Masked Ballot Receipt-Free Elections

Roland Wen

University of New South Wales, Australia
rolandw@cse.unsw.edu

Technical Report
UNSW-CSE-TR-0807
April 2008

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Online election schemes mitigate bribery and coercion by precluding the generation of receipts that can prove how voters voted. In order to guarantee that each voter's public election data appears ambiguous, existing approaches to receipt-free schemes rely on problematic assumptions when voters cast ballots. We take a new approach by using the novel properties of the Damgård-Jurik cryptosystem to construct Masked Ballot, a receipt-free scheme that avoids such assumptions during the election. The Masked Ballot scheme assumes the existence of untappable channels for a trusted registrar to send private masking values to voters before the election, but does not require these channels during the election. Voters cast ballots over completely public channels without relying on untappability, anonymity or trusted devices.

1 Introduction

Bribery and coercion pose grave threats to the integrity of democratic elections. To prevent these forms of fraud, existing approaches to constructing online election schemes limit an adversary's knowledge during the election. We develop a new approach that eliminates the need for such limitations and results in a more practical election scheme.

The traditional secret ballot is a fundamental instrument for protecting the freedom of choice of voters. It is resistant to bribery and coercion because nobody knows whether voters are lying about how they voted. But at the same time, this secrecy prevents verifiability.

In contrast, online elections must be publicly verifiable. To achieve verifiability, the election process must expose the ballots in some way. But exposed ballots introduce the potential for voters to prove how they voted, which opens the door to large scale bribery and coercion. To stop voters from being able to prove their votes, election schemes must be *receipt-free*. A *receipt* is a set of values that identifies both the voter and the vote. These values fall into three categories:

1. Public data - All public values generated by the voter or the authorities. This includes the exposed ballot and all related public messages, such as proofs of correctness. In most schemes, the ballot is a probabilistically encrypted vote.

2. Private data - All secret values generated by the voter or privately revealed to the voter. This includes the vote and any private keys.

3. Eavesdropped data - All ostensibly non-public values that the voter sends or receives via non-untappable communication channels.

For example, consider a simple hypothetical election scheme. To cast a ballot, a voter probabilistically encrypts a vote with the public key of the authorities and posts the ciphertext to an authenticated bulletin board. The ballot is the encrypted vote, and the public data is the ballot and the public key. The private data is the vote and the randomness used to encrypt the vote. In this example, a receipt consists of all the public and private data. The authenticated posting of the ballot identifies the voter. The ballot, public key, vote and randomness identify the vote.

To prevent an election from generating receipts, the voter's private data for a valid vote must be indistinguishable from fake private data for any other valid vote. Hence the public data must appear ambiguous with respect to the private data. Cryptographic protocols provide ambiguity by using secret randomness.

The problem in the above example is that the public data is unequivocal, and so it is possible to verify whether the private data is genuine. But if

the randomness used to encrypt the vote were secret, then the public ballot would be ambiguous. Thus an obvious modification of the simple scheme is for the authorities to generate a ballot for each valid vote and keep the randomness secret. The authorities must also provide some secret information to reveal the votes to the voter. For each ballot, the authorities send the voter a zero-knowledge proof that the ballot corresponds to a particular vote. Then to cast a vote, the voter posts the desired ballot. In practice, almost all receipt-free schemes take a similar approach to provide ambiguity.

Although in this improved example the public data is ambiguous with respect to the private data, there remains a problem with potentially eavesdropped data. By intercepting all messages, a powerful adversary knows all the information that a voter knows. In this context, election schemes must consider potentially eavesdropped data to be public data. But receipt-freeness requires the voters and authorities to exchange some private data that remains secret from the adversary. This forces existing receipt-free schemes to make assumptions that limit the adversary's knowledge during the election. There are three alternative assumptions:

1. Voters send and/or receive some private data via untappable channels.
2. Voters construct ballots using trusted devices.
3. Voters cast ballots via anonymous broadcast channels.

All of these assumptions are problematic for practical online elections. Trusted devices can still be compromised, while untappable channels and anonymous broadcast channels are impractical to implement over the Internet.

A more desirable approach is for the voters to obtain some private data before the election. Then during the election all communication is public. The advantage of transmitting private data offline is that there are practical implementations of untappable channels. For example, a voter can obtain private data in person or via registered post.

This approach relies on a trusted party to maintain the secrecy of the private data. But the existence of a trusted party before the election appears to be an unavoidable assumption even in traditional elections.

1.1 Contribution

We introduce Masked Ballot, a new election scheme that achieves receipt-freeness without imposing restrictions on the adversary's knowledge during the election. Since Masked Ballot transmits no private data during the election, all communication is via public channels. The Masked Ballot scheme requires one-way untappable channels before the election but not during the election.

The idea behind Masked Ballot is to ambiguously disguise encrypted votes with private masking values. Without knowing the masking value, the public data is plausibly consistent with any vote.

Before the election, a trusted registrar sends each voter a masking value via a one-way untappable channel. The registrar also posts an encryption of the masking value.

During the election, voters use their masking values to construct masked ballots. First, the voter encrypts the vote. Then the voter subtracts the masking value from the encrypted vote to produce the masked ballot. The voter posts the masked ballot via completely public channels. The voting takes place without untappable channels, anonymous broadcast channels or trusted devices. After the voting, the tallying authorities remove the masking values from the ballots using only the encrypted masking values. Unmasking the ballots and revealing the votes is made possible by the Damgård-Jurik cryptosystem.

1.2 Organisation

First we examine existing methods for achieving receipt-freeness in election schemes in Section 2. We describe the Damgård-Jurik cryptosystem and verifiable mix-nets used in Masked Ballot in Section 3. Then we present the details of the Masked Ballot scheme in Section 4. Finally, we discuss the security properties of Masked Ballot with a focus on receipt-freeness in Section 5.

2 Related Work

The nature and practicability of the assumptions in existing online election schemes vary widely, depending on how the schemes use randomness in the voting process to provide ambiguity. There are three different approaches, each of which limits the adversary's knowledge in a different way.

The first approach [3, 5, 12, 14] requires untappable channels during the election. The authorities generate secret randomness and reveal some information about it to the voters, or vice versa. If an adversary learns the randomness, then receipt-freeness is compromised. Hence all information about the randomness must be sent via untappable channels.

Relying on untappable channels poses several problems. First, it is impractical to implement these channels over insecure public networks like the Internet. Furthermore if a voter claims that the secret values sent by an authority are invalid, then only the voter and the authority know which party is dishonest. Resolving such disputes can disrupt the entire election process.

Another concern is the extent of trust in authorities. If an authority using an untappable channel is dishonest, then an adversary can discover

the secret values sent via the channels. Hirt and Sako [5] observe that distributing the trust among multiple authorities is an unsatisfactory solution because the voter still needs to know at least one honest authority to safely lie (otherwise if the voter lies to a corrupt authority, then the adversary will know). For this reason schemes generally assume that with respect to receipt-freeness, an adversary cannot corrupt or collude with any authorities who communicate with voters via untappable channels.

The second approach [2, 7, 8, 9] requires trusted devices during the election. Voters communicate with a trusted randomiser via untappable channels. The voter sends the randomiser an encrypted vote. The randomiser changes the randomness in the ciphertext, and sends the voter the re-encrypted vote along with a proof of correctness of the re-encryption. The voter then casts the re-encrypted vote via public channels. In this approach, neither the voters nor the authorities know the randomness.

The randomisers can be implemented by tamper-resistant devices such as smart cards. The untappable channel is reduced to a local channel rather than a network communication channel. But tamper-resistant devices are generally impractical for online elections because the necessary hardware is not widespread. More concerning is the catastrophic failure model for these devices. An adversary who compromises the devices can commit large scale fraud.

The third approach requires anonymous broadcast channels during the election. The secret randomness is in a credential rather than the ballot. Voters secretly obtain their credentials in a registration stage before the election. To ensure the credentials are secret, all communication during the registration stage is via untappable channels. During the election, voters cast their ballots along with their credentials via anonymous broadcast channels. In the scheme of Niemi and Renvall [11], the voter collaborates with all the authorities to construct a credential. In the scheme of Juels et al. [6], a trusted registrar provides voters with credentials.

Using untappable channels before the election is more practical than during the election, and avoids many of the pitfalls. But implementing anonymous broadcast channels over the Internet is impractical because it requires all voters to simultaneously participate in a distributed protocol. Juels et al. point out that in contexts where a voter is restricted to voting from public locations (for example an Internet café) or an adversary cannot eavesdrop on all communication channels, the Internet provides sufficient anonymity. But in general this assumption is fairly onerous.

3 Cryptographic Primitives

Masked Ballot uses a threshold version of the Damgård-Jurik cryptosystem to encrypt messages. Masked Ballot also uses a mix-net to anonymise

encrypted votes.

3.1 Damgård-Jurik Cryptosystem

The Damgård-Jurik cryptosystem [4] is a probabilistic public key cryptosystem based on composite residuosity modulo n^{k+1} for $k \geq 1$. It is a generalisation of the Paillier cryptosystem [13], with Paillier corresponding to the case $k = 1$.

The public key is (g, n), where $n = pq$ is an RSA modulus and $g = n + 1$. A message $m \in \mathbb{Z}_{n^k}$ is encrypted by randomly generating $r \in \mathbb{Z}_n^*$ and computing the ciphertext

$$E_k(m, r) = g^m r^{n^k} \bmod n^{k+1} \in \mathbb{Z}_{n^{k+1}}^*.$$

Damgård-Jurik has several novel properties.

Additive Homomorphism

Multiplying two ciphertexts performs addition of the plaintexts. For plaintexts $m_1, m_2 \in \mathbb{Z}_{n^{k-1}}^*$,

$$\begin{aligned} E_k(m_1, r_1) \times E_k(m_2, r_2) &= \left(g^{m_1} r_1^{n^k}\right)\left(g^{m_2} r_2^{n^k}\right) \\ &= g^{m_1+m_2}(r_1 r_2)^{n^k} \\ &= E_k(m_1 + m_2, r_1 r_2). \end{aligned}$$

Scalar Homomorphism

Exponentiating a ciphertext by a scalar multiplies the plaintext by the scalar. For a plaintext $m \in \mathbb{Z}_{n^{k-1}}^*$ and scalar $s \in \mathbb{Z}_{n^{k-1}}^*$,

$$\begin{aligned} E_k(m, r)^s &= \left(g^m r^{n^k}\right)^s \\ &= g^{ms}(r^s)^{n^k} \\ &= E_k(ms, r^s). \end{aligned}$$

Multiple Layers of Encryption

Adida and Wikström [1] observe that multiple layers of encryption are possible by iteratively encrypting a message with increasing k in the modulus. Masked Ballot uses an inner layer ($k = 1$) and an outer layer ($k = 2$). Messages encrypted with both layers are double-encrypted messages $E_2(E_1(m, r_1), r_2)$.

Double Re-encryption

Another result by Adida and Wikström [1] is re-encryption of both layers of a double-encrypted message. This method uses the scalar homomorphic property to re-encrypt the inner layer. To re-encrypt a double-encrypted message $E_2\left(E_1\left(m,r\right),x\right)$ using secret randomness $s,y\in\mathbb{Z}_n^*$, compute

$$\begin{aligned} E_2\left(E_1\left(m,r\right),x\right)^{s^n}\cdot y^{n^2} &= E_2\left(E_1\left(m,r\right)\cdot s^n, x^{s^n}y\right) \\ &= E_2\left(E_1\left(m,rs\right),x^{s^n}y\right). \end{aligned}$$

Double re-encryption is publicly verifiable using honest verifier zero-knowledge proofs.

3.2 Verifiable Mix-net

A mix-net is a series of servers that each randomly mix (permute and re-encrypt) a batch of messages. As long as at least one mix server is honest, the process conceals the correspondence between input and output messages. The mixing is publicly verifiable using honest verifier zero-knowledge proofs.

The Masked Ballot scheme only mixes the inner layer of ciphertexts, and so any Paillier-based mix-net is suitable, for example the mix-net by Nguyen et al. [10].

4 The Masked Ballot Scheme

The Masked Ballot scheme has three stages: registration, voting and counting. The registration stage takes place in advance of the election, and the election itself consists of the voting and counting stages. The participants in the scheme are voters, a trusted registrar who registers voters, and multiple tallying authorities who count the votes. Apart from the use of one-way untappable channels in the registration stage, all communication is via an authenticated bulletin board.

In the *registration stage*, a trusted registrar randomly generates a private masking value for each voter. The registrar encrypts the masking value and creates a designated verifier proof to convince only the voter that the ciphertext is an encryption of the masking value. Then the registrar publicly posts the encrypted masking value and sends the masking value and proof to the voter via a one-way untappable channel. The untappable channel and designated verifier proof ensure that no other parties gain any information about the masking value.

In the *voting stage*, each voter publicly posts a masked ballot $E_1\left(v,x\right)-m$, where $E_1\left(v,x\right)$ is the encryption of the vote v with randomness x, and m is the masking value. Without knowing m, the masked ballot $E_1\left(v,x\right)-m$ appears consistent with any fake v',x',m' such that $E_1\left(v',x'\right)-m'=$

$E_1(v, x) - m$. Hence there is no way to verify the genuine private data v, x, m, and so it is safe for voters to cast their ballots over public channels.

In the *counting stage*, the authorities use the encrypted masking values to unmask the ballots. Then to ensure the secrecy of the voter's private data, the authorities introduce new randomness through re-encrypting. Finally, the authorities use a mix-net to anonymise encrypted votes before revealing the votes.

We now describe the Masked Ballot scheme in greater detail.

4.1 Initialisation

First, the tallying authorities perform the necessary initialisation steps.

1. Set up an authenticated bulletin board and establish access mechanisms for the registrar, authorities and voters.
2. Set up the threshold version of the Damgård-Jurik cryptosystem with public key (n, g). Each authority has a share of the private key.
3. Publish the public key and any system parameters.

4.2 Registration Stage

In advance of the election, the trusted registrar provides each voter with a private masking value.

1. Randomly generate a masking value $m \in \mathbb{Z}_{n^2}$.
2. Using the outer key, encrypt m with random $r \in \mathbb{Z}_n^*$ by computing

$$E_2(m, r) = g^m r^{n^2}.$$

   Construct a designated verifier proof ρ that $E_2(m, r)$ is a ciphertext of m.
3. Post the encrypted masking value $E_2(m, r)$ next to the voter's name.
4. Send the masking value m and proof ρ to the voter via a *one-way untappable channel*.

4.3 Voting Stage

Each voter constructs a ballot using a vote v and a private masking value m.

1. Using the inner layer, encrypt the vote v with random $x \in \mathbb{Z}_n^*$ by computing
$$E_1(v, x) = g^v x^n.$$
Subtract the masking value m to produce the masked ballot
$$E_1(v, x) - m.$$

2. Post the masked ballot $E_1(v, x) - m$.

The masked ballot is consistent with any fake encrypted vote $E_1(v', x')$. It is trivial for the voter to compute a suitable fake masking value

$$m' = E_1(v', x') - (E_1(v, x) - m).$$

4.4 Counting Stage

After the voting period has ended, multiple authorities collaborate to extract the votes in five steps.

1. Unmasking

The authorities combine each masked ballot $E_1(v, x) - m$ with its corresponding encrypted masking value $E_2(m, r)$ to remove the masking value.

1. Using the outer layer, *deterministically* encrypt the masked ballot by computing
$$E_2(E_1(v, x) - m, 1) = g^{E_1(v,x)-m}.$$

2. Cancel out the masking value by computing
$$\begin{aligned} E_2(E_1(v, x) - m, 1) \times E_2(m, r) &= \left(g^{E_1(v,x)-m}\right)\left(g^m x^{n^2}\right) \\ &= g^{E_1(v,x)} x^{n^2} \\ &= E_2(E_1(v, x), r). \end{aligned}$$

3. Post the encrypted masked ballot $E_2(E_1(v, x) - m, 1)$ and double-encrypted vote $E_2(E_1(v, x), r)$.

2. Re-encrypting

Each authority re-encrypts both layers of each double-encrypted vote $E_2(E_1(v, x), r)$ to conceal the encrypted vote generated by the voter.

1. Double re-encrypt $E_2\left(E_1\left(v,x\right),r\right)$ with random $s,t\in\mathbb{Z}_n^*$ by computing

$$\begin{aligned} E_2\left(E_1\left(v,x\right),r\right)^{t^n}s^{n^2} &= \left(g^{E_1(v,x)}r^{n^2}\right)^{t^n}s^{n^2} \\ &= g^{E_1(v,xt)}\left(r^{t^n}s\right)^{n^2} \\ &= E_2\left(E_1\left(v,xt\right),r^{t^n}s\right). \end{aligned}$$

2. Post this new double-encrypted vote $E_2\left(E_1\left(v,xt\right),r^{t^n}s\right)$ along with a proof of correctness of the double re-encryption.

To simply the notation, let s and t denote the products of the respective random values generated for the double-encrypted vote by all the authorities. Provided at least one authority keeps its random values secret, s and t remain secret.

3. Revealing the encrypted votes

For each double-encrypted vote $E_2\left(E_1\left(v,xt\right),r^{t^n}s\right)$, a quorum of authorities decrypts the outer layer and publicly posts the encrypted vote $E_1\left(v,xt\right)$ along with a proof of correctness of threshold decryption.

4. Mixing

A mix-net shuffles all the encrypted votes of the form $E_1\left(v,xt\right)$. The output is a list of permuted and re-encrypted votes of the form $E_1\left(v,xtu\right)$, where $u\in\mathbb{Z}_n^*$ is the secret randomness introduced by the mix-net. These shuffled encrypted votes cannot be linked to the voters.

5. Revealing the votes

For each encrypted vote $E_1\left(v,xtu\right)$, a quorum of authorities decrypts the ciphertext and publicly posts the vote v along with a proof of correctness of threshold decryption.

4.5 Complexity

We now provide rough estimates of the computation and communication complexity.

Each voter posts $O\left(1\right)$ bits and performs $O\left(1\right)$ modular exponentiations in the voting stage. The voter submits a single message, and so the round complexity is 1, as required for any practical election scheme.

For V voters, each authority posts $O\left(V\right)$ bits and performs $O\left(V\right)$ modular exponentiations in the counting stage. The dominating operations in the complexity are the mixing, threshold decryptions and re-encryptions. All of these operations are linear in V.

5 Security Properties

The Masked Ballot scheme satisfies the common security properties for online elections.

Receipt-Freeness

Voters cannot prove how they voted. Although we are yet to construct a formal proof of receipt-freeness, we provide an informal explanation.

After the mixing step in the counting stage, the votes are identifiable but are no longer linked to the voters. We assume that every possible voting option is likely to receive votes from honest voters. Hence the votes themselves reveal no information that can identify the voters. This assumption is reasonable when the number of voting options is very small compared to the number of voters.

Conversely, before the mixing step the public data is linked to the voters, but the vote is not identifiable. In the steps before the mixing, the relevant public data consists of:

1. The encrypted masking value $E_2(m, r)$.
2. The masked ballot $E_1(v, x) - m$.
3. The results of operations performed by the authorities:
   $E_2(E_1(v, x) - m, 1)$, $E_2(E_1(v, x), r)$, $E_2(E_1(v, xt), r^{t^n} s)$ and $E_1(v, xt)$.

The voter's private data is the vote v, the randomness x used to encrypt the vote, and the masking value m.

Each individual public value is ambiguous with respect to the private data. The public ballot $E_1(v, x) - m$ is ambiguous because for any fake private data v', x', m' such that $E_1(v', x') - m' = E_1(v, x) - m$, the fake ballot $E_1(v', x') - m'$ is in fact a genuine ballot for a plausible masking value m'. For any given v', x', there is a suitable $m' = E_1(v', x') - (E_1(v, x) - m)$.

Apart from the ballot, all of the public values are ciphertexts. Each ciphertext is ambiguous because the Damgård-Jurik cryptosystem satisfies indistinguishability under chosen plaintext attack and the randomness r, s, t remains secret.

Since these individual public values are ambiguous, what remains to be shown is that combining all the public values reveals no additional information about the private data. From our preliminary results, this appears to be true.

Public Verifiability

Any observer is confident that the revealed votes are consistent with the ballots cast. Through the use of honest verifier zero-knowledge proofs, an

observer can verify the correctness of every operation performed on the posted messages.

Eligibility and Uniqueness

Only eligible voters participate and each voter has only one vote. The bulletin board provides authentication, thus ensuring eligibility and uniqueness.

Completeness and Soundness

All valid votes and no invalid votes contribute to the result. As every vote is revealed, any observer can count the valid votes. As there is a negligible probability that an invalid ballot corresponds to a valid vote, no invalid votes contribute to the result.

Robustness

An adversary compromises or disrupts the election only by corrupting a threshold of authorities. Through public verifiability, multiple authorities and the use of threshold decryption, the election tolerates a minority of corrupt authorities.

6 Conclusion

The Masked Ballot scheme shifts the problematic exchange of private data from the election to a registration stage in advance of the election. This enables the election to take place over public channels and without the assistance of trusted devices. As a result, Masked Ballot is suitable for receipt-free elections over the Internet.

Although Masked Ballot is receipt-free, it still does not completely guarantee a voter's freedom of choice. Receipt-freeness only deals with proving valid votes, but an adversary can exclude a voter from an election by verifying that the voter abstains or casts an invalid or random vote. Juels et al. [6] consider these attacks in the stronger property of coercion-resistance, which subsumes receipt-freeness. But note that coercion-resistance is not possible in any compulsory election. After the election, the authorities issue penalty notices to abstaining voters. These penalty notices are receipts that provide undeniable evidence of abstention.

The minimum assumption for coercion-resistance is that voters cast ballots via anonymous broadcast channels. Otherwise, the adversary can determine who voted. But anonymous broadcast channels are impractical for voting over the Internet. Another issue is that in order to ensure the identities of participating voters remain secret the bulletin board must be unau-

thenticated, which makes the election more vulnerable to denial of service attacks.

Masked Ballot does not achieve coercion-resistance because the authenticated bulletin board identifies voters. Therefore an adversary with only publicly available knowledge can verify that a voter abstains or casts a specified random value as the ballot. Such an adversary can also verify that a voter casts a specified invalid vote by checking the list of revealed votes.

Comparing the weaknesses in Masked Ballot with Juels et al.'s coercion-resistant scheme [6] raises some interesting questions. To what extent can we guarantee a voter's freedom of choice against an adversary who can intercept and trace all public communication during the election? Against which classes of adversaries can we achieve coercion-resistance? These issues are interesting avenues for future research.

Bibliography

[1] Ben Adida and Douglas Wikström. How to shuffle in public. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 555–574. Springer, 2007.

[2] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *PODC*, pages 274–283, 2001.

[3] Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, pages 544–553, 1994.

[4] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136. Springer, 2001.

[5] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 539–556. Springer, 2000.

[6] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In Vijay Atluri, Sabrina De Capitani di Vimercati, and Roger Dingledine, editors, *WPES*, pages 61–70. ACM, 2005.

[7] 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, *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 245–258. Springer, 2003.

[8] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In Pil Joong Lee and Chae Hoon Lim, editors, *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 389–406. Springer, 2002.

[9] Emmanouil Magkos, Mike Burmester, and Vassilios Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channels. In Beat Schmid, Katarina Stanoevska-Slabeva, and Volker Tschammer, editors, *I3E*, volume 202 of *IFIP Conference Proceedings*, pages 683–694. Kluwer, 2001.

[10] Lan Nguyen, Reihaneh Safavi-Naini, and Kaoru Kurosawa. Verifiable shuffles: a formal model and a paillier-based three-round construction with provable security. *Int. J. Inf. Sec.*, 5(4):241–255, 2006.

[11] Valtteri Niemi and Ari Renvall. How to prevent buying of votes in computer elections. In Josef Pieprzyk and Reihaneh Safavi-Naini, editors, *ASIACRYPT*, volume 917 of *Lecture Notes in Computer Science*, pages 164–170. Springer, 1994.

[12] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors, *Security Protocols Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 25–35. Springer, 1997.

[13] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.

[14] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *EUROCRYPT*, pages 393–403, 1995.