Receipt-Freeness and Coercion Resistance in Remote E-Voting Systems

Yefeng Ruan

Department of Computer and Information Science,
Indiana University Purdue University Indianapolis,
Indianapolis, IN, USA
E-mail: yefruan@iupui.edu

Xukai Zou

Department of Computer and Information Science,
Indiana University Purdue University Indianapolis,
Indianapolis, IN, USA
E-mail: xzou@iupui.edu

Abstract: Remote Electronic Voting (E-voting) is a more convenient and efficient methodology when compared with traditional voting systems. It allows voters to vote for candidates remotely, however, remote E-voting systems have not yet been widely deployed in practical elections due to several potential security issues, such as vote-privacy, robustness and verifiability. Attackers' targets can be either voting machines or voters. In this paper, we mainly focus on three important security properties related to voters: receipt-freeness, vote-selling resistance, and voter-coercion resistance. In such scenarios, voters are willing or forced to cooperate with attackers. We provide a survey of existing remote E-voting systems, to see whether or not they are able to satisfy these three properties to avoid corresponding attacks. Furthermore, we identify and summarize what mechanisms they use in order to satisfy these three security properties.

Keywords: Remote E-voting systems; Receipt-Freeness; Vote-Selling; Coercion Resistance.

Reference to this paper should be made as follows: Yefeng Ruan and Xukai Zou (xxxx) 'Receipt-Freeness and Coercion Resistance in Remote E-Voting Systems', *International Journal of Security and Networks*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Yefeng Ruan is a current PhD student in Computer and Information Science Department of Indiana University Purdue University Indianapolis. His research area is social network analysis and network security.

Xukai Zou received the PhD in computer science from University of Nebraska-Lincoln. He is an Associate Professor in the Department of Computer and Information Science at Indiana University Purdue University Indianapolis. His research interests include Applied Cryptography, Communication Networks and Security.

1 Introduction

An election is a significant event in many democratic countries, however, traditional voting systems are often not efficient and convenient enough given the large number of areas and population involved in modern elections. Also, in some cases traditional voting systems have design flaws, for example the confusing "butterfly ballot" in the United States presidential election in Florida, 2000 [1] [2]. Faced with these shortcomings of traditional paper based voting systems, many research works have explored the possibility of using E-voting systems in real

This is the author's manuscript of the article published in final edited form as:
Ruan, Y., & Zou, X. (2017). Receipt-freeness and coercion resistance in remote E-voting systems. International Journal of Security and Networks, 12(2), 120–133. https://doi.org/10.1504/IJSN.2017.083836

elections [3] [4] [5] [6] [7] [8]. Based on the locations where E-voting systems are used, they can be divided into three categories: poll, info-kiosk and remote E-voting systems [9]. In the former two categories, authorities still have at least partial physical control of elections, while in remote E-voting systems the processes are not physically supervised by authorities [10]. As [11] pointed out, remote E-voting systems allow people to cast their votes over the Internet, from home, or any other location as long as they have access to Internet. In this paper, we mainly focus on remote E-voting systems. Compared with traditional paper based voting systems, remote E-voting systems provide voters a convenient and efficient approach. There are a few countries, including Estonia and Switzerland, that currently have the option (or in partial areas) of casting ballots through Internet in their elections [12].

Remote E-voting systems have not yet been widely deployed in practical elections due to several potential security issues, e.g. accountability and verifiability [13]. For example, Tadayoshi Kohno et al. analyzed an E-voting system in [14], and they concluded that there are still several potential security issues which are previously undetected in this system. Like traditional paper based voting systems, remote E-voting systems should satisfy several very basic requirements in order to be acceptable in elections, for example, correctness, verifiability, privacy and so on [15]. apart from these requirements, remote E-voting systems pose some new challenges which traditional paper based voting systems do not, for example, software flaws, man-in-middle attacks and denial of services [16].

One of the main challenges that prevent remote E-voting systems from being practically deployed is that some of their requirements or properties are incompatible and difficult to be implemented simultaneously [17] [18]. Such incompatibility also provides attackers opportunities. As elections are very sensitive political topics in many countries, they become popular targets for malicious attackers.

In our previous paper [19], we classified remote E-voting systems into four categories based on how they achieve vote-privacy. Although we mentioned receipt-freeness, the focus of interest of [19] is to provide a general survey for existing remote E-voting systems' primary cryptographic techniques: mix-nets, blind signature, threshold homomorphic encryption and secret sharing. In this paper, we specifically focus on three important security properties related to voters: receipt-freeness, vote-selling resistance and voter-coercion resistance. In such scenarios, voters are willing or forced to cooperate with attackers, which makes the elections very complicated [15] [20] [21].

The rest of this paper is organized as follows: we introduce the general voting flow and background of remote E-voting systems in Section 2. Definition and features of three voter related security properties are described in Section 3, as well as some protection measures that defend against corresponding attacks in Section 4. We then examine several existing remote E-voting systems in Section 5. In Section 6, we compare these systems from different perspectives. Finally, we conclude this paper in Section 7.

2 Remote E-voting systems

2.1 An overview of a remote E-voting system's voting flow

There are many proposed remote E-voting systems [8] [10] [22] [12] [23] [24]. Although these systems have some slight differences, most of them have a general voting flow. In remote E-voting systems, involved entities include voters, candidates, registrars (or authorities), tallies, bulletin board, and potentially other additional components in some systems such as auditors. Candidates' information should be available to all the voters. The main role of registrars is to authenticate voters' eligibility. Tallies are responsible for collecting and verifying ballots, and finally tabulating the results. From the bulletin board, all the observers can see the manipulation of ballots in the elections. A general remote E-voting voting flow includes the following four steps [9]:

- Setup or initialization. In this stage, systems should be initialized and make necessary information available. For example, authorities (or registrars) publish candidates' information and instructions of how to vote for candidates on bulletin board. Many remote E-voting systems encrypt voters' ballots (e.g. ElGamal [25], RSA [10] and so on) in order to achieve vote-privacy and confidentiality, therefore authorities or tallies need to publish encryption information, like public keys which will be used in the voting stage. For those systems which need to decrypt encrypted ballots, tallies hold the private keys and keep them secret.
- Registration. Before casting votes for candidates, voters firstly have to be

authenticated by registrars for their eligibility. Registrars will check voters' eligibility and record eligible voters. At the same time eligible voters will obtain necessary information, e.g. tokens, credentials or encrypted ballots, from registrars which will be used during the voting process. In order to be more robust, many systems usually have multiple registrars working together, rather than having a single registrar.

- Voting. In this stage, voters choose their intended candidate and include them in their ballots. To achieve vote-privacy, most remote E-voting systems encrypt ballots. Voters' evidences of their eligibility should be included in order for tallies to further verify ballots' eligibility. In systems using credentials, voters will include them in their ballots; voters can also use blind signature to show their eligibility [26]. Additionally, some systems also contain proof to prove that they follow predefined protocols [25].

- Tallying. In this stage, tallies verify and validate ballots, count them and then publish the results. Counting policies can be defined by the administrators or governments in different elections. Moreover, many systems provide audiences or auditors with proof that can prove the correctness of elections.

As we described above, Figure 1 shows the general voting flow of remote E-voting systems. Apart from these four steps, some remote E-voting systems may contain some additional steps for various specific security purposes. For example, encrypted ballots are anonymized using mix-nets in [10] before being tallied.

2.2 Security properties of remote E-voting systems

As we stated before, remote E-voting systems should satisfy certain requirements before they can be widely deployed in important elections. Remote E-voting systems, as voting systems, are supposed to have basic functional properties, such as correctness, democracy, fairness, accountability and transparency [19] [18] [15]. However, in this paper we mainly focus on the following properties related to security.

- Privacy [27]. Privacy in remote E-voting systems contains two aspects [18]. First of all, voters' personal information, such as social security number or biometrics, should not revealed to any one else. Secondly, voters' ballots, including their choices, should also be kept unrevealed. In remote E-voting systems, even authorities and tallies should not have any knowledge of voters' choices.

- Verifiability [28]. Verifiability is an important property to guarantee the correctness of elections; it can be divided into two categories: universal verifiability and individual verifiability. Universal verifiability requires that the correctness of elections' results can be verified by all the observers. To clearly define individual verifiability, we further divide it into two subcategories: weak individual verifiability and strong verifiability. Weak individual verifiability allows voters to verify if their ballots reach authorities or bulletin board and are included in the tallying stage; however, in most remote E-voting systems ballots are encrypted. In other words, voters may not be able to verify their votes, which include their candidate choices. Thus, individual verifiability in [26] and [25] actually refers to weak individual verifiability. On the other hand, in addition to weak individual verifiability's properties, strong individual verifiability also allows voters to verify their own candidate choices (i.e., their votes in clear or plain format) and to verify that their votes are accurately counted in tallied results. For example in [8], voters can not only verify their cast ballots (each ballot is a mixed value containing a part of one's own vote and the partial values of other voters' votes) but also use their secret location information to verify for whom their votes are casted (i.e., their true vote) and to visually verify that their votes are counted in the final tally. Unfortunately this is not receipt-free. It is obvious, though, that strong individual verifiability provide more verification than weak individual verifiability.

- Robustness or reliability [29]. Robustness here refers to systems' vulnerabilities to attacks. In the case that some registrars, tallies or voters are corrupted by attackers, remote E-voting systems should still be able to work correctly.

- Receipt-freeness [27]. There is no way (or receipt from voting machines) for voters to prove to attackers that they voted in a certain way even when they are willing to do so. Receipt-freeness can prohibit voters from

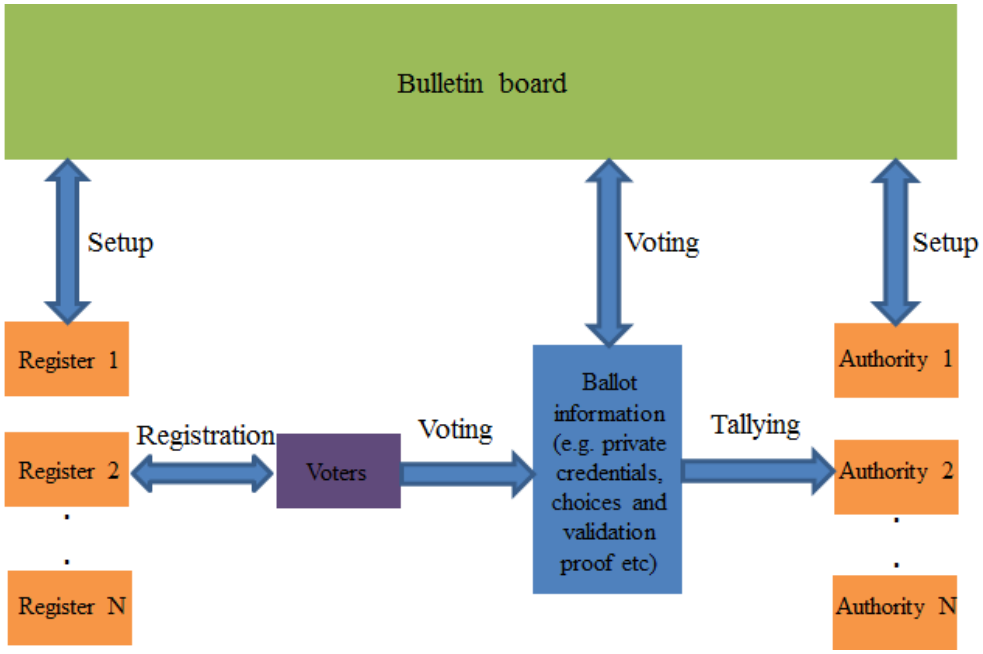

Figure 1: General voting flow of remote E-voting systems

proving to others their choices. Therefore it is used as a common method to mitigate vote-selling [30].

- Vote-selling resistance [30]. In some cases, voters are willing to sell their votes to buyers and follow buyers' instructions. However, in order to convince buyers, they have to prove that they vote in the way the buyers desire. Sellers have two ways of accomplishing this: directly selling their credentials (or identities) to buyers, or providing buyers with proof of vote, e.g. receipt [31] [32].

- Voter-coercion resistance [12]. Voters should be able to cast their ballots as they intended, even when they are forced by coercers to vote for other candidates. At the same time coerced voters should be able to disguise themselves without letting coercers know. This assumes that voters and attackers can communicate with each other in the voting process [15].

In the following of this paper, we will specifically focus on the last three properties. We will see how remote E-voting systems can satisfy these three properties in order to defend against corresponding attacks.

2.3 Cryptography primitives

In this section, we briefly introduce some cryptography primitives which are widely used in remote E-voting systems. More complete details can be found in [33].

2.3.1 Blind signature (BS)

Blind signature was proposed by David Chaum in 1983 [34]. Before the sender sends her/his message to the signer, she/he blinds or disguises her/his message with some random factors. In such a way, the signer signs the sender's message but without knowing its content. In remote E-voting systems, this is used for voters to obtain signatures from authorities without leaking their votes [26]. Blind signature can be implemented by very common public key cryptosystems, such as RSA.

2.3.2 Threshold encryption and decryption/ secret sharing

Threshold encryption and decryption are widely used in many applications in order to avoid single point failure. In such cases, encryption and decryption are conducted by authorities together [35]. Agents always use secret sharing [36] to share keys. Many remote E-voting systems use this mechanism to perform registration and decryption distributively [10] [12] [23] [25]. This can prevent

attackers from breaking into systems, unless they control the majority of authorities involved.

2.3.3 Homomorphic encryption and re-encryption

Homomorphic encryption enables a system to do some specific encryption of plain-texts, without exposing plain-texts [37]. For example, $Encrypt(m_1 \oplus m_2) = Encrypt(m_1) \otimes Encrypt(m_2)$. In this case, plain-text m_1 and m_2 are not revealed, although we know the encryption of $m_1 \oplus m_2$. This feature is used to achieve anonymization in remote E-voting systems [10], to add up ballots without leaking votes [10] [12] [25]. In addition, by using the properties of homomorphic encryption, it is possible to construct another cipher-text $c^{'}$, given cipher-text c, for the same plain-text m. This is called re-encryption of plain-text m.

2.3.4 Mix-nets

Mix-nets M is another very important and widely used cryptography primitive in remote E-voting systems [38]. It takes a sequence of inputs, which can be encrypted ballots generated in the voting stage; each agent in M will perform some secure and random permutations on their inputs and output permuted results to the next agent [39]. In this case, the order of the inputs will be shuffled, thus maintaining voters' anonymity in remote E-voting systems.

2.3.5 Plain-text equivalence test (PET)

Given two cipher-texts, authorities are able to make a judgment as to whether two underlying plain-texts are the same or not, without revealing two plain-texts [38]. It can be used to help tallies verify valid ballots [25].

2.3.6 Zero-knowledge proof (ZKF)

Zero-knowledge proof is one of the most commonly used methods for security protocols, as well as for remote E-voting systems. It was first introduced in [40]. Zero-knowledge proof allows the prover to convince the verifiers that a given statement is true or false without revealing any additional information. It is essential for interactive protocols, such as universal verifiability and coercion-evidence in remote E-voting systems [25].

2.3.7 Designated verifier proof

For knowledge proof, some can be verified by many users; however, in some cases, we want to make sure that such proof are only verifiable to the specific users. Designated verifier proof is designed only for intended verifiers; except the designated verifiers, no one else is able to obtain conviction, even if the verifiers give their private information to others [41].

3 Properties: receipt-freeness, vote-selling resistance and voter-coercion resistance

3.1 Receipt-freeness

Receipt-freeness, first introduced in [27] and [30], is considered a very important property for remote E-voting systems and many research works have been dedicated to it [7] [23] [42] [43]. It means that voters are not able to provide any proof of their voting strategies to others. Voting machines (including also registrars and tallies) do not provide voters receipts, but also voters themselves are not able to construct such type of proof. As we stated above, receipt-freeness is essential for remote E-voting systems to prevent vote-selling and voter-coercion.

3.2 Vote-selling resistance

Vote-selling refers to scenarios in which voters, for some reason, are willing to sell their votes to buyers. This process includes following buyers' instructions and communicating with buyers during the voting process [25]. Traditionally, vote-selling is considered illegal and is prohibited in elections.

Voters typically have two ways to sell their ballots [30]. They can directly sell their credentials or tokens to buyers; in such cases, buyers can the cast the compromised ballots by themselves using the sellers' identities. From tallies' or registrars' points of view, it is impossible to identify whether or not such ballots are cast by real, eligible voters or buyers, as they use the same credentials. Apart from this, without selling private credentials, sellers can cast for candidates following buyers' instructions.

However, there exists a challenge for sellers in both scenarios. How can sellers convince buyers that they are doing what buyers pay them to do? In the first case, how do buyers know whether or not sellers give them real credentials? If they are real, how can buyers make sure that sellers will not re-submit ballots [9]? Similarly, how can sellers prove to buyers that they have followed buyers'

instructions, and do not make any changes after that? Without such proof, sellers cannot convince buyers. This can provide opportunity for a remote E-voting system to prevent vote-selling.

3.3 Voter-coercion resistance

In elections, it is possible for voters to be coerced by attackers to follow their instructions. In such cases, attackers can disrupt the elections or even control the elections' results. Therefore, to make remote E-voting systems fair and robust, they should be voter-coercion resistant [43].

Voter-coercion resistance requires remote E-voting systems to allow coerced voters still be able to cast for their intended candidates as if they were not coerced, and without being detected by attackers. However, as pointed out in [15], it has no well-agreed formal definition for the concept of voter-coercion resistance; it is usually only stated in natural language. Juels et al. proposed a computational definition in [12], and authors in [44] analyzed voter-coercion resistance in an epistemic approach. [25] divided voter-coercion into two categories: explicit coercion and silent coercion, where silent coercion also takes credential leakage into account. Based on observational equivalence, [15] defined voter-coercion resistance as:

> "Coercion-resistance is formalized as an observational equivalence too. In the case of coercion-resistance, the attacker (which we may also call the coercer) is assumed to communicate with Alice during the protocol, and can prepare messages which she should send during the election process. This gives the coercer much more power. [15]"

Again receipt-freeness is necessary to defeat voter-coercion. If voters have ways to prove their voting strategies to a third party, then attackers can also force voters to prove to them their choices. We see several different measures in literature intended to defeat voter-coercion [10] [12] [45] [46].

[12] also described three types of attacks – Randomization attack, Forced-abstention attack and Simulation attack, depending on coercers' purposes in remote E-voting systems, and some of attackers' capabilities are described in [10]. Note that even when receipt-freeness is achieved, these attacks are still possible.

3.4 Relation among receipt-freeness, vote-selling resistance, and voter-coercion resistance

Receipt-freeness is an important property for remote E-voting systems, particularly, for vote-selling resistance and voter-coercion resistance in order to counter the corresponding two types of attacks. They are similar and interrelated, and their definitions have evolved as more research works appeared in this field. For example receipt-freeness originally was used to denote the security property including both receipt-freeness and voter-coercion resistance in [31]. While in [43], [15] and [12] authors differentiated them and suggested that voter-coercion is a stronger concept than receipt-freeness. Here we illustrate the relation among the three properties in Figure 2.

Among these three properties, receipt-freeness serves as the most basic and necessary requirement for vote-selling resistance and voter-coercion resistance [15] [12]. If systems are not receipt-free, voters are given or are able to construct receipts which can prove to others for whom they cast their votes. In such scenarios, voters can convince buyers, or coercers can force coerced voters, to show them receipts. Therefore, without receipt-freeness, it is impossible for systems to achieve vote-selling resistance and voter-coercion resistance [27]. Of course, in order to achieve vote-selling resistance and voter-coercion resistance, several additional assumptions (or policies) are needed. For example systems should allow voters to revote, attackers cannot physically monitor voters' behaviors [30] [10], and so on.

Besides showing buyers receipts, sellers have another way to convince buyers: directly selling private information, e.g. credentials or tokens. Therefore, in order to prevent voters from selling their votes, in addition to receipt-freeness, systems should also prohibit selling credentials or tokens. This can be implemented through some policies or hardware factors. For example, associating credentials with identities or biometrics, such as social security numbers and finger-prints [47], may make it more difficult to sell credentials. Conversely, the introduction of such sensitive information into remote E-voting systems creates other privacy and security concerns [48]. Another way to prevent a seller from selling credentials is to make the proof of correctness of credentials non-transferable to buyers [30].

Voter-coercion resistance, as stated in [12] and [15], is a stronger and broader concept than receipt-

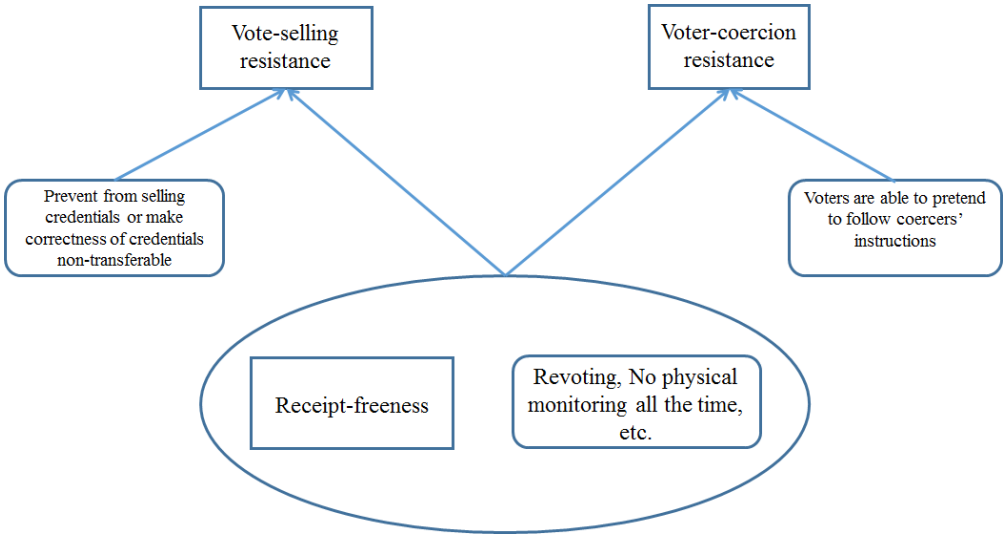

Figure 2: Relation among receipt-freeness, vote-selling resistance and voter-coercion resistance

freeness. We can see that receipt-freeness itself is not enough to defend against randomization attack, forced-abstention attack and simulation attack as defined in [10] [21]. In voter-coercion resistant systems, coerced voters should be able to pretend to follow coercers' instructions. In other words, given the public results from bulletin board, coercers are not able to find whether or not coerced voters disobey their instructions [25]. Voter-coercion is a tough problem, and as pointed out in [49], remote E-voting systems are inherently coercible. Some researchers have even proposed voter-coercion with modified meaning: coercion-evidence [25].

3.5 Trust assumptions

As we have seen already, in order to satisfy these security properties, many systems make assumptions, such as attackers cannot physically monitor voters' behaviors. Here we are mainly concerned two types of assumptions: channel assumptions and agent assumptions. Channel assumptions describe which kinds of channels are needed in remote E-voting systems. Agent assumptions refer to the notion that in order for systems to work correctly, there is a maximum threshold of how many agents, including authorities, voters, registrars, can be corrupted by attackers.

3.5.1 Channel trust assumptions

There are two possible trust assumptions regarding channels in remote E-voting systems: anonymous channels and untappable channels [9] [26].

- Anonymous channel. Anonymous channel is also called untraceable channel and was firstly proposed in [39] using mix-net technique. It is designed for anonymous communication between the sender and the receiver. The receiver receives messages from the sender through anonymous channel, but the receiver cannot identify the sender's identity. Therefore, if an anonymous channel is eavesdropped by attackers, they cannot figure out the sources of ballots. As indicated in the full version of [12], this can be achieved by use of anonymization, such as mixing, homomorphic encryption and so on. An anonymous channel is essential in receipt-free and voter-coercion resistant remote E-voting systems [15] [12].

- Untappable channel. Untappable channel is more difficult to achieve in reality than anonymous channel. Apart from anonymity, it also requires that messages in untappable channels cannot be revealed to attackers. In other words, it is a perfect secure channel that will never leak any messages [42]. Unfortunately, in practical remote E-voting systems, untappable channel usually cannot be guaranteed or realized. It is the known weakest physical assumption in receipt-free remote E-voting systems [50].

3.5.2 Agent trust assumptions

Like many other applications, remote E-voting systems also make some trust assumptions about

participants or agents [25]. For example, in some systems none of the agents, like registrars and tallies, can be corrupted by attackers in order to maintain systems' normal operations; other systems make assumptions that attackers can compromise part of the agents, and the systems can still work correctly. In the following discussions, we divide this into two categories: non-corrupted assumption and minor (or threshold) corrupted assumption. Obviously, non-corrupted is a more strict assumption than minor-corrupted. Usually systems requiring non-corruptability are not as robust as those requiring minor-corrupted requirements.

4 Protection measures

In this section, we introduce the attributes of receipt-freeness, vote-selling resistance and voter-coercion resistance, as well as some protection measures in order to satisfy them. In the later section, we will see whether or not existing remote E-voting systems satisfy them, and how to achieve them.

4.1 Protection measures for receipt-freeness

In order to achieve receipt-freeness, as pointed out in [31], there are several requirements that remote E-voting systems have to satisfy.

- Private ballots. Voters' ballots should be encrypted and attackers cannot get access to plain-texts, for example attackers cannot physically witness remote terminals [30]. This can be achieved by using encryption in the voting stage.
- Secret decryption key. For those systems which have decryption keys to decrypt every single encrypted ballot, voters should not have any knowledge of the decryption keys in order to prevent them from providing decrypted information to attackers. For robustness, many remote E-voting systems distribute decryption keys among multiple tallies [10] [21] [25] [50].
- Randomness. Randomness contains two parts: randomness in voters' encrypted ballots and ballots' random orders [31]. In many cryptosystems, plain-text is usually encrypted with a random number, e.g. RSA. As shown in [50], voters can use random numbers to generate receipts. Therefore, voters should have no knowledge about random numbers, and the ballots should be encrypted by a third party. To convince voters that their votes are correctly encrypted, the third party should provide voters proof. In addition to that, in order to prevent voters from transferring proof to attackers, proof should be non-transferable [50]. In remote E-voting systems, zero-knowledge proof and designated verifier proof are among the most commonly used non-transferable proof [41]. In order to achieve anonymity, encrypted ballots need to be randomized before they are sent to the bulletin board. This randomization can be achieved by using randomization techniques, e.g. mix-net. Similarly, this randomness information must be unrevealed to voters and tallies.

4.2 Protection measures for vote-selling resistance

As we stated above, sellers can either sell their credentials to buyers or cast ballots following buyers' instructions [30]. Therefore, in order to prevent vote-selling, remote E-voting systems also need to focus on these two possibilities.

The idea to defend against these two selling strategies is the same: make the system function such that sellers cannot convince buyers of their votes. For users' ballots, as long as systems satisfy receipt-freeness, sellers cannot provide proof to buyers of their actual votes [51] [24] [50]. Similarly for credentials, if sellers cannot prove to buyers that the given credentials are real, buyers will not pay them, so proof of credentials' correctness should be non-transferable [25] [30]. In other words, proof can only be verified by voters themselves. Also systems can make credentials associated with voters' biometrics such that sellers may not be willing to sell their biometrics [47]; however this type of defense mechanism is out of this paper's scope.

4.3 Protection measures for voter-coercion resistance

To protect systems from voter-coercion attacks, several measures have been proposed in recent years.

- Explicit notification. This is a straightforward and intuitive method. In some cases, when voters are coerced, they can use another separate channel to send an explicit coercion proof to registrars or tallies to indicate that they are coerced [52]. This method requires

there exist an untappable channel that can be used to send notifications.

- Fake credential. In the case that voter is coerced, she/he can generate a fake credential and give it to the coercer or use the fake credential to cast for the candidates the coercer has indicated [10] [12] [45] [46]. In such cases, a coercer cannot distinguish fake credentials from real credentials, and only the majority part of tallies together can detect these invalid credentials contained in the ballots [12].
- Panic password. This measure is very similar to that of fake credential. Panic password systems can generate two types of passwords: panic password and real password [53]. When voters log onto systems to cast ballots, using either panic passwords or real passwords, systems will not promote error message to users when users enter panic passwords; however, systems can distinguish panic passwords from real passwords, and only with a real password can users be identified in the later tallying stages. In other words, although attackers holding panic passwords can go through the whole process as real voters, systems will not count their ballots. In voter-coercion cases, voters can give panic passwords to coercers and at the same time keep their real passwords secret [26].
- Coercion-evidence. Using this measure, the suspiciously coerced ballots are recorded. In this case, instead of using fake credentials to deceive coercers, the systems ask coerced voters to submit their ballots at least for two different candidates (one follows coercer's instruction, and one for their own intended candidates). System using coercion-evidence [25] can finally tell auditors how many voters are suspiciously being coerced.

5 Existing remote E-voting systems

Here, we present an overview of some existing remote E-voting systems. Among them, we examine whether or not they satisfy the above three security properties, and if so what mechanisms they used.

5.1 Hirt et al. [50]

In [50], candidate options are encrypted by authorities in advance and then sent to voters in the voting stage. However, the order of candidate options are randomized by multiple authorities. In order to instruct voters to select encrypted candidate options for their intended candidates, the order is conveyed to voters using designated verifier proof [41]. In the tallying stage, tallies use homomorphic encryption to sum up ballots and also provide correctness proof for universal verifiability.

As candidate options are encrypted by authorities, voters do not have any randomness knowledge about the encryption. Authorities shuffle and re-encrypt votes, and prove the order of candidate options to voters using 1-out-of-L re-encryption proof [54] [55] [56]. Also using designated verifier proof, voters are not able to transfer this proof to buyers or coercers [41]. However, coerced voters' deceits can still be detected by coercers, therefore, it is not voter-coercion resistant.

5.2 Lee et al. [23]

Lee et al. proposed a mechanism for mix-net based remote E-voting systems to achieve receipt-freeness in [23]. The main idea of [23] is similar to [50]. In it voters have no knowledge of the randomness of the encryptions. Instead of being encrypted by the authorities, voters and a third party randomizer encrypt votes together using so called designated-verifier re-encryption proof (DVRP). Basically, after receiving voters' encrypted ballots, the randomizer re-encrypts them and generates the final encrypted ballots. Also, the randomizer uses designed verifier proof to show voters that their votes are re-encrypted correctly. This proof cannot be transferred to third party.

The randomizer in this scheme is a very important component, as without it the system cannot provide randomness. The randomizer is implemented by a secure hardware device called a tamper resistant randomizer (TRR) [5]. The communications between voters and TRR should not be revealed to attackers. In their case, the authors assume that TRR is part of voters' computer, and the communication links are internal physical links. So it is equal to the assumption of untappable channel between voters and TRR.

In the final tallying stage, without using homomorphic encryption, [23] has to decrypt every single encrypted ballot one by one, using uses threshold cryptography scheme in the encryption and decryption processes. Unfortunately, this scheme is not voter-coercion resistant, as voters are not able to disguise themselves.

5.3 Juels et al. [12]

In [12], each eligible voter gets a secret credential from registrars. In the voting stage, voters encrypt both their credentials and their choices using tallies' public key. Apart from this, ballots also include the proof that voters know encrypted credentials and candidate options, along with proof of the validations of their options. Any ballot without valid proof will be eliminated in the tallying stage. Credentials and candidate options are stored in two separate vectors, but with the same corresponding orders. Both credentials and candidate options will then pass through mix-nets with the same permutation in order to achieve anonymity. In the case that coercion happens, coerced voters can give coercers fake credentials or vote with fake credentials for them. Coercers cannot verify whether or not these given credentials are valid.

This protocol requires untappable channels in the registration stage, which means voters' secret credentials cannot be disclosed to coercers. Smith et al. [57] proposed an efficient scheme based on [12], but it is not secure [46].

5.4 Civitas [10]

Civitas [10] was proposed by Michael R. Clarkson et al. in 2008; it aims to refine and improve [12]. Like other systems, voters are assumed to be able to get their private credentials with guaranteed integrity, and the corresponding public credentials are published in the bulletin board. Private credentials (shares) are generated by distributed registrars (called registration tellers in the paper) using Needham-Schroeder-Lowe protocol [58], which means unless attackers corrupt all or the majority of the registration tellers, they cannot know voters' private credentials.

The key idea to defend against voter-coercion in Civitas is the same as [12] – coerced voters can construct fake credentials and use fake credentials to do whatever coercers desire. To forge a fake credential, voters are required to modify at least one of the private credential shares (obtained from distributed registration tellers). Therefore, it requires at least one registration teller not be corrupted by attackers, and that there exists an untappable channel between voters and this trustworthy registrar.

Apart from this, for scalability reasons, voters are grouped into blocks. This helps Civitas reduce computation time, while [12] does not take time efficiency into account. Other differences between [12] and Civitas, including concrete instantiations of cryptographic components, can be found in [10].

5.5 Helios [52]

Helios was designed mainly for the purpose of low-coercion applications [52]. It highlights the verifiability in remote E-voting systems. In Helios, voters do not need to be authenticated until they cast votes. Under such a situation, anyone can participate and test the system. Ballots are encrypted and displayed in a hash of cipher-texts. Voters can audit their ballots by asking the authorities to display their corresponding cipher-texts. After a voter audits her/his ballot, she/he can choose to re-encrypt her/his ballot again. To deal with coercion, Helios allows voters to send a separate explicit coercion proof to the authorities; however it does not take interactions among voters and coercers into account, which means it is possible that coercers can detect the explicit coercion proof as well. Therefore, Helios is still vulnerable to coercion, which makes it suitable only for low-coercion elections. Also, Helios has printable non-encrypted receipts, so it is not receipt-free.

5.6 Helios2.0 [22]

Ben Adida et.al. analyzed Helios and improved it in [22]. Helios2.0 uses the same encryption mechanism as Helios; however, instead of using mix-net to shuffle votes in the tallying stage, Helios2.0 uses homomorphic encryption mechanism and so does not need to decrypt ballots one by one. It uses exponential ElGamal system and distributive encryption. In order to achieve verifiability, they designed two programs: election tallying program and election verification program. They also improved the user interface. Unfortunately, Helios2.0 is also not receipt-free.

5.7 PGD [59]

In Pretty Good Democracy (PGD) [59], voters use code sheets [60] to cast votes for their intended candidates. In each code sheet, vote codes for candidates are randomized and pairwise distinct. It means that given a voter's vote code, no one else knows her/his intended candidate. Moreover, vote codes are encrypted by authorities and some trustees. In the voting stage, voters enter their ballots IDs and vote codes; voting servers will then check ballots IDs for eligibility; and then encrypt and send them to trustees if they are valid. If trustees find vote codes are

matchable with their corresponding ballots' codes, they will decrypt acknowledgment codes which will be sent back to voters by voting servers. Therefore, PGD provides individual verifiability. Furthermore, acknowledgment codes are per code sheet rather than for individual candidate choices, so PGD offers receipt-freeness. However voters can sell their code sheets to attackers, therefore PGD is neither vote-selling resistant nor voter-coercion resistant.

5.8 Meng et al. [45]

[45] uses [61]'s commitment scheme (called BCP commitment scheme in [45]) to generate keys which are used in voters' identity validation in an interactive way. After validation registrars send voters deniable encryptions of candidate options, voters will select and encrypt candidate choices using BCP commitment scheme. In the tallying stage, tallies can use collision-find algorithm to verify valid ballots. Since voters are able to generate fake credentials (using dishonest opening in deniable encryption scheme [62]) and deceive coercers, it is receipt-free and voter-coercion resistant. More importantly, this system does not need untappable channels, which is a relaxed physical assumption when compared with other systems.

5.9 AFT [46]

AFT [46], developed by R.Araujo et al., employs some of [12]'s ideas. It also allows voters to generate fake credentials under decision Diffie-Hellman (DDH) problem's assumption [63] and give them to coercers in the case that they are coerced. Instead of using random strings as credentials, it uses a tuple with the form of $\sigma = (r, a, b, c)$. Unless the voter reveals her/his r to coercers, coercers cannot distinguish a fake credential from a real one; and (a, b, c) is not necessary to be sent through an untappable channel. Apart from this, if (r, a, b, c) is valid, then (r, a^l, b^l, c^l) is also valid. In other words, the voter can change the value of (a, b, c), and she/he can still be identified as long as she/he keeps using the same r.

The underlying idea to defend against voter-coercion is the same as [12]. One improvement this scheme made is that it reduces the time complexity from quadratic to linear. In order to identify valid votes in liner time, instead of using pairwise plain-text equivalence test (which takes $O(N^2)$ time, N is the number of voters), it only checks the equivalence of two elements in ballots. This reduces identification time to $O(V)$, where V is the total number of votes. The authors also demonstrated that their scheme is resistant to three attacks defined in [12].

5.10 Philip et al. [24]

In [24]'s registration stage, voters are provided with pairs of usernames and passwords, which are used for later identity validation. They modified ElGamal cryptosystem to make it additive homomorphic and used it for ballots encryption. Voters' encrypted ballots are re-encrypted again with some random numbers which are unknown to voters. In such a way, voters cannot construct receipts to prove their votes. In order to make the system robust, they designed multiple authorities and tallies. Although it is receipt-free, voters can still sell their passwords to buyers, as it has no protection measures for passwords. Similarly, coercers can force voters to tell them their passwords.

5.11 Selections [26]

Selections is a voter-coercion resistant protocol for Internet voting [26]. It requires voters to register at a private booth in person. In the registration stage, each voter can get a password from a panic password system [53], and its homomorphic encryption will be published in the public roster. Voters can choose passwords and a large number of panic passwords. By giving panic passwords to coercers, voters can avoid following coercers' instructions. In the voting stage, voters submit commitments to their asserted passwords (real or panic), and the re-encryptions of their public entries, and votes are passed through a mix-net for anonymization. Tallies check the validity of each submission and eliminate duplicate records.

5.12 Caveat Coercitor [25]

Instead of considering voter-coercion resistance, Caveat Coercitor [25] proposed a protocol which satisfies coercion-evidence. Unlike voter-coercion resistant systems, Caveat Coercitor tolerates coercions but records unforgeable evidence for voter-coercions. It outputs the evidence of the amount of suspicious voter-coercions that occurred in the elections. Observers can decide whether or not the outcome is valid based on the number of suspicious ballots. By relaxing strict voter-coercion resistance requirements, it weakens the system's assumptions and makes them more practically acceptable. It allows coerced voters to be able to cast their intended votes without letting an adversary find it.

Caveat Coercitor is composed of two parts: coercion test and coercer independence. Coercion test outputs a number (it is called degree of coercion in the paper) within an estimated range. This number estimates the number of coerced voters in the election. it is crucial that number is an approximation of coerced voters, so that dishonest voters can also claim that they were coerced by attackers even though this was not the case. Coercer independence allows coerced voters to still be able to cast votes for their desired candidates without being detected by coercers. To achieve coercer independence, for each coerced voter V_c who is forced to vote for C_c, there must be at least one free honest voter V_f who really wants to vote for C_c. The idea here is that, by switching V_c and V_f's votes such that coercer cannot distinguish between them. In this case, in addition to V_f's original vote for C_c, V_f can also vote for another candidate, so V_f's ballots will be canceled. One the other hand, if V_c abstains from the election, only the coercer's instruction will be implemented, which is C_c in this case. Therefore, the final outcome is the same in these two scenarios: only one vote for C_c is valid, and one of them will be counted as a suspicious coerced vote. We can see that although the coerced voter can still cast for her/his intended candidate, the coercion increases suspicious ballots, which are invalid in elections. The number of suspicious ballots may have an effect on election results if there is a very small gap between the winner and the loser.

6 Comparisons of remote E-voting systems

In this section, we compare the above existing remote E-voting systems based on several criteria. For example, we will see whether or not they satisfy receipt-freeness, vote-selling resistance and voter-coercion resistance, and what cryptography primitives they used.

6.1 Anonymization techniques

As we know, anonymization is a very basic requirement for remote E-voting systems. Without anonymizing voters' identities and their votes, attackers can easily check for whom each voter votes. This violates vote privacy as defined in the previous section. According to [10], remote E-voting systems can be divided into three categories based on the techniques they used to achieve anonymization.

Blind signature allows registrars to sign voters' ballots, but without disclosing voters' choices to authorities. It keeps links between voters and their choices secret. Systems using blind signature include: [42] [64] [65] [66].

Similarly, homomorphic encryption can directly operate on encrypted ballots without leaking voters' choices (plain-texts). Therefore it also can be used to prevent the relationships between voters and their choices from being revealed to others. [50] [67] [56] belong to this category.

Mix-net is another widely used technique and it shuffles voters' ballots and makes them untraceable [10] [12] [25] [68] [69]. Each mixer only knows its own permutation order, but has no knowledge about others such that even an individual mixer cannot figure out the combined permutation.

6.2 Security properties

In Section 2.2, we listed several security properties that practical remote E-voting systems should satisfy, including the three focused on in this paper. In this section, we analyze how many properties these systems satisfy. Results are shown in Table 1. We can see that some properties, like democracy and vote-privacy, are satisfied by all the listed systems. This means these are very basic requirements for remote E-voting systems, or even for traditional paper based voting systems.

In Section 4.3 we mentioned that there are four protection measures for voter-coercion attacks. Of these, fake credential and panic password are very similar: both of them generate invalid tokens and give them to coercers. Apart from fake tokens, voters use their real tokens to cast ballots. Explicit notification is another a straightforward way to deal with voter-coercion; however it is not robust, as notification itself can be eavesdropped by attackers, or tallies can be corrupted in some cases. Coercion-evidence uses a similar idea to explicit notification, instead of prohibiting coercion it records coercion-evidence. However, it is much more complex and robust than intuitively explicit notifications.

For the above voter-coercion resistant systems, we categorize them based on the protection measures they used. Table 2 shows each system's protection measure.

6.3 Comparison based on trust assumptions

From these existing remote E-voting systems, we can see that more or less they all make some trust assumptions in order to satisfy those properties. We can briefly divide assumptions into two categories:

Table 1 Voting systems in several satisfied security properties

Systems	Democracy	Verifiability	Vote-privacy	Receipt-freeness	Vote-selling resistance	Voter-coercion resistance
Hirt et al. [50]	Yes	UV	Yes	Yes	Yes	No
Lee et al. [23]	Yes	UV	Yes	Yes	Yes	No
Juels et al. [12]	Yes	UV	Yes	Yes	Yes	Yes
Civitas [10]	Yes	UV	Yes	Yes	Yes	Yes
Helios [52]	Yes	UV&SIV	Yes	No	No	No
Helios2.0 [22]	Yes	UV&SIV	Yes	No	No	No
PGD [59]	Yes	UV&WIV	Yes	No	No	No
Meng [45]	Yes	UV	Yes	Yes	Yes	Yes
AFT [46]	Yes	UV	Yes	Yes	Yes	Yes
Philip et al. [24]	Yes	UV	Yes	Yes	No	No
Selections [26]	Yes	UV&WIV	Yes	Yes	Yes	Yes
Caveat Coercitor [25]	Yes	UV&WIV	Yes	Yes	Yes	Yes

UV: Universal Verifiability; WIV: Weak Individual Verifiability; SIV: Strong Individual Verifiability;

Table 2 Categories based on protection measures

Schemes	Fake credential	Coercion-evidence	Explicit message	Panic password
Juels et al. [12]	√			
Civitas [10]	√			
Helios [52]			√	
Helios2.0 [22]			√	
Meng [45]	√			
AFT [46]	√			
Selections [26]				√
Caveat Coercitor [25]		√		

Table 3 Trust assumptions in remote E-voting systems

Systems	Registration channel	Voting channel	Registrars	tallies
Hirt et al. [50]	untappable channel	untappable channel	minor-corrupted	minor-corrupted
Lee et al. [23]	untappable channel	untappable channel	non-corrupted (only one registrar)	minor-corrupted
Juels et al. [12]	untappable channel	anonymous channel	minor-corrupted	minor-corrupted
Civitas [10]	untappable channel	anonymous channel	minor-corrupted	minor-corrupted
Helios [52]	regular channel	anonymous channel	non-corrupted (only one registrar)	non-corrupted
Helios2.0 [22]	regular channel	anonymous channel	non-corrupted (only one registrar)	minor-corrupted
PGD [59]	untappable channel	anonymous channel	non-corrupted (only one registrar)	minor-corrupted
Meng [45]	regular channel	anonymous channel	minor-corrupted	minor-corrupted
AFT [46]	untappable channel	anonymous channel	minor-corrupted	minor-corrupted
Philip et al. [24]	untappable channel	anonymous channel	minor-corrupted	minor-corrupted
Selections [26]	in-person	anonymous channel	minor-corrupted	minor-corrupted
Caveat Coercitor [25]	regular channel	anonymous channel	minor-corrupted	minor-corrupted

Table 4 Cryptography primitives used by schemes

Systems	BS	SS	HERE	MN	PET	ZKP	DVP/ DVRP	Cryptosystem
Hirt et al. [50]		√	√				√	ElGamal
Lee et al. [23]		√		√			√	ElGamal
Juels et al. [12]		√		√	√	√	√	Modified ElGamal
Civitas [10]		√	√ (construct encrypted public credentials)	√	√	√	√	RSA and ElGamal
Helios [52]				√		√		ElGamal
Helios2.0 [22]		√	√			√		Exponential ElGamal
PGD [59]		√	√	√	√	√		ElGamal&Exponential ElGamal
Meng [45]		√		√	√		√	BCP&Exponential ElGamal
AFT [46]		√	√	√	√	√	√	Modified ElGamal
Philip et al. [24]		√	√			√		Modified ElGamal
Selections [26]	√	√	√	√	√	√	√	ElGamal
Caveat Coercitor [25]		√	√	√	√	√	√	ElGamal

BS: Blind Signature; SS: Secret Sharing; HERE: Homomorphic Encryption and Re-Encryption; MN: Mix-nets; PET: Plain-text Equivalence Test; ZKP: Zero Knowledge Proof; DVP: Designated Verifier Proof; DVRP: Designated Verifier Reencryption Proof;

strong assumptions and weak assumptions, based on the extent of difficulties to achieve them in reality. For example, compared with untappable channels, anonymous channels are weaker assumptions as they are easier to be implemented in reality.

Remember that in this paper we divided assumptions into channels assumptions and agents assumptions. Regarding the channels, there are two main types of channels: anonymous and untappable channels. Regarding agents, as we indicated before, we categorize them into two categories: non-corrupted and minor-corrupted. Table 3 shows the analysis for the above remote E-voting systems' trust assumptions. For channels, we divide them into registration channel and voting channel, while agents include registrars and tallies. Basically the registration channel is used for voters to obtain their credentials or keys, while the voting channel is involved in the whole voting stage. Here regular channel only requires that messages are able to reach the destinations. Note that as indicated in the full version of [12], we regard systems using anonymization techniques as using anonymous channels.

6.4 Used cryptography primitives

In this section, we summarize what cryptography primitives were used in the above remote E-voting systems in order to achieve security properties. We also specify which cryptosystems, e.g. ElGamal cryptosystem, were used by these remote E-voting systems. We summarize this in Table 4.

Note that although almost all the listed systems use the secret sharing mechanism, there is some difference among them. Secret sharing can be used in two stages: the registration stage and the tallying stage. Some systems, like Caveat Coercitor [25], use secret sharing in both stages; however, some of them, like [12], only use secret sharing for decryption in the tallying stage. Also, we can see that Zero-knowledge proof is almost essential for all remote E-voting systems.

7 Conclusions

As people and governments are now treating remote E-voting systems more and more seriously, their underlying security issues are attracting much more attentions. Due to the nature of remote E-voting systems, they are more difficult to be supervised compared with traditional methods. One of the most serious issues is security in elections, it can cause elections to be invalid if systems cannot figure it out.

In this paper, we examine several existing remote E-voting systems, and analyze their properties in terms of receipt-freeness, vote-selling resistance and voter-coercion resistance. In addition to that, we also examine other common properties that remote E-voting systems must satisfy.

By examining these existing schemes, we find that although some of them exhibit more robust attributes than others, more further works are needed in this field.

References

[1] R. Mercuri. A better ballot box? *Spectrum, IEEE*, 39(10):46–50, Oct 2002.

[2] Susan Bell, Josh Benaloh, Michael D Byrne, Dana DeBeauvoir, Bryce Eakin, Gail Fisher, Philip Kortum, Neal McBurnett, Julian Montoya, Michelle Parker, et al. Star-vote: A secure, transparent, auditable, and reliable voting system. *The USENIX Journal of Election Technology Systems, 1 (1)*, pages 18–37, 2013.

[3] Roy E Anderson, Richard L Frey, and James R Lewis. Electronic voting system, September 15 1981. US Patent 4,290,141.

[4] Wen-Shenq Juang and LEI Chin-Laung. A secure and practical electronic voting scheme for real world environments. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 80(1):64–71, 1997.

[5] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In *Information Security and CryptologyICISC 2002*, pages 389–406. Springer, 2003.

[6] Rui Joaquim, André Zúquete, and Paulo Ferreira. Revs–a robust electronic voting system. *IADIS International Journal of WWW/Internet*, 1(2):47–63, 2003.

[7] Bo Meng. An internet voting protocol with receipt-free and coercion-resistant. In *Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on*, pages 721–726, Oct 2007.

[8] Xukai Zou, Huian Li, Yan Sui, Wei Peng, and Feng Li. Assurable, transparent, and mutual restraining e-voting involving multiple conflicting parties. In *INFOCOM, 2014 Proceedings IEEE*, pages 136–144, April 2014.

[9] Geir Røsland. *Remote Electronic Voting*. PhD thesis, Citeseer, 2004.

[10] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a secure voting system. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 354–368, Washington, DC, USA, 2008. IEEE Computer Society.

[11] Aviel D. Rubin. Security considerations for remote electronic voting. *Commun. ACM*, 45(12):39–44, December 2002.

[12] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, WPES '05, pages 61–70, New York, NY, USA, 2005. ACM.

[13] Benjamin B Bederson, Bongshin Lee, Robert M Sherman, Paul S Herrnson, and Richard G Niemi. Electronic voting system usability issues. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152. ACM, 2003.

[14] Tadayoshi Kohno, Adam Stubblefield, Aviel D Rubin, and Dan S Wallach. Analysis of an electronic voting system. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 27–40. IEEE, 2004.

[15] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.

[16] David Jefferson, Aviel D Rubin, Barbara Simons, and David Wagner. Analyzing internet voting security. *Communications of the ACM*, 47(10):59–64, 2004.

[17] Benoit Chevallier-Mames, Pierre-Alain Fouque, David Pointcheval, and Jacques Traoré. On some incompatible properties of voting schemes. In *In IAVoSS Workshop On Trustworthy Elections, WOTE06*. Citeseer, 2006.

[18] Laure Fouard, Mathilde Duclos, and Pascal Lafourcade. Survey on electronic voting schemes. *supported by the ANR project AVOTÉ*, 2007.

[19] Huian Li, A.R. Kankanala, and Xukai Zou. A taxonomy and comparison of remote voting schemes. In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pages 1–8, Aug 2014.

[20] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying properties of electronic voting protocols. In *in" Proceedings of the IAVoSS Workshop On Trustworthy Elections (WOTE06*. Citeseer, 2006.

[21] Bo Meng. A critical review of receipt-freeness and coercion-resistance. *Information Technology Journal*, 8(7):934–964, 2009.

[22] Ben Adida, 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 *Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections*, pages 10–10. USENIX Association, 2009.

[23] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Providing receipt-freeness in mixnet-based voting protocols. In Jong-In Lim and Dong-Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 245–258. Springer Berlin Heidelberg, 2004.

[24] Adewole A Philip, Sodiya Adesina Simon, and Arowolo Oluremi. A receipt-free multi-authority e-voting system. *International Journal of Computer Applications*, 30(6):15–23, 2011.

[25] G.S. Grewal, M.D. Ryan, S. Bursuc, and P.Y.A. Ryan. Caveat coercitor: Coercion-evidence in electronic voting. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 367–381, May 2013.

[26] Jeremy Clark and Urs Hengartner. Selections: Internet voting with over-the-shoulder coercion-resistance. In George Danezis, editor, *Financial Cryptography and Data Security*, volume 7035 of *Lecture Notes in Computer Science*, pages 47–61. Springer Berlin Heidelberg, 2012.

[27] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 544–553, New York, NY, USA, 1994. ACM.

[28] Riza Aditya, Byoungcheon Lee, Colin Boyd, and Ed Dawson. An efficient mixnet-based voting scheme providing receipt-freeness. In *Trust and Privacy in Digital Business*, pages 152–161. Springer, 2004.

[29] Zuzana Rjašková. Electronic voting schemes. *Diplomová práca, Bratislava*, 2002.

[30] Valtteri Niemi and Ari Renvall. How to prevent buying of votes in computer elections. In *Advances in CryptologyASIACRYPT'94*, pages 164–170. Springer, 1995.

[31] Emmanouil Magkos, Mike Burmester, and Vassilis Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channels. In *Towards The E-Society*, pages 683–693. Springer, 2001.

[32] Josh Benaloh. Rethinking voter coercion: The realities imposed by technology. *The USENIX Journal of Election Technology and Systems*, 82, 2013.

[33] Ben Adida. *Advances in cryptographic voting systems*. PhD thesis, Massachusetts Institute of Technology, 2006.

[34] David Chaum. Blind signatures for untraceable payments. In David Chaum, RonaldL. Rivest, and AlanT. Sherman, editors, *Advances in Cryptology*, pages 199–203. Springer US, 1983.

[35] Torben Pryds Pedersen. A threshold cryptosystem without a trusted party. In *Advances in CryptologyEUROCRYPT91*, pages 522–526. Springer, 1991.

[36] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in CryptologyCRYPTO91*, pages 129–140. Springer, 1992.

[37] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[38] Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In *Advances in CryptologyASIACRYPT 2000*, pages 162–177. Springer, 2000.

[39] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981.

[40] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[41] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *Advances in CryptologyEUROCRYPT96*, pages 143–154. Springer, 1996.

[42] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols*, pages 25–35. Springer, 1998.

[43] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Coercion-resistance and receipt-freeness in electronic voting. In *Computer Security Foundations Workshop, 2006. 19th IEEE*, pages 12–pp. IEEE, 2006.

[44] R. Kusters and T. Truderung. An epistemic approach to coercion-resistance for electronic voting protocols. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 251–266, May 2009.

[45] Bo Meng, Zimao Li, and Jun Qin. A receipt-free coercion-resistant remote internet voting protocol without physical assumptions through deniable encryption and trapdoor commitment scheme. *Journal of Software*, 5(9), 2010.

[46] Roberto Araujo, Sébastien Foulle, and Jacques Traoré. A practical and secure coercion-resistant scheme for internet voting. In *Towards Trustworthy Elections*, pages 330–342. Springer, 2010.

[47] K Memon, Dileep Kumar, and S Usman. Next generation a secure e-voting system based on biometric fingerprint method. In *International Conference on Information and Intelligent Computing (IPCSIT)*, pages 26–32, 2011.

[48] Rebecca Mercuri. A better ballot box? *Spectrum, IEEE*, 39(10):46–50, 2002.

[49] Georgios Tsoukalas, Kostas Papadimitriou, Panos Louridas, and Panayiotis Tsanakas. From helios to zeus. In *USENIX Journal*

of Election Technology and Systems (JETS) and Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, EVT/WOTE, volume 13, 2013.

[50] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In *Advances in CryptologyEUROCRYPT 2000*, pages 539–556. Springer, 2000.

[51] Hugo L Jonker and Erik P de Vink. Formalising receipt-freeness. In *Information Security*, pages 476–488. Springer, 2006.

[52] Ben Adida. Helios: Web-based open-audit voting. In *USENIX Security Symposium*, volume 17, pages 335–348, 2008.

[53] Jeremy Clark and Urs Hengartner. Panic passwords: Authenticating under duress. *HotSec*, 8:8, 2008.

[54] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in CryptologyCRYPTO94*, pages 174–187. Springer, 1994.

[55] Ronald Cramer, Matthew Franklin, Berry Schoenmakers, and Moti Yung. Multi-authority secret-ballot elections with linear work. In *Advances in CryptologyEUROCRYPT96*, pages 72–83. Springer, 1996.

[56] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. *European transactions on Telecommunications*, 8(5):481–490, 1997.

[57] Warren D Smith. New cryptographic election protocol with best-known theoretical properties. In *Proc. of Workshop on Frontiers in Electronic Elections*, 2005.

[58] Gavin Lowe. An attack on the needham-schroeder public-key authentication protocol. *Information processing letters*, 56(3):131–133, 1995.

[59] Peter YA Ryan and Vanessa Teague. Pretty good democracy. In *Security Protocols XVII*, pages 111–130. Springer, 2013.

[60] David Chaum. Surevote: technical overview. In *Proceedings of the workshop on trustworthy elections (WOTE01)*, 2001.

[61] Emmanuel Bresson, Dario Catalano, and David Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 37–54. Springer Berlin Heidelberg, 2003.

[62] Bo Meng and JiangQing Wang. An efficient receiver deniable encryption scheme and its applications. *Journal of Networks*, 5(6):683–690, 2010.

[63] Dan Boneh. The decision diffie-hellman problem. In *Algorithmic number theory*, pages 48–63. Springer, 1998.

[64] Miyako Ohkubo, Fumiaki Miura, Masayuki Abe, Atsushi Fujioka, and Tatsuaki Okamoto. An improvement on a practical secret voting scheme. In *Information Security*, pages 225–234. Springer, 1999.

[65] I. Ray, I. Ray, and N. Narasimhamurthi. An anonymous electronic voting protocol for voting over the internet. In *Advanced Issues of E-Commerce and Web-Based Information Systems, WECWIS 2001, Third International Workshop on.*, pages 188–190, 2001.

[66] S. Ibrahim, M. Kamat, M. Salleh, and S.R.A. Aziz. Secure e-voting with blind signature. In *Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on*, pages 193–197, Jan 2003.

[67] Josh Benaloh. *Verifiable secret-ballot elections*. PhD thesis, PhD thesis, Yale University, 1987.

[68] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, PODC '01, pages 274–283, New York, NY, USA, 2001. ACM.

[69] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme. In *Advances in CryptologyEUROCRYPT95*, pages 393–403. Springer, 1995.