Robust Coercion-Resistant Registration for Remote E-voting (Extended Abstract)

Taisya Krivoruchko
tkrivoru@cpsc.ucalgary.ca
Department of Computer Science
University of Calgary
Calgary, AB Canada, T2N 1N4

Abstract

During the registration phase of remote electronic voting schemes, the voter obtains a credential that proves the validity of her vote later, in the tallying phase. Most of the existing remote coercion-resistant schemes either omit the description of the the registration phase or assume the existence of a trusted registering authority that authenticates voters as well as creates and distributes credentials among them.

We present a registration protocol suitable for some of the existing remote coercion-resistant schemes. The advantage of our protocol is that it is based on weaker assumptions on the trustworthiness of the registration entities. In our protocol, the trust is distributed among several entities and the functionality of the trusted entities is simplified. This results in higher security and an increased degree of robustness.

1. Introduction

One criterion in the classification of electronic voting protocols is the place where voters cast their vote. Accordingly, electronic voting schemes belong to two groups: *poll-site* and *remote*. In a poll-site scheme, votes are cast at special locations only, and those locations can be either supervised or not. These types of schemes provide a more controlled environment, which aids in achieving security. Remote schemes, on the other hand, are designed such that voters can vote from any location with Internet connection. It increases flexibility but poses additional threats, such as large-scale voter coercion.

Remote e-voting schemes have the following phases. First, during the *set-up phase*, election parameters, such as the list of candidates and public keys of the participants, are published. Next, during the *registration phase*, eligible voters register to receive a credential. Then, during the *voting phase*, voters cast their votes along with the attached credentials. Finally, the results are computed in the *tallying phase*, based on the values of those votes that have valid credentials attached. The structure of the poll-site voting schemes is similar, but it does not require the separation of the registration and the voting phases, since voters can identify themselves at the poll-site on the election day.

While the proper implementation of all four phases is crucial to the security of a voting system, the difficulties of designing the particular phases differ. Papers that describe electronic voting schemes focus on the last two phases - voting and tallying. The registration phase has received far less attention, in spite of having been admitted to be a crucial phase from the point of view of security. For example, if non-eligible voters are able to register, none of the security precautions of the subsequent phases will defend against rigged results.

The majority of electronic voting schemes either omit the description of the registration phase or assume the presence of a trusted registering authority that both generates the credentials and distributes them among the voters. In this paper, we present a registration protocol that avoids this strong assumption. Our protocol is suitable for a number of existing coercion-resistant remote e-voting schemes.

1.1. Coercion-Resistance

The security and suitability of electronic voting schemes is measured with respect to a set of properties. The three properties that are known as *basic properties* are *democracy* (a voter can vote if, and only if, she is eligible; she can cast at most one vote), *accuracy* (the final tally reflects the values of all valid votes) and *privacy* (no information can be obtained about the individual votes of the voters).

While the basic properties are satisfied by the majority of electronic voting schemes, there are other properties that may also be required, for example: *individual verifiability* (the voter can verify that her vote was included in the final tally), *universal verifiability* (any participant or observer can verify the results of the tallying), *receipt-freeness* (the voter cannot prove to a third party the value of her vote) and *coercion-resistance* (to be defined in the next paragraph). For a detailed discussion of different properties see, e.g, [6, 10].

In this paper, we take into account all the listed properties, while focusing on the property of coercion-resistance. The notion of coercion-resistance was introduced in [8] and was aimed to strengthen the notion of receipt-freeness [4]. While receipt-freeness protects against a passive adversary, who does not communicate with a voter throughout the voting process, coercion-resistance protects against an active, more powerful, adversary: he can communicate with the voter throughout the elections and demand of the voter to take certain actions in order to prevent the voter from casting a vote of her choice. We call a voting scheme coercion-resistant, if the voter cannot prove to the coercer that she followed his demands. Just like a receipt-free scheme, a coercion-resistant scheme ensures that a voter cannot prove to a third party how she voted. However, a coercion resistant scheme has to protect against three additional attacks.

The first attack, the *randomization attack*, allows a coercer to prevent a voter from casting a vote of her choice with a large probability. We believe that such an attack can be mount on the schemes similar to the ones described in [3, 13] in the following way. In these schemes, voters cast ballots at a poll-site. A ballot consists of two parts: the left part of the ballot contains the names of the candidates in a random order and the right part of the ballot contains the corresponding check boxes. The voter puts a check mark in the check box next to the name of the preferred candidate, and then her ballot is scanned. She can take either side of the ballot home. The choice of the voter can be randomized in the following way. The coercer can tell the voter where to put the check mark, without knowing in advance the order of the names (e.g., tell the voter to put the check mark in the second check box regardless of the order of candidates). With a high probability (depending on the order of candidates on the left side of the ballot), the voter will put a check mark next to one of the candidates that she did not intend to vote for. The voter can prove to the coercer that she complied with the demands by showing him the right side of her ballot.

The second type of an attack, the *forced-abstention attack*, is aimed to prevent a voter from participating in the elections. For example, consider a voting scheme in which a list of voters that participated in the elections is published (cf., [14, 20]). We believe that coercion in those schemes can be performed in the following way. The attacker can demand of the voter to refrain from participating in the elections. He can check whether the voter complied with his demands by examining the list of participated voters.

Finally, the *simulation attack* allows an attacker to impersonate a voter if he manages to obtain the voter's credential after the registration phase. This attack is possible in any scheme that permits an observer to verify the validity of credentials. In that case, an attacker can make sure that the credential he buys from a voter is valid.

We focus on coercion-resistance, because, currently, this is the strongest property preventing vote-buying, which is a major issue of remote voting schemes. More specifically, we consider the registration process of remote schemes and design a registration protocol that is suitable for a group of coercion-resistant remote voting schemes.

1.2. Previous Work

We are not aware of a remote e-voting scheme that gives considerable amount of attention to the reg-

istration phase. The majority of current electronic voting schemes either do not mention the registration phase or assume complete trust in the registering authority. We are aware of one paper [12] that describes a method of voter authorization using visual cryptography. The idea is to mail all the eligible voters a printed transparency, generated by election officials, which permits the voters to identify the correct elections server and allows the elections server to identify eligible voters. The advantage of this method is that it is simple to use and does not require the voter to go to a poll-site for the registration. However, this method assumes some trust in the elections officials and the correct work of the elections server. Furthermore, the method is not suitable for coercion-resistant schemes, since, as its authors point out, the voter can easily sell her transparency.

To our knowledge, the list of remote electronic voting schemes that claim to be coercion-resistant is the following: [8, 20, 17, 1, 9, 11, 5, 16]. Among them, [9, 16] do not mention the registration phase, [8, 20, 17, 5] assume a fully trusted registering authority that generates the credentials itself and distributes them among the voters and [11, 1] describe a more robust approach.

The method of registration described in [11, 1] (by Schweisgut and Acquisti, respectively) has the following structure. The registering authorities send messages to the voter that serve as a basis for constructing a credential. They also post messages on the bulletin board needed for the tallying authorities to be able to check the validity of the credentials. The registering authorities send a designated-verifier proof to the voter that shows the correct relationship between the messages they send to the voter and the ones they post on the bulletin board.

While Schweisgut does not specify what actions the voter should take in case she is not satisfied with the proofs received from one or more of the registering authorities, in Acquisti's scheme the voter can complain about those authorities. We believe that the coercer can mount a forced-abstention attack by telling the voter to complain about all of the registering authorities or a sufficiently large number of them, in which case the voter is not able to obtain a credential. Hence, the registration schemes of Acquisti and Schweisgut are not suitable for a coercion-resistant e-voting scheme.

The registration phase of the schemes in [8, 20, 17, 5] involves one trusted registering authority R that performs both credential generation and credential distribution. This makes R a centralized point of failure, because:

- R knows the values of the credentials and thus can impersonate any voter.
- R can give away credentials to non-eligible voters.
- R knows the relationship between the voters and their credentials, which might compromise privacy.
- Without appropriate verification mechanisms in place, R can give fake credentials to voters or give the same credentials to different voters.
- It is often undefined what the voter should do in case R fails to issue her a credential.

1.3. Our Contribution

In this paper, we propose a registration protocol that is suitable for the e-voting schemes described in [8, 17, 20]. The registration phase of these protocols is based on one registering authority. We modify their registration protocol to make it more robust by distributing most of the functions of the registering authority among several entities, which results in two improvements. First, a higher degree of robustness is achieved, which increases the security of the system through increasing its fault-tolerance. Second, the functionality of the entity that needs to be trusted is simplified, which means that ensuring and verifying the correct performance of the trusted entity is easier.

1.4. The Organization of the Paper

In Section 2, we describe the elections model that we work with and present a brief overview of the cryptographic primitives that we use. Then we describe our proposed robust registration protocol in Section 3. In Section 4, we discuss the security properties of our registration protocol and compare it to the currently used approach that involves one trusted registering authority.

2. Framework and Tools

2.1. Framework

The participants of elections are voters $V_1, \ldots, V_m$, an identifying authority I, registering authorities $R_1, \ldots, R_n$, tallying authorities $T_1, \ldots, T_s$, and a coercer. We note that the registering and the tallying authorities can be the same entities, i.e., the functionality of the registering authorities can be performed by the tallying authorities.

The objective of the coercer is to prevent the voters from casting the votes of their choice. As discussed in Section 1.1, the coercer communicates with the voter throughout the election process and can achieve his goals in a number of ways, e.g., by making the voter vote in a certain way or abstain from the elections.

The majority of the registering and tallying authorities as well as the identifying authority is assumed to be honest. A minority of Rs and Ts can collaborate with the coercer, in which case the coercer knows all the information that is known to the coerced authorities. Furthermore, it is only the coercer and the malicious authorities who know which of the authorities are honest and which of them are coerced.

The participants communicate through different types of channels and a bulletin board. We assume that the content of the bulletin board is accessible to all participants and observers and that everyone can write to it but noone can delete information from it. Furthermore, we assume the existence of anonymous channels (for the tallying phase) and untappable channels (for the registration phase).

2.2. Cryptographic Primitives

In this section, we give a high-level description of the cryptographic primitives that we use. The details, as well as the descriptions of cryptosystems that implement these primitives, can be found in [18, 2, 19].

A *designated verifier proof* is a proof that is aimed at a particular participant in such a way that the participant himself can verify the correctness of the proof but he cannot use that proof to convince other participants.

A *probabilistic encryption scheme* is an encryption scheme that uses randomness in the process of encryption letting the same plaintext be encrypted into different ciphertexts. It is computationally hard for a person who does not possess the private key to determine whether two given ciphertexts are encryptions of the same plaintext and whether a given ciphertext is an encryption of a given plaintext. A probabilistic encryption scheme with *verifiable re-encryption* allows a participant who does not know the value of the private key to re-encrypt a given ciphertext and provide a proof of correct re-encryption.

A *threshold encryption scheme* is an encryption scheme in which the private key is shared among a group of entities in such a way that the decryption of a given ciphertext can be performed if, and only if, a large enough subset of the entities collaborate. In that case, the ciphertext is decrypted without revealing the private key shares. A threshold encryption scheme with *verifiable decryption* allows the owners of shares of the private key to prove that the decryption of a given ciphertext was performed correctly.

A *plaintext equivalence test (PET)* of a probabilistic threshold encryption scheme is an algorithm that allows a large enough group of owners of shares of the private key to determine, in a provable manner, whether two given ciphertexts are encryptions of the same plaintext, without decrypting the ciphertexts or revealing the shares of the private key.

A *shuffle of ciphertexts* is an algorithm that takes a list of ciphertexts and produces a new list of ciphertexts by re-encrypting the ciphertexts on the old list and then, outputting them in a new order. A shuffle is *verifiable* if it provides a proof of correctness of shuffling. A verifiable re-encryption mixnet is a chain of connected verifiable shuffles.

Before we proceed to the description of our registration protocol, we present a high-level description of the scheme from [8] by Juels et al.

2.3. An e-voting scheme by Juels et al

First, during the set-up phase, a probabilistic threshold cryptosystem with PETs, verifiable decryption and verifiable re-encryption is chosen and the required domain parameters are published. The tallying authorities generate in a threshold manner

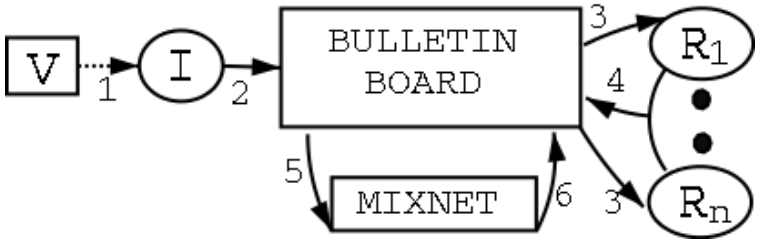

Figure 1: The outline of our registration protocol

a public–private key pair and publish the public key PK_T.

During the registration phase, each eligible voter V_i obtains a credential σ_i from the registering authority R through an untappable channel, after which the registering authority R publishes a list $\vec{A}_1$ of the encrypted credentials $E_{PK_T}(\sigma_i)$.

During the voting phase, the voters attach the encryptions of their credentials to their votes and publish them anonymously on the bulletin board.

During the tallying phase, the tallying authorities Ts determine the validity of the published votes by examining the validity of the attached encrypted credentials in the following manner. First, the Ts eliminate votes with the same credential and form a list $\vec{B}_1$. Then the Ts pass the lists $\vec{A}_1$ and $\vec{B}_1$ through a verifiable re-encryption mixnet to obtain the lists $\vec{A}_2$ and $\vec{B}_2$, respectively. Then, in order to determine the validity of the submitted credentials, the Ts compare the encrypted credentials in $\vec{B}_2$ to the valid encrypted credentials in $\vec{A}_2$ using PETs. If there is a match, then the ciphertext from $\vec{B}_2$ encrypts a valid credential. Ts form the list ℓ of those votes that had a valid credential attached. Finally, the Ts decrypt the votes in ℓ in a verifiable manner.

In the registration protocol described in the next section, we make use of the technique used by Juels et al during the tallying phase that allows the tallying authorities to blindly determine the validity of credentials. In our protocol, we apply that technique in the registration phase to let the registering authorities blindly determine the eligibility of voters.

3. A new Registration Protocol

Figure 1 depicts an outline of our registration protocol. First, a voter V_i generates a token and encrypts it. Then she sends through an untappable channel the encrypted token along with a proof of her identity to the identifying authority I (step 1 on Figure 1), which publishes information for the registering authorities Rs (step 2 on Figure 1). Based on that information, the Rs determine whether the voter is eligible to receive a credential, without learning the voter's identity (step 3). With the approval of the majority of the Rs, the voter's token is added to the list of valid credentials posted on the bulletin board (step 4). If the majority of Rs decide that V_i is not eligible to obtain a credential (i.e., either she is not eligible to vote or she has already received a credential), then her token is discarded. Finally, the list of valid encrypted credentials is formed using a verifiable re-encryption mixnet (steps 5 and 6).

Our protocol consists of four parts: set-up, credential generation, authorization and forwarding information. The information posted by the participating authorities on the bulletin board during each part is authenticated and its integrity is protected.

Set-Up

Choosing a threshold encryption scheme: a threshold probabilistic public key cryptosystem with PETs as well as verifiable decryption and re-encryption is chosen and the necessary domain parameters are published. An example of an appropriate cryptosystem is the threshold ElGamal described in, e.g., [19]. The registering authorities Rs generate public–private key pair and publish the public key PK_R while sharing the private key SK_R. Similarly, the tallying authorities Ts gener-

ate a public–private key pair in a threshold manner and publish PK_T.

Initializing a mixnet: A verifiable re-encryption mixnet is initialized and all the necessary parameters are published.

Compiling a list of valid PINs: a deterministic algorithm that computes a digital representation of the identity of voters is chosen. This algorithm converts an identity (i.e. a string of characters and numbers that uniquely identifies an individual) to the corresponding number that we call a PIN. Using the chosen algorithm, the PINs of the eligible voters are computed and the list of valid PINs is created. This list is first encrypted with PK_R. Then it is passed through the verifiable mixnet which publishes the list ℓ_{pin} of re-encrypted, reshuffled valid PINs.

Credential Generation

Token Generation: a voter V generates a random number σ_V that we call a *token*. If a voter is eligible to obtain a credential, then this token is going to become her credential during the information forwarding step.

Encrypting the token: V encrypts her token with PK_T to obtain $E_{PK_T}(\sigma_V)$.

Authorization

Identification: V presents a proof of her identity along with her encrypted token $E_{PK_T}(\sigma_V)$ to the identifying authority I through an untappable channel. During this step, the voter must be separated from the coercer (e.g., using a voting booth).

Computing the encrypted PIN: I computes the PIN_V corresponding to V's identity and encrypts it using PK_R to obtain $E_{PK_R}(\text{PIN}_V)$.

Re-encrypting the token: I re-encrypts the voter's token to obtain $ReE_{PK_T}(\sigma_V)$. It provides two zero-knowledge designated verifier proofs to the voter: one that proves the correct encryption of her PIN and one that proves the correct re-encryption of her token. (This step is optional, since I is assumed to be trustworthy.)

Posting information for the Rs: I posts $E_{PK_R}(\text{PIN}_V)$ and $ReE_{PK_T}(\sigma_V)$ on the bulletin board.

Determining eligibility: the Rs determine whether V is eligible to vote and whether she has already received a credential. Using PETs, the Rs determine whether $E_{PK_R}(\text{PIN}_V)$ has the same plaintext as any of the encrypted valid PINs in ℓ_{pin}. If there is a match, say, $E_{PK_R}(\text{PIN}_V)$ corresponds to the $i-$th encrypted PIN in ℓ_{pin}, the voter is eligible. Then the Rs check whether the voter with the $i-$th PIN has already received a credential.

If $E_{PK_R}(\text{PIN}_V)$ is approved by the majority of the Rs (which happens in case they decide that the voter whose identity corresponds to the encrypted PIN forwarded from the identifying authority is eligible to receive a credential), the registration proceeds to the next step. On the other hand, if the majority of the registering authorities decide that either the voter is not eligible to vote or she has already received a credential, the voter is denied in registration.

Forwarding Information

Updating the list of valid credentials: Rs add V's re-encrypted token $ReE_{PK_T}(\sigma_V)$ to the list of valid re-encrypted tokens $\ell_{\text{re\_cr}}$. Those tokens that get added to the list become credentials.

Shuffling the credentials: the list $\ell_{\text{re\_cr}}$ passes through a verifiable re-encryption mixnet that reshuffles and re-encrypts the credentials using PK_T. The output of the mixnet is the final list of encrypted credentials $\ell_{\text{final\_cr}}$. This list contains the credential of each eligible voter that participated during the registration phase. This list will be used by the tallying authorities in the tallying phase to determine whether a given credential is valid.

Recording participation: upon adding $ReE_{PK_T}(\sigma_V)$ to the list $\ell_{\text{re\_cr}}$, the Rs record the fact that the voter with the $i-$th encrypted PIN in ℓ_{pin} has obtained a credential: each authority maintains a list of encrypted PINs and puts a "digital checkmark" against the PIN of the participating voter.

4. Analysis of the Proposed Scheme

4.1. Applicability

Our registration protocol is designed to achieve two goals by the end of the registration phase. First, each eligible voter has a credential. Second, the list of encrypted valid credentials is published on the bulletin board. This means that our protocol is suitable to replace the current registration phase of

[8] (and the similar schemes from [17] and [20]) that achieves the same two goals.

The voting and the tallying phases of the Juels et al scheme can proceed as described in [8]. During the voting phase, the voter attaches the obtained credential to her vote and anonymously posts the vote/credential pair. During the tallying phase, the tallying authorities determine the validity of the votes by examining the attached credentials. They determine the validity of the credentials using the list of the re-encrypted valid credentials $\ell_{\text{final\_cr}}$ published by the registering authorities during the registration phase.

4.2. Security

We assume the honest behavior of the identifying and the majority of the registering authorities. We also assume that the coercer cannot constantly observe the voter's actions. We assume the existence of untappable channels from each voter to the identifying authority. Furthermore, we rely on the security of the underlining cryptographic primitives.

We note that we do not make any assumptions that are not already present in the scheme by Juels et al. Thus, replacing the current registration phase of their scheme by our registration phase does not result in the need for additional assumptions.

Under the assumptions above, the following conditions are satisfied:

1. Every voter is able to obtain a valid credential if, and only if, she is eligible to vote.
2. No participant, except for the voter, learns the value of her credential.
3. The link between the voter and her credential cannot be reconstructed.
4. Each voter can verify the validity of her credential.
5. The coercer cannot determine whether the voter abstained from participating in the registration phase. Furthermore, the voter cannot prove to the coercer the value of her credential.

The above conditions are necessary for achieving the properties of democracy, accuracy, privacy, verifiability and coercion-resistance, respectively.

The first condition is satisfied because I and the majority of the Rs are assumed to be honest, hence, the credentials of eligible voters are included in the final list of valid encrypted credentials. Keeping track of the voters that already obtained one credential allows eliminating multiple voting.

The second condition is satisfied because the voter never discloses the value of her token.

The third condition is achieved due to the final reshuffling of valid encrypted credentials. Without that reshuffling, I knows the link between the voters and their re-encrypted credentials.

The forth condition is satisfied due to the fact that I provides voters with proofs of correct re-encryption of their token and that the voters can see whether their re-encrypted token was added to the list of valid credentials. Furthermore, during the information forwarding step, the signed credentials are passed through a re-encryption mixnet that is verifiable.

The last condition is achieved due to the following reasons. First, the coercer cannot determine whether the voter abstained from participating. This is ensured due to the integrity of I and the fact that the coerced Rs deal with encrypted PINs only, which means they do not learn the PINs of the participated voters (provided the security of the encryption scheme). Furthermore, the voter cannot prove to the coercer the validity of her credential, since she cannot prove that one of the re-encrypted credentials posted on the bulletin board is an encryption of her credential. Thus, even if the coercer instructs the voter to send a certain token to I, the voter can choose to send a different token without the coercer finding it out.

4.3. Comparison with the Previous Approach

The participating authorities of the Juels et al scheme are a trusted registering authority and a group of tallying authorities the majority of which are honest. In our registration protocol, we distribute most of the functionality of their trusted registering authority among a group of authorities, the majority of which are honest. However, there is still one function that used to be performed by the trusted registering authority and now is performed

by the trusted identifying authority – voter authorization. The reason why this function has to be performed by a trusted entity is because if any malicious authority learns the identity of a voter, then the coercer can mount a forced-abstention attack by demanding of the voter to not register and checking if the voter complied with his demands using the information from the malicious authorities.

By distributing most of the functionality of one trusted authority over a group of several entities, we achieved a number of benefits:

- The trust is divided among several entities, which ensures a higher degree of fault-tolerance.
- The functionality of the trusted entities is simplified, which makes it easier to ensure their correct functioning.
- The failure of a minority of the registering authorities does not affect the security of the system.
- The values of the credentials are known to the voters only.

We note that the disadvantage of our approach is its complexity, which also implies the difficulty of understanding and explaining it, and increased overhead on the registering authorities and other participants. Our approach is useful in those election scenarios where a registration phase is necessary and it is difficult to find a trusted third party to facilitate it.

4.4. Issues Not Addressed

We do not consider the risks related to any remote voting scheme, such as Denial of Service attacks or shoulder surfing (see [7, 15] for discussions of those risks). We do not address the details and feasibility of implementation of anonymous and untappable channels as well as the bulletin board. Furthermore, we do not discuss the implementation of the identifying methods. We do not consider the procedures of choosing the required cryptoprimitives and determining the corresponding domain parameters. Finally, we do not address the fact that the voter needs to use a software for different computations and that the software should be trusted to perform calculations correctly and to not leak information to the coercer.

Conclusion

The majority of the existing coercion-resistant remote electronic voting schemes either ignores the registration phase or assumes the existence of a trusted registering authority that both generates and distributes credentials to the voters. We have presented a registration protocol suitable for a group of coercion-resistant remote electronic voting schemes that avoids such a strong assumption. In our protocol, trust is distributed among several participating entities, which results in a minimized functionality of the trusted entities and an increased degree of robustness.

Acknowledgment

I would like to thank Renate Scheidler for our many helpful discussions, for her time and suggestions. I would like to thank Márton Naszódi for numerous very inspiring conversations and comments.

References

[1] Alessandro Acquisti, *Receipt-free homomorphic elections and write-in ballots*, Cryptology ePrint Archive, Report 2004/105, `http://eprint.iacr.org/`, 2004.

[2] Ben Adida, *Advances in cryptographic voting systems*, Ph.D. thesis, Massachusetts Institute of Technology, 2006.

[3] Ben Adida and Ronald Rivest, *Scratch & vote: Self-contained paper-based cryptographic voting*, ACM Workshop on Privacy in the Electronic Society, 2006.

[4] Josh Benaloh and Dwight Tuinstra, *Receipt-free secret-ballot elections (extended abstract)*, STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing (New York, NY, USA), ACM Press, 1994, pp. 544–553.

[5] Michael Clarkson and Andrew Myers, *Coercion-resistant remote voting using decryption mixes*, Workshop on Frontiers in Electronic Elections, Milan, Italy, 2005.

[6] Stéphanie Delaune, Steve Kremer, and Mark Ryan, *Verifying properties of electronic voting protocols*, Workshop On Trustworthy Elections, ftp://ftp.cs.bham.ac.uk/pub/authors/M.D.Ryan/06-wote.pdf, 2006.

[7] David Jefferson, Avi Rubin, Barbara Simons, and David Wagner, *Analyzing internet voting security*, Communications of the ACM **47** (2004), no. 10, 59–64.

[8] Ari Juels, Dario Catalano, and Markus Jacobsson, *Coercion-resistant electronic elections*, Workshop on Privacy in the Electronic Society, Alexandria, Virginia, US, November 2005.

[9] Mirosław Kutyłowski and Filip Zagórski, *Coercion-free internet voting with receipts*, Workshop on e-Voting and e-Government in Edinburgh, UK, February 2006.

[10] Peter Neumann, *Security criteria for electronic voting*, 16th National Computer Security Conference Baltimore, Maryland, http://www.csl.sri.com/users/neumann/ncs93.htm, September 1993.

[11] Jörg Schweisgut, *Coercion-resistant electronic elections with observer*, 2nd International Workshop on Electronic Voting, Bregenz, August 2006.

[12] Nathanael Paul, David Evans, Avi Rubin, and Dan Wallach, *Authentication for remote voting*, ACM Workshop on Human-Computer Interaction and Security Systems, Fort Lauderdale, Florida, 2003.

[13] Brian Randell and Peter Ryan, *Voting technologies and trust*, IEEE Security and Privacy **4** (2006), no. 5, 50–56.

[14] Ronald Rivest, *The threeballot voting system*, http://theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf, October 2006.

[15] Avi Rubin, *Security considerations for remote electronic voting over the internet*, The Magazine of USENIX and SAGE **26** (2001), no. 1, 20–28.

[16] Anna Shubina and Sean Smith, *Design and prototype of a coercion-resistant, voter verifiable electronic voting system*, Proceedings of the 2nd Annual Conference on Privacy, Security and Trust, 2004, pp. 29–39.

[17] Warren Smith, *New cryptographic election protocol with best-known theoretical properties*, Workshop on Frontiers in Electronic Elections, Milan, Italy, September 2005.

[18] Warren D. Smith, *Cryptography meets voting*, http://www.math.temple.edu/~wds/homepage/cryptovot.pdf, 2005.

[19] Stefan Weber, *A coercion-resistant cryptographic voting protocol - evaluation and prototype implementation*, Master's thesis, Darmstadt University of Technology, 2006.

[20] Stefan Weber, Roberto Araújo, and Johannes Buchmann, *On coercion-resistant electronic elections with linear work*, International Conference on Availability, Reliability and Security (2007), 908–916.