An Efficient Mixnet-based Voting Scheme Providing Receipt-Freeness ⋆

Riza Aditya[1], Byoungcheon Lee[1,2], Colin Boyd[1], and Ed Dawson[1]

[1] Information Security Research Centre,
Queensland University of Technology,
GPO BOX 2434, Brisbane, QLD, 4001, Australia
{r.aditya, b6.lee, c.boyd, e.dawson}@qut.edu.au
[2] Joongbu University,
101 Daebak-Ro, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea
sultan@joongbu.ac.kr

Abstract. Receipt-freeness is an essential security property in electronic voting to prevent vote buying, selling or coercion. In this paper, we propose an efficient mixnet-based receipt-free voting scheme by modifying a voting scheme of Lee *et al.* The receipt-freeness property is obtained through the randomization service given by a trusted administrator, and assuming that two-way untappable channel is used between voters and the administrator. The efficiency is improved by employing a more efficient mixnet, which is a modification of Golle *et al.*'s optimistic mixnet. In the proposed scheme, the administrator provides both randomization (ballot re-encryption) and mixing service in the voting stage. Afterward, the ballots are mixed using the proposed efficient mixnet. Our mixnet-based voting scheme offers receipt-freeness in an efficient manner.

Keywords: Electronic voting, receipt-freeness, mixnet, re-encryption, randomization, designated-verifier re-encryption proof.

1 Introduction

Voting is often related to political and financial gain, and cheating is an inherent threat to voting. Thus, security aspects in voting must also be thoroughly considered. This results in extensive security requirements for e-voting.

- **Privacy:** Normally, the vote is encrypted prior to submission, where the ballot is in the form of an encrypted vote. Voter-vote relationship must be kept private, to ensure that voters express their true opinion without fear of being intimidated.
- **Eligibility:** Only authorized voters are allowed to vote, preventing fraudulent votes from being counted in the tally stage.

⋆ This is the full paper containing illustration of the proposed protocol. The proceeding version of this paper will be published in *Trust and Privacy in Digital Business—TrustBus 04*, volume ?? of LNCS, pages ??-??, 2004

- **Prevention of double voting:** This ensures that all voters are allowed to vote only once, such that each voter has equal power in deciding the outcome of the voting.
- **Fairness:** No partial tally is revealed before the end of the voting period, to enforce privacy and ensure that all candidates are given a fair chance during the voting period.
- **Receipt-freeness:** Introduced by Benaloh and Tuinstra [4], voters must neither be able to obtain nor construct a receipt which can prove the content of their vote to a third party. This is to prevent vote selling/buying, ensuring that voters are not used as a proxy to cast votes.
- **Robustness:** The system must be able to tolerate certain faulty conditions by managing some disruptions.
- **Verifiability:** Correct voting processes must be verifiable to prevent incorrect voting result.

Secret-ballot e-voting schemes typically employ either mixnet or homomorphic encryption to provide voters privacy. Our proposed scheme employs mixnet since it offers more flexibility on the ballot structure as opposed to employing homomorphic encryption, e.g. in preferential voting [3].

To provide receipt-freeness, many schemes normally employ a trusted authority to randomize the ballot prior to vote submission stage. Many of the schemes are oriented toward homomorphic encryption approach since accumulation of votes is obtained by decrypting combination of the ballots, where individual ballots are never decrypted. Providing receipt-freeness in mixnet-based voting schemes is more problematic since all ballots are decrypted individually for tallying, and a voter can prove the content of his ballot using his knowledge of the random value used to construct his ballot (encrypt his vote).

Obtaining both receipt-freeness and efficiency, both the receipt-free mixnet-based voting scheme of Lee *et al.* [9] and the optimistic mixnet scheme of Golle *et al.* [8] are modified, and then combined as follows:

- the administrator provides both the re-encryption service (by the tamper-resistant hardware randomizer in [9]) and mixing service (by the first mix server in [8]) together,
- the administrator is trusted not to collude with the mix servers to reveal voter-vote relationship, and
- the communication channel between the administrator and the voter is untappable.

The remainder of the paper is organized as follows. Section 2 provides more background and motivation to our proposed scheme. Reviews of the mixnet-based voting scheme in [9] and the optimistic mixnet scheme in [8] are provided in more detail. Section 3 describes the modification made to the optimistic mixnet scheme to provide receipt-freeness and cancel known attacks to it. Section 4 presents our proposed efficient mixnet-based receipt-free voting scheme using the proposed efficient mixnet. Section 5 analyses the security and efficiency of the proposed scheme. Section 6 is a conclusion.

2 Related Work

Based on the verification of mixing, mixnet schemes are classified into verifiable mixnet and optimistic mixnet. **Verifiable mixnets** [1, 7] offer robustness at the cost of efficiency. Proof of correct mixing is accurate and requires more computation and bandwidth compared with optimistic mixing. **Optimistic mixnets** [5, 8] offer efficiency at the cost of robustness. Proof of correct mixing is quite simple, though less accurate, compared with the verifiable ones. Confidence of correct mixing provided by optimistic mixing is less than that offered by verifiable mixnet schemes. However, it is much more efficient.

Schemes using verifiable mixnet can be made more efficient by employing an optimistic mixnet. We recall a mixnet-based receipt-free voting scheme in the following subsection, and recall an optimistic mixnet scheme in the subsection afterward.

2.1 Mixnet-based Receipt-free Voting Scheme by Lee *et al.*

The mixnet-based receipt-free voting scheme by Lee *et al.* [9] focuses on removing user-chosen randomness in ballots to provide receipt-freeness. This is achieved as ballots are randomized by a third party. In their scheme, a tamper-resistant hardware device named tamper-resistant randomizer (TRR) is used to act as the third party randomizer and also provide untappable channel. Correct re-encryption by the randomizer is verifiable by the use of designated verifier re-encryption proof (DVRP). The re-encrypted ballots are anonymized by the mixnet, and the outputs of the mixnet are individually decrypted by a quorum of decryption authorities.

The voting stage consists of four sub-stages. First, each voter prepares a *first ballot* by encrypting his vote. The ballot is then sent to TRR for randomization. Second, the TRR randomizes the first ballot with re-encryption to produce a *final ballot*. Third, the TRR also produces a *Designated Verifier Re-encryption Proof* (DVRP) to prove the correctness of re-encryption to the voter. The final ballot and the DVRP are then sent to the voter. Finally, the voter checks the DVRP, then signs and *submits the final ballot* if the check is accepted.

As the scheme employs verifiable mixnet, efficiency improvement is possible by alternatively using an optimistic mixnet.

2.2 Optimistic Mixnet by Golle *et al.*

Golle *et al.* [8] proposed a very efficient mixnet scheme using the optimistic approach. Correct mixing is proved by using the *proof of product* (POP), proving that the product of input messages is preserved in the product of the output messages. The proof of product exploits the homomorphic property of the underlying ElGamal encryption scheme. However, a checksum is required to verify the integrity of the messages. Also, the inputs are required to be encrypted twice, named *double enveloping*, to support backup mixing.

Double enveloping protects the anonymity of the original sender from a relation attack by a dishonest server. When the input message is encrypted only once, a dishonest server can modify its output by multiplying two inputs b_i and $b_{i'}$ and outputs the re-encryptions of $b_i b_{i'}$ and 1, where $i \neq i'$. This attack passes the proof of product test. By observing the attacked (combined) plaintext output after decryption, the related ciphertexts can be identified. Double encryption is used to prevent such attack, so that when the first mixing for the outer encryption is found to be incorrect, the inner encrypted messages are recovered by the decryption authorities and mixed again using a more robust, heavy-weight verifiable mixnet.

Based on the scheme by Pedersen [10], a threshold version of ElGamal cryptosystem is employed with properly generated parameters as in [10], private key x, and public key $(g, y = g^x)$. Several decryption authorities share the private key x using Shamir's (t, m) secret sharing scheme [11]. A message v is encrypted with a random value r using an encryption function E and the public key y as $E_y(v, r) = (\alpha = g^r, \beta = vy^r)$. A collision-resistant hash function H is used to produce the hash checksum as $h = H(\alpha, \beta)$. The double encrypted ciphertext is then produced with different random values r_1 and r_2 as $E_y(\alpha, r_1), E_y(\beta, r_2)$ and the hash checksum is also encrypted with a random value r_3 as $E_y(h, r_3)$. For n messages $(i = 1, \ldots, n)$, inputs to the mixnet is a triple of the form $(E_y(\alpha_i, r_{i,1}), E_y(\beta_i, r_{i,2}), E_y(h_i, r_{i,3}))$, where $h_i = H(\alpha_i, \beta_i)$.

The mixnet scheme is a basic re-encryption mixnet, where each mix server receives inputs (α_i, β_i, h_i), re-encrypts them by selecting different random values $r'_{i,1}, r'_{i,2}, r'_{i,3}$ and compute $(\alpha'_i, \beta'_i, h'_i)$, and outputs them in a random order. Afterward, the mix server proves the preservation of product of messages in the mixing (proof of product) by proving:

$$\prod \alpha_i = \prod \alpha'_i \ \wedge \ \prod \beta_i = \prod \beta'_i \ \wedge \ \prod h_i = \prod h'_i \tag{1}$$

Computational complexity (in terms of modular exponentiations) using this technique is independent of n, the number of messages.

After the mixing is finished, each output is decrypted using threshold decryption by a quorum of decryption authorities. The final output of the mixnet are triplets in the form of $(\alpha'_{\pi(i)}, \beta'_{\pi(i)}, h'_{\pi(i)})$, where $\pi(i)$ represents the result of total permutation of i. The integrity of each result is also verified by checking:

$$h'_{\pi(i)} = H(\alpha'_{\pi(i)}, \beta'_{\pi(i)}) \text{ for } i = 1, \ldots, n \tag{2}$$

Recent research revealing possible attacks on this mixnet scheme include the paper by Abe and Imai [2] and Wikström [12].

3 Proposed Efficient Mixnet Scheme

To provide the required receipt-freeness property in the proposed voting scheme and to eliminate attacks as in [2, 12], we apply the following modifications to the scheme by Golle *et al.* [8].

- The hash checksum is removed to invalidate the relation attacks as in [2, 12].
- Single encryption is used instead of double encryption to prevent a sender from using the inner encryption of the double enveloping as a receipt.
- We only check that $\prod v_i = \prod v'_i$ in the proof of product, where v_i and v'_i are messages before and after the mixing.

Threshold version of ElGamal cryptosystem is employed as in Section 2.2. The two primes are p and $q|p-1$, the secret key is x, and the public key is $(g, y = g^x)$. Assume that there are n voters V_i where $i = 1, \ldots, n$. Each voter V_i interacts with the administrator to generate a ciphertext (α_i, β_i) for his vote v_i (will be detailed in Section 4). These ciphertexts are input to the mixnet.

The proposed mixnet protocol works as follows:

1. Re-encrypt and randomly permute the ordering of messages:
   Each mix-server receives n input ciphertexts (α_i, β_i). Choosing random values $r_i \in_R \mathbb{Z}_q$, the ciphertexts are re-encrypted as $(\alpha'_i, \beta'_i) = (\alpha_i g^{r_i}, \beta y^{r_i})$. The mix-server then outputs the re-encrypted ciphertexts in a random order $(\alpha'_{\pi(i)}, \beta'_{\pi(i)})$, where $\pi(i)$ is a random permutation of i.
2. Prove preservation of products (individual mix server verification):
   Each mix-server proves the following equation in zero knowledge.

$$\log_g \frac{\prod_{i=1}^{n} \alpha'_i}{\prod_{i=1}^{n} \alpha_i} = \log_h \frac{\prod_{i=1}^{n} \beta'_i}{\prod_{i=1}^{n} \beta_i} \tag{3}$$

   The correctness of the mixing is verifiable by anyone as g, y, and input (α_i, β_i) and output ballots (α'_i, β'_i) are made public. This zero-knowledge proof requires 2 exponentiations for proving and 6 exponentiations for verification using the Chaum-Pedersen protocol [6].

If the mixnet is highly trusted, a variation named *global verification* can be used. This verification technique takes a more optimistic approach as the preservation of product is verified, not by the mix servers in each mixing stage, but by the decryption authorities after all mixings are finished. The decryption authorities decrypt the product of the first input ballots to the mixnet and the product of the last output ballots from the mixnet, and check the equality of these two values.

Individual mix server verification offers early detection of error in the mixing. Thus, mixing can be aborted and done by other mix servers. This verification technique is preferable as it provides a correctness check on each mix server. Using global verification, each mix server is not required to produce any proof. Thus, mixing process can be performed more efficiently, however errors will only be detected when the proof of products are decrypted.

Our proposed mixnet scheme uses a single encryption removing the use of double encryption and hash checksum (3 encryptions). Thus attacks [2, 12] on the original mixnet scheme [8] are not applicable to our scheme, while efficiency is improved three times.

4 Proposed Voting Scheme

Our efficient mixnet-based receipt-free voting protocol uses the proposed optimistic mixnet as described in Section 3. The parameters p, q, g, y are made public, while x is kept secret. Each voter registers to a registration authority and obtains a public-private key pair through an already established key distribution mechanism such as Public Key Infrastructure (PKI). Illustrated in Figure 1, the voting protocol consists of the following three stages:

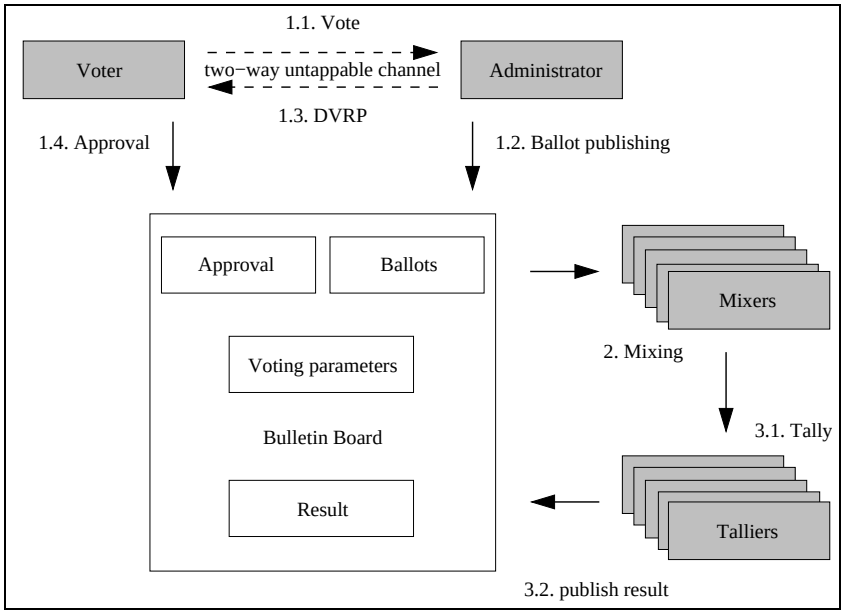

Fig. 1. The Proposed Voting Protocol.

Stage 1. Voting
Voting stage is an interactive protocol between the voters and the administrator through an untappable channel. During the actual voting period, votes are cast by the voters, published by the administrator and approved by the voters.

1. Vote casting (using two-way untappable channel):
   Each voter V_i chooses and encrypts his vote v_i as $(\alpha_i, \beta_i) = E_y(v_i, r_i)$, where r_i is a random value chosen by the voter. The encrypted vote (α_i, β_i) is then sent to the administrator with voter's signature. The administrator checks the eligibility of the voter and the validity of voter's signature.
2. Ballot publishing:
   After the voting period finishes, the administrator re-encrypts each ballot

using a new random value τ_i as $(\alpha'_i, \beta'_i) = (\alpha_i g^{\tau_i}, \beta_i h^{\tau_i})$ [3], and posts the re-encrypted vote (α'_i, β'_i) in a random order on the bulletin board.

3. DVRP (using two-way untappable channel):
   The administrator provides each voter with a DVRP which proves the correctness of the re-encryption. Using DVRP, the administrator proves personally to the voter that he knows either the random value τ_i or the private key of the voter x_i (public key of voter is $y_i = g^{x_i}$) as the following:
   (a) The prover selects random values of $k, r, t \in_R \mathbb{Z}_q$.
   (b) The prover computes commitments of $(a, b) = (g^k, y^k)$ and $d = g^r y_i^t$.
   (c) The prover computes the challenge using a one-way collision-resistant hash function H as $c = H(a, b, d, \alpha'_i, \beta'_i)$.
   (d) The prover then calculates the response $u = k - \tau_i(c + r)$.
   (e) The prover sends (c, r, t, u) to V
   (f) The verifier checks:

$$c \stackrel{?}{=} H(g^u(\alpha'_i/\alpha_i)^{c+r}, y^u(\beta'_i/\beta_i)^{c+r}, g^r y_i^t, \alpha'_i, \beta'_i) \tag{4}$$

4. Approval:
   Each voter checks the validity of the DVRP (Equation 4) and posts an approval message with his signature on the bulletin board if the DVRP is accepted, and refutes otherwise. The approval message format can be pre-agreed in the system such that it is fresh but not include any personal information which can be used as a receipt. For example, voters can sign the hash value of all the published ballots.

Stage 2. Mixing
The input ballots are re-encrypted and outputted in a random order by the mix servers using the proposed efficient mixnet described in Section 3. Depending on the confidence level of the voting process, an individual mix server verification or a global verification can be employed.

Stage 3. Tally
During this stage, votes are tabulated by the talliers and the result is published on the bulletin board.

1. The output of the mix-network are individually decrypted by a quorum of talliers using threshold decryption. The threshold decryption is publicly verifiable as each tallier proves that his decryption share is correct.
2. The voting result is published by the talliers on the bulletin board.

If an invalid vote (not in pre-determined format) is found after decryption, the particular output can be traced back to identify the entity who had invalidated it. This can either be a mix-server, the administrator or the voter.

Trace-back protocol:

[3] The re-encryption exploits the homomorphic property of ElGamal cryptosystem. In the re-encryption, the random value r of the original ciphertext is changed by τ to be $r + \tau$. Thus, the re-encrypted ciphertext will still decrypt to v.

1. The last mix-server is required to reveal the i-th input corresponding to the $\pi(i)$-th invalid output, and prove the correctness of his re-encryption by revealing his random number. This process is repeated to all mix-servers in the reverse order of mixing until an invalid mixing is found.
2. If mixing was found to be correct, the administrator is required to reveal the corresponding input and output re-encryption, and prove the re-encryption by revealing the random number.
3. If the re-encryption by the administrator was found to be correct, the voter is identified to submit the invalid vote.

5 Analysis

5.1 Security

Our proposed voting scheme is based on known building blocks whose security properties are already known. This section discusses the security of our mixnet scheme and the overall security of our mixnet-based voting protocol. We analyse our proposed scheme based on the security requirements in Section 1.

- **Privacy:** The ballots are randomized and mixed first by the administrator and then by the mix servers. If at least one of these entities remains honest, privacy of voters is kept. A threat in privacy can occur when a specific invalid ballot is traced back to the voter. If the invalid ballot is traced back only to the mix servers, privacy is kept since we assume that the administrator does not disclose the voter-vote relationship.
- **Eligibility:** The list of eligible voters are made public and only authenticated voters are allowed to participate.
- **Prevention of double voting:** Voters can vote only once since they participate in voting with their signature. Any misbehaviour by the administrator, for example, deletion or addition of ballot, is prevented, since voter's approval is required to be a valid ballot.
- **Fairness:** Since we assume the threshold trust for the talliers, no partial tally is revealed. This guarantees the fairness of voting.
- **Receipt-freeness:** Since voter's ballot is randomized additionally by the administrator, a voter loses his knowledge of the randomness of the encrypted ballot and cannot construct any receipt. Also the voter cannot transfer the DVRP of the administrator to any third party, since it is a personal proof and the voter can open it in any way using his private key. Since a two-way untappable channel is used between the voter and the administrator, a buyer cannot observe the communication between the voter and administrator during the voting stage.
- **Robustness:** Using individual mix server verification, backup mixing is possible when an invalid mixing in the proof of product is detected.
- **Verifiability:** In the voting stage a voter can personally verify the correctness of administrator's randomization by checking the DVRP. Correct mixing

operation is publicly verifiable as anyone can observe and verify the equality of the product of input and output ballots. The tally stage is publicly verifiable.

A corrupt mix server can disrupt the voting by invalidating some ballots when he mixes the ballots. For example, a mix server takes two messages c_i and c_j, with $i \neq j$, and produces two output messages which are re-encryptions of 1 and $c_i c_j$. As the product of messages is still preserved, the proof of correct mixing is accepted, but recovered messages are invalid. However, the cheating mix-server will be identified using the trace-back protocol and be punished. When a trace-back occurs to a specific mix server in the middle of the mix servers, the voter-vote relationship will not be revealed. When an invalid ballot is traced back to the first mix server, the administrator will know the voter-vote relationship. Thus, we assume that the administrator is a reputable entity and does not disclose his knowledge when a trace-back occurs. The mix servers can easily perform this invalidation attack, but they cannot obtain any useful information unless they can collude with the administrator, while their identity can be easily found through a public trace-back protocol. Compared to the current manual paper-based voting, although our scheme may not offer improvement for anonymity control, it provides better protection against fraudulent votes.

5.2 Efficiency

Compared with Golle *et al.*'s scheme, our voting scheme is more efficient both in computational (number of exponentiations) and communication (message size in bits) complexity as shown in Table 1. The efficiency mainly comes from the fact that our scheme uses single encryption, while the scheme by Golle *et al.* [8] uses three encryptions for the double enveloping.

In the voting stage, our scheme requires each voter to encrypt the vote once (2 modular exponentiations), submit it to the administrator, and later verify DVRP from the administrator (6 modular exponentiations). The scheme by Golle *et al.* [8] requires each voter to perform double encryption (8 modular exponentiations). We do not compare the cost for digital signature, since it is an essential operation and requires the same cost.

In the mixing stage, our scheme requires three times less computation compared with the scheme by Golle *et al.* [8], since our scheme uses single encryption while [8] uses three encryptions for the double enveloping. In terms of proof of product (POP), our scheme requires three times less computation, if we use the individual mix server verification. If we use the global verification (Section 3), our scheme is much more efficient, since only the initial input product and final output product are decrypted by a quorum of decryption authorities and compared.

In the tally stage, our scheme only requires one threshold decryption for each ballot, where the scheme by Golle *et al.* [8] requires four threshold decryption.

Ballot size in our scheme is $2|p|$ bits as we use single ElGamal encryption, and the DVRP by the administrator is $4|q|$ bits. Ballot size in the scheme by

Table 1. Comparison of computational and communication efficiency of our scheme against Golle *et al.*'s scheme, where n is the number of voters.

Computational efficiency			Proposed	Golle *et al.* [8]
Voting	Voter	Encrypt	2	8
		Verify (DVRP)	6	N/A
	Admin	Re-encrypt	2	N/A
		Prove (DVRP)	4	N/A
Mixing	Mixer	Re-encrypt	$2n$	$6n$
		Prove	2	6
	Public	Verify	6	18
Communication efficiency			**Proposed**	**Golle *et al.* [8]**
Voting	Voter	Encrypt	$2\|p\|$	$6\|p\|$
	Admin	Proof (DVRP)	$4\|q\|$	N/A
Mixing	Mixer	Re-encryption	$2n\|p\|$	$6n\|p\|$
		Proof	$2\|p\| + \|q\|$	$6\|p\| + 3\|q\|$

Golle *et al.* [8] is $6|p|$ bits as they use double encryption. In the mixing stage, our scheme requires three times less bandwidth compared with the scheme by Golle *et al.* [8]. However, in the voting stage our scheme requires interactive communication between voters and the administrator since voters have to cast ballot first and approve it later.

6 Conclusion

An efficient mixnet-based voting scheme providing receipt-freeness has been presented. We successfully combined two mixnet-based voting schemes by Lee *et al.* [9] and Golle *et al.* [8] to provide both efficient mixing and receipt-freeness together. In our scheme, the administrator provides both randomization service and mixing service together in the voting stage. Our proposed optimistic mixnet is more light-weight because single encryption is used. Although it is more optimistic and invalidation attack by mix servers is possible, public trace-back procedure discourages any misbehaviour by the administrator or the mix servers. Because of its efficiency, the proposed voting scheme can be preferred in practical real world election applications such as political elections in which the administrator is considered to be a reputable entity and a timely tally is required. Moreover this mixnet-based voting scheme can offer more flexibility on the ballot structure, such as preferential voting.

Two major problems of our scheme are the trust assumption on the administrator and the possibility of invalidation attack by mix servers, although any misbehaviour causing invalidation can be traced back easily. Our future work will be focused on solving these problems.

7 Acknowledgements

We acknowledge the support of the Australian government through ARC Linkage International fellowship 2003, Grant No: LX0346868.

References

1. Masayuki Abe. Mix-networks on permutations networks. In *Advances in Cryptology—ASIACRYPT 99*, pages 258–273, 1999.
2. Masayuki Abe and Hideki Imai. Flaws in some robust optimistic mix-nets. In *Advances in Cryptology—ACISP 03*, pages 39–50, 2003.
3. Riza Aditya, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Secure e-voting for preferential elections. In *Second International Conference, EGOV 2003*, pages 246–249, 2003.
4. Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 544–553, 1994.
5. Dan Boneh and Philippe Golle. Almost entirely correct mixing with applications to voting. In *9th ACM Conference on Computer and Communications Security—CCS 02*, pages 68–77, 2002.
6. David Chaum and Torben Pryds Pedersen. Wallet databases with observers. In *Advances in Cryptology—CRYPTO 92*, pages 89–105, 1993.
7. Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. In *Advances in Cryptology—CRYPTO 01*, pages 368–387, 2001.
8. Philippe Golle, Sheng Zhong, Dan Boneh, Markus Jakobsson, and Ari Juels. Optimistic mixing for exit-polls. In *Advances in Cryptology—ASIACRYPT 02*, pages 451–465, 2002.
9. Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *Information Security and Cryptology—ICISC 03*, pages 245–258, 2004.
10. Torben P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In *Advances in Cryptology—EUROCRYPT 91*, pages 522–526, 1991.
11. Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.
12. Douglas Wikström. How to break, fix and optimize "optimistic mix for exit-polls". Technical report, Swedish Institute of Computer Science, 2002. Available from `http://www.sics.se/libindex.htlm`, last accessed 08 October 2003.