and access to source code in electronic voting. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop*, EVT'06, pages 8–8, Berkeley, CA, USA, 2006. USENIX Association.

28. 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.
29. Rui Joaquim and Carlos Ribeiro. *E-Voting and Identity: Third International Conference, VoteID 2011, Tallinn, Estonia, September 28-30, 2011, Revised Selected Papers*, chapter An Efficient and Highly Sound Voter Verification Technique and Its Implementation, pages 104–121. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
30. Douglas W. Jones. Early requirements for mechanical voting systems. In *First International Workshop on Requirements Engineering for e-Voting Systems, RE-VOTE 2009, Atlanta, Georgia, USA, August 31, 2009*, pages 1–8. IEEE, 2009.
31. 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.
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, pages 7–7. USENIX Association, 2012.
33. 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.
34. Ester Moher, Jeremy Clark, and Aleksander Essex. Diffusion of voter responsibility: Potential failings in e2e voter receipt checking. *USENIX Journal of Election Technology and Systems (JETS)*, 1(3):1–17, 2014.
35. C. Andrew Neff. Verifiable mixing (shuffling) of elgamal pairs. Technical report, In proceedings of PET '03, LNCS series, 2003.
36. C. Andrew Neff. Practical high certainty intent verification for encrypted votes, 2004.
37. Kun Peng. Theory and practice of secure e-voting systems. In *Theory and Practice of Cryptography Solutions for Secure Information Systems*, pages 428–459, 2013.
38. 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.
39. 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.
40. Ronald L. Rivest. The threeballot voting system, 2006.
41. Peter Y. A. Ryan, Steve A. Schneider, and Vanessa Teague. End-to-end verifiability in voting systems, from theory to practice. *IEEE Security & Privacy*, 13(3):59–62, 2015.
42. Naveen Sastry, Tadayoshi Kohno, and David Wagner. Designing voting machines for verification. In *Fifteenth USENIX Security Symposium (USENIX Security 2006)*, August 2006.
43. Naveen K. Sastry. *Verifying Security Properties in Electronic Voting Machines*. PhD thesis, Berkeley, CA, USA, 2007.
44. T. Tjostheim, T. Peacock, P. Ryan, and University of Newcastle upon Tyne. Computing Science. *A Case Study in System-based Analysis: The ThreeBallot Voting*

System and Prêt à Voter. Technical report series. University of Newcastle upon Tyne, Computing Science, 2007.