A New Receipt-Free E-Voting Scheme Based on Blind Signature (Abstract)

Zhe Xia
University of Surrey
z.xia@surrey.ac.uk

Steve Schneider
University of Surrey
s.schneider@surrey.ac.uk

May 25, 2006

Abstract

Electronic voting has attracted much interest recently and a variety of schemes have been proposed. Generally speaking, all these schemes can be divided into three main approaches: based on blind signature, based on mix networks and based on homomorphic encryption. Schemes based on blind signature are thought to be simple, efficient, and suitable for large scale elections. Fujioka, Okamoto and Ohta introduced a scheme typical of this approach in 1992. This scheme achieved a number of attractive properties, however, it did not provide receipt-freeness and public verifiability. Later, Okamoto extended this work to provide receipt-freeness and public verifiability, but the later work lost a useful property of the original scheme: each voter can only verify the ballot recording process but not the ballot counting process any more. In this paper, we propose a simple and efficient method, applying the secret ballot technique introduced by the Prêt á Voter scheme to improve individual verifiability to the later work. To the best of our knowledge, our scheme is the only receipt-free scheme in which voters can verify both the ballot recording process and the ballot counting process, and our scheme provides some mechanisms for honest voters to accuse dishonest authorities during the election process.

1 Introduction

Electronic voting has attracted much interest recently and a variety of schemes have been introduced. Some papers [14, 13] give a good general introduction to different e-voting approaches, their mechanisms, security requirements, and so on. In this paper, we will not repeat them, but only focus on three security properties, receipt-freeness, individual verifiability and no-cheating, which are the focus of this paper.

Receipt-freeness is related to privacy, but it is much stricter than privacy. It ensures that even if a voter wants to, he can not show others how he voted. Privacy and receipt-freeness guarantee that the voters can vote without coercion. Our definition of individual verifiability is stricter than in most existing schemes (except [5]). We require that the system could not only allow each voter to verify whether her ballot is correctly recorded as intended, but also whether her ballot is correctly counted as recorded. If honest voters find out that some authorities are cheating in the election process, the no-cheating property ensures the system provides voters with some mechanisms to accuse the cheating authorities. But dishonest voters cannot successfully accuse honest authorities.

Electronic voting schemes based on blind signature [5, 20, 21, 19] are thought to be simple, efficient, and suitable for large scale elections. The involved parties in these schemes are the voters, the administrator, the counter and the bulletin board. The main process is as follows: at first, a certain voter generates her ballot form with her choice v, encrypting it by bit-committment $\{v\}_k$ and blind signature $\{\{v\}_k\}_{blind}$. Then she sends it to an administrator. The administrator will only sign the ballot if this voter is eligible and has not applied before. When the voter receives the signed ballot $\{\{\{v\}_k\}_{blind}\}_{sign}$ from the administrator, she will unblind it $\{\{v\}_k\}_{sign}$ and send it to the counter anonymously through an anonymous channel. Normally, the anonymous channel is implemented by mix networks. As follows, the counter checks whether the ballot contains the administrator's signature. If yes, the counter will put it $\{v\}_k$ onto a bulletin board which can be read by everyone, otherwise, he will reject this ballot. Now, the voter can verify whether her ballot $\{v\}_k$ is correctly displayed on the bulletin board. If no, she can accuse the counter to a trusted third party. Otherwise, she will send her de-committment key k to the counter anonymously after some designated time T. Finally, the counter decrypts each ballot v and put them onto the bulletin board.

Compared to the other two approaches, e-voting schemes based on blind signature have several advantages: first, before the voter sends the de-committment key to the counter, the fairness and voter's privacy are unconditionally maintained even if all authorities colluded together. Second, the scheme introduced by Fujioka et al. [5] can let voters not only verify the ballot recording process but also the ballot counting process. And to the best of our knowledge, it is the only existing scheme which achieves our definition of individual verifiability, but it has not achieved receipt-freeness and public verifiability.

However, in blind signature based schemes, the voter has to be there during the whole election process. Besides, if the anonymous channel is implemented by mix networks, the communication and computation complexity will be another disadvantage.

In all receipt-free election schemes, the receipt-freeness is achieved by the assumption of an existing untappable channel which is a physical apparatus by which voters and voting authorities can communicate, perfectly secure from all other parties. The voting booth and some tamper-resistant techniques can be considered as untappable channels as well. Hirt and Sako [8] suggest that an untappable channel from voting authorities to voters has the weakest physical assumption. However, in all e-voting schemes based on blind signature, receipt-freeness is achieved by untappable channels from voters to voting authorities.

Although the receipt-free property has been achieved in a lot of schemes, in most of these schemes, there are two conflicts that have not been solved.

- The first conflict is between individual verifiability and receipt-freeness. In most of the receipt-free schemes, the system only allows the voters to verify the ballot recording process but not the ballot counting process. This is not enough, the ballots can be altered during storage or transit, and they could be lost or left out of the tally [18]. The reason of this conflict is that if the voter can verify her result directly and has some proof to accuse the dishonest authority, the voter can also prove the proof to coercers and let them to verify her result as well. Therefore, receipt-freeness is violated. Recently, some schemes [8, 16, 15, 14] have tried to use non-transferable proof, such as *Designated Verifier Proofs* [12]. However, the proof in these schemes can only be used to verify the ballot recording process. The ballot counting process can only be ensured by public verifiability.

- The second conflict is between receipts and receipt-freeness. Most of the latest schemes suggest providing each voter with a receipt. It can not only make the voter feel more confident about the election system but also give the voter a mechanism to accuse the authority if the authority is cheating. However, in most of the election schemes based on mix networks and homomorphic encryption, the receipt can only be used to verify the ballot recording process. Besides, the voter's privacy and receipt-free properties can be violated if the voter shows the receipt and all the authorities are colluding together.

In this paper, we will propose a simple and efficient method to improve individual verifiability to the scheme [20], meanwhile maintaining all other security properties. Our work is similar to [20], by using trap-door bit-commitment. But our work is superior to [20] in two aspects. First, our scheme does not suffer the flaw of [20], which has been fixed by [21]. In the scheme [20], the coercer can force the voter to use some special parameters that the ballot can only be opened in one way. As a result, the receipt-freeness is violated. Second, our scheme solves the two conflicts introduced above by applying the secret ballot technique in Prêt á Voter [3] scheme.

Our scheme allows each voter to verify not only the ballot recording process but also the ballot counting process without violating the receipt-freeness property. Moreover, our scheme can provide some mechanisms to let honest voters accuse dishonest authorities during the whole election process but dishonest voters cannot successfully accuse honest authorities. The receipt-freeness property of our scheme is achieved by the assumption of two-way untappable channel between voters and the administrator and one-way untappable channel from voters to the counter.

2 Anonymous Channel by Mix Networks

In our proposal scheme, the anonymous channel is implemented by mix networks. The concept of mix networks was first introduced by Chaum in [2] and was followed by a number of schemes, such as [10, 9, 1, 17, 11, 6, 18, 7]. Mix-net is a cryptography construction that enables one or more mix servers to take a sequence of encrypted input messages, re-encrypt or decrypt them, and output them in an unrevealed, randomly permuted order [17]. In theory, if there is one honest mix server, the permutation links are kept secure. Generally speaking, mix networks can be divided into re-encryption mix-net and decryption mix-net. Both can be used to provide an anonymous channel.

2.1 Anonymous Channel by Re-encryption Mix-Nets

Our proposed re-encryption mix network is illustrated as Figure 1. The original ballots are all encrypted under the same ElGamal public key y. Each mix server is required to perform two re-encryption. For each re-encryption, mix servers first re-encrypt a list of ElGamal encrypted ciphertexts $\{E_1, E_2, \ldots, E_n\}$ and randomly permute them, then output them as another list of ElGamal encrypted ciphertexts $\{E'_1, E'_2, \ldots, E'_n\}$. Therefore, after re-encryption mix networks, the final threshold authorities in possession of the ElGamal secret key x can decrypt all the ballots.

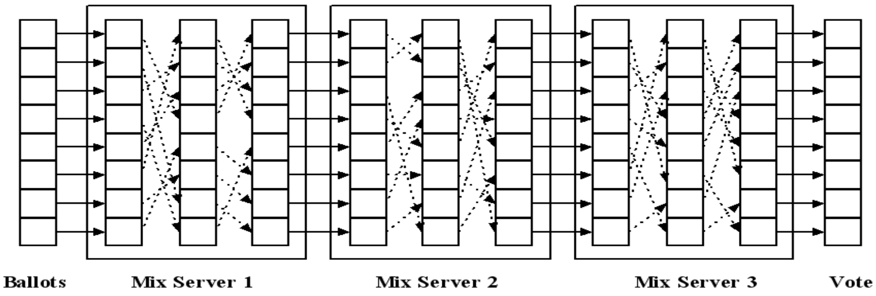

Figure 1: Mix Networks

When auditing whether mix servers have performed correctly in the re-encryption process, we can audit each mix server one by one by using *Randomized Partial Checking* [11]. Recently, Golle et al. [7] has introduced a new method which can dramatically increase the efficiency in the auditing process. Instead of encrypting message m directly as $E(m, t) = (\alpha, \beta) = (my^t, g^t)$, each voter has to encrypt the ballot in three steps:

- Compute hash checksum: compute $h = H(\alpha, \beta)$
- Double encryption: choose $r_1, r_2 \in_R Z_q$; $E(\alpha, r_1)$; $E(\beta, r_2)$
- Encrypt h: choose $r_3 \in_R Z_q$; $E(h, r_3)$

By doing this, the complexity of ballot generation and re-encryption has increased. But the whole re-encryption mix-nets can be verified more efficiently. If more than one mix servers cheat in the mix-nets, the ballot value can not hold to the checksum hash value after decryption. Then the threshold authorities in charge of decryption can detect cheating. If this happens, *Schnorr Identification Algorithm* [22] or *Fiat-Shamir heuristic* [4] can be implemented with *Randomized Partial Checking* [11] to audit each mix server one by one.

2.2 Anonymous Channel by Decryption Mix-Nets

Generating anonymous channel by decryption mix-nets is similar. The whole decryption mix-nets can also be illustrated by Figure 1. However, the ballots are not encrypted under the public key of ElGamal encryption. Instead, the ballots are encrypted under each mix server's public key, from the last mix server to the first mix server. The original ballots can be illustrated as follows:

$$\{r_{2k-1}, \{r_{2k-2}, \ldots, \{r_1, \{r_0, m\}_{PK_0}\}_{PK_1} \cdots\}_{PK_{2k-2}}\}_{PK_{2k-1}}$$

When decrypting, each mix server will first decrypt the input ballots by using her own secret key. Then, throws away the random value. And as follows, permutates the outputs and put them onto the bulletin board.

Randomized Partial Checking [11] can also be used to audit the decryption mix networks. But the auditor can only audit each mix server one by one. When auditing, the mix server should provide the random value, which she throwed away, to the auditor. And by using these random values, the auditor can verify whether this mix server has performed correctly in the decryption mix-nets.

2.3 Discussion of Re-encryption Mix-nets and Decryption Mix-nets

Anonymous channel can be generated by both re-encryption mix-nets and decryption mix-nets. Re-encryption mix networks have several advantages to decryption mix networks:

- In re-encryption mix-nets, the re-encryption and decryption process are separated. Therefore, the absence of one or more mix servers will not affect the correctness of mix networks. In contrast, the absence of one mix server will make the decryption mix networks unusable, and hence vulnerable to denial-of-service.
- In re-encryption mix networks, the auditing process is done by using zero-knowledge proof such as *Schnorr Identification Algorithm* and *Fiat-Shamir heuristic.* The mix servers can be verified without revealing the re-encrypt value r or the ballot information m. However, in decryption mix-nets, the auditing process will more or less leak out some information.
- In re-encryption mix networks, the power of the decryption key can be distributed by using threshold techniques. But in decryption mix-nets, the last mix server has too much power.
- Some latest schemes [6, 18, 7] have improved the efficiency of the auditing process for re-encryption mix networks. But in decryption mix-net, to the best of our knowledge, the auditor can only audit each mix server one by one.

However, re-encryption mix-nets have a big drawback. That is, when re-encrypting, the mix servers can not use the same re-encrypt value to encrypt all the ballots, otherwise, the attacker can easily find out all the permutation links between the inputs and the outputs. But if mix servers use different re-encrypt values for each ballot, they have to remember all these re-encrypt values because they have to use these value to prove they has performed correctly in the auditing process. In large scale election, requiring all mix servers to remember such a great amount of random values and their links to each ballot respectively is very impractical. In contrast, the decryption mix-nets do not suffer such a drawback. The mix servers in decryption mix-nets do not need to remember anything. All they need to do is to decrypt the input ballots again and they can recover all the permutation links and every random value for each ballots.

3 Brief Introduction of [Oka96]

The parties involved in the scheme [20] are the voters, the administrator, the counter and the bulletin board. The whole election process can be classified as the following five stages:

- **Preparation Stage:** a certain voter creates a ballot $m_i = BC(v_i, r_i) = g^{v_i} G_i^{r_i} \ mod\ p$, where $G_i = g^{\alpha_i} \ mod\ p$, hiding the choice v_i by using trap-door bi-committment and blind signature $x_i = H(m_i || G_i) t_i^e \ mod\ n$, and then sends x_i to the administrator.

- **Administration Stage:** the administrator checks each voter's eligibility, if a certain voter is eligible and has not voted before, the administrator will sign her ballot $y_i = x_i^{1/e}\ mod\ n$ and return y_i to the voter.
- **Voting Stage:** when the voter receives the ballot signed by administrator, she will unblind it $s_i = y_i/t_i = H(m_i||G_i)^{1/e}\ mod\ n$, and send $(m_i||G_i, s_i)$ to the counter anonymously.
- **Ballot Recording Stage:** the counter publishes all eligible ballots $(m_1, m_2, \ldots, m_k)$ onto a bulletin board which can be read by every one. Each voter checks whether her ballot m_i is correctly recorded. If yes, she send the de-committment key (v_i, r_i, m_i) to the counter anonymously. Otherwise, she can accuse the counter to a trusted third party.
- **Ballot Counting Stage:** counter decrypts all the ballots $(m_1, m_2, \ldots, m_k)$ and publishes the result $(v'_1, v'_2, \ldots, v'_k)$ on the bulletin board. Also, the counter has to prove that she knows (π, r_i) such that $m_i = BC(v_i, r_i)$ and $v'_i = v_{\pi(i)}$ without revealing (π, r_i).

The scheme [20] has achieved a lot of security properties such as completeness, uniqueness, privacy, fairness, receipt-freeness and public verifiability. However, this scheme does not achieve our definition of individual verifiability. It does not allow voters verify the ballot counting process.

4 Our Proposed Scheme

In this section, we will introduce our proposed scheme. The receipt-freeness of our scheme is achieved by the assumption of two-way untappable channel between voters and the administrator and one-way untappable channel from voters to the counter. We use the secret ballot technique from Prêt á Voter [3] to let only the voter verify whether her ballot is correctly counted in the final tally.

Before election, the system parameters are generated and published by election authorities. Any third party can verify whether the parameteres are satisfied. Suppose p and q are large numbers where $q|p-1$, and g is the order of Z_p^*.

4.1 Preparation Stage

In this stage, suppose there are k voters (V_1, V_2, $\ldots$, V_k) and 4 candidates with the names Alice, Bob, Caroline and David respectively. As a result, by using cyclic shift, the candidate list has four possibilities as shown in Figure 2.

Alice	**Bob**	**Caroline**	**David**
Bob	**Caroline**	**David**	**Alice**
Caroline	**David**	**Alice**	**Bob**
David	**Alice**	**Bob**	**Caroline**

Figure 2: Four possibilities of candidate list

A certain voter V_i creates a ballot by choosing v_i, u_i and r_i, which are defined as follows: v_i is in $\{1, 2, \ldots, m\}$, u_i is in $\{0, 1, \ldots, m-1\}$ and r_i is a random number. m is the number of candidates. In our case, there are 4 candidates, thus $m = 4$. The v_i contains the voter's choice. If $v_i = 1$, it means the voter want to vote for the first candidate Alice. If $v_i = 2$, the candidate the voter want to vote is the second one Bob, and so on. The u_i means on the displayed candidate list, the voter wants to read the result u_i cyclic shift from the top. In other words, if $u_i = 0$, then the voter wants to read her result on the top row of the candidate list. If $u_i = 1$, the voter wants to read her result on the second row of the candidate list, and so on. The function of r_i is to make (v_i, u_i, r_i) to be a trap-door bit-committment. For example, if a certain voter V_i wants to vote for the fourth candidate David and she randomly decides to read the result on the second row of the candidate list, she can choose v_i and u_i, such that $v_i = 4$ and $u_i = 1$.

Each voter first creates a ballot $BC(v_i, u_i, r_i)$ in the following form:

$$m_i = g^{v_i+u_i+r_i}\ mod\ p \qquad (1)$$

Therefore, even coercers know the information m_i, they can not know in which way the voter will open the ballot because the voter can open m_i in a lot of ways such as (v_i, u_i, r_i) and (v_i', u_i', r_i') and so on. The voter just need to make sure that $v_i + u_i + r_1 \equiv (v_i' + u_i' + r_i')\ mod\ q$. Define $R_i = g^{r_i}\ mod\ p$. As follows, similar to [20], the voter does the blind signature:

$$x_{1i} = H(m_i||R_i)t_i^e\ mod\ n \qquad (2)$$

$$x_{2i} = H(v_i||u_i||r_i)t_i^e\ mod\ n \qquad (3)$$

Here, t_i is a random number in Z_n and (e, n) is the public key of a RSA encryption. The corresponding decryption key $1/e$ is hold by administrator A. Later, V_i signs x_{1i} by using her own secret key $z_i = sign(x_{1i})$. After that, V_i sends $(x_{1i}||x_{2i}||z_i||ID_i)$ to the administrator A.

4.2 Administration Stage

A checks the eligibility of V_i. If V_i is not eligible or she has voted before, administrator A will reject administrating this ballot, Otherwise, administrator A will sign the ballot by using her secret key and send it back to the voter.

$$y_{1i} = x_{1i}^{1/e} \qquad (4)$$

$$y_{2i} = x_{2i}^{1/e} \qquad (5)$$

4.3 Voting Stage

When the voter V_i receive the signed ballot (y_{1i}, y_{2i}) from the administrator A, she can unblind the ballot by $s_{1i} = y_{1i}/t_i = H(m_i||R_i)^{1/e}$ and $s_{2i} = y_{2i}/t_i = H(v_i||u_i||r_i)^{1/e}$. Then V_i sends $(m_i||R_i, s_{1i})$ to the counter anonymously through a proposed mix network as Figure 3.

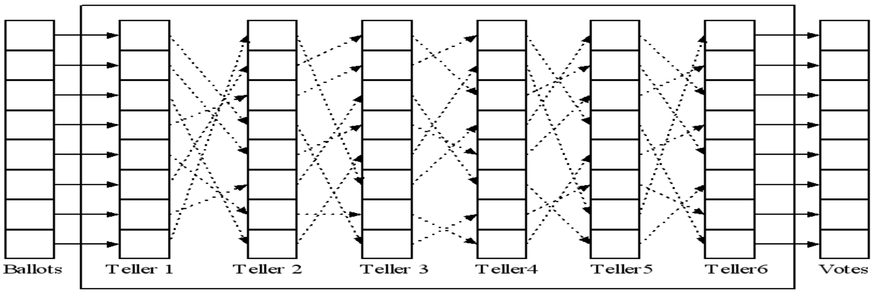

Figure 3: Proposed Mix Networks

4.4 Ballot Recording Stage

The counter publish all the eligible ballot $(m_1, m_2, \ldots, m_k)$ in random order on to the bulletin board. Each voter checks whether her ballot m_i is correctly displayed on the bulletin board. If not, V_i can use $s_{1i} = H(m_i||R_i)^{1/e}$ as a receipt to accuse the counter to a trusted third party.

4.5 Ballot Counting Stage

Define T is the deadline that every voter should verify the ballot recording stage. If V_i finds out her ballot m_i is correctly recorded on the bulletin board and the serial number of her ballot on the bulletin board is l, then after the deadline T, V_i should send (l, v_i, u_i, s_{2i}) to the counter through an untappable anonymous channel (also by mix networks). By decrypting, the counter should display the result in secret ballot form onto the bulletin board. The counter is not allowed to publish the

s_{2i} on the bulletin board but she has to record it in some place else. Otherwise, the receipt-freeness property will be violated. At this moment, each voter can check whether the counter has displayed her ballot result correctly. Only the voter can verify her own ballot but others can not verify this ballot because they do not know the de-committment key (v_i, u_i, r_i). Moreover, even if the voter proves the coercer how she voted, the coercer will not trust her because she can cheat them by using a different de-committment key (v_i', u_i', r_i'). Therefore, the receipt-freeness is maintained.

For example, in our case, if V_i wants to vote for the fourth candidate David and she randomly decides to read the result on the second row of the candidate list, she chooses $v_i = 4$ and $u_i = 1$ and then chooses r_i randomly in the preparation stage. When the counter display the final result, suppose the counter display the result as show in Figure 4, V_i can check whether the result in the second row of the candidate list is her choice, David. But the result makes no sence to others.

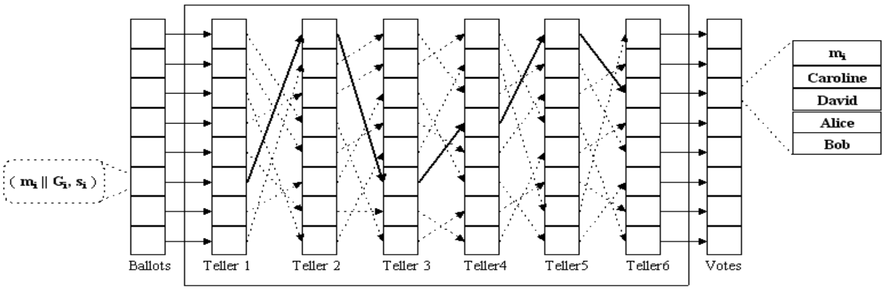

Figure 4: Mix Networks and Result

4.6 No-cheating in our scheme

It is clear that in the whole election process, the authorities may cheat in administration stage, ballot recording stage and ballot counting stage.

If the administrator cheats in administration stage, the ballot sent back from the administrator will not contain the administrator's signature. Then V_i can use ID_i to accuse the administrator to a trusted third party.

If the counter cheats in the ballot recording stage, recording the ballot incorrectly, then V_i can use $s_{1i} = H(m_i||R_i)^{1/e}$ as a receipt to accuse the counter to a trusted third party.

If V_i finds our that the counter cheating in the ballot counting stage, she has two choice. One is the election system can let V_i vote again without reason, then the previous vote will be deleted and the system only deals with the new vote. By doing this, the voter's privacy is maintained but the voter can not accuse the counter. The other solution, if the voter wants to accuse the counter, she has to violate her privacy. She can accuse the counter to a trusted third party with the help of the administrator and there are three situations:

- First, it is clear that without colluding with the adminstrator, the counter can not fabricate s_{2i}' which contains the administrator's signature. In other words, if the counter fabricate s_{2i}' without the administrator's signature, the counter is cheating.
- Second, if the counter uses another received information s_{2j} where $i \neq j$, and says it is s_{2i}, the administrator can prove that the counter is cheating because s_{2j} does not belong to V_i.
- Third, if the counter shows the real s_{2i}, V_i can open (v_i, u_i, r_i) to show that the counter is cheating. And because if the hash function is secure, V_i can not fabricate (v_i', u_i', r_i') to match with the hash value s_{2i}, so a dishonest voter cannot successfully accuse honest counter.

Therefore, honest voters can accuse a cheating counter, but dishonest voters can not successfully accuse honest counters. The drawback is that if a honest voter accuses the cheating counter, her privacy will be violated. The threat of our scheme is the same as other blind signature based schemes. If the administrator collude with the counter, a lot of security properties will be violated.

5 Conclusion

The main object of this paper is to introduce how to provide individual verifiability property to the scheme [20], while maintain receipt-freeness. Most receipt-free election schemes based on blind signature [20, 21, 19] are based on the assumption of one-way anonymous channel from the voter to the counter, but because we wish to give each voter a mechanism to accuse the counter if the counter is cheating in the ballot counting process, we also need the assumption of a two-way untappable channel between the voter and the administrator to achieve receipt-freeness.

Our proposed scheme maintains all the properties of the scheme [20] except public verifiability. But the method introduced in the scheme [20] can easily improve public verifiability to our scheme. That is, the counter publishes two lists of result, one for individual verifiability and another one for public verifiability. Also, she proves that she knows the links between the two lists without revealing the links. For reason of space, we will not describe the detailed technique issues.

Note that the scheme Prêt á Voter [3] does not achieve our definition of individual verifiability. But we can use the same method introduced in this paper to provide individual verifiability to [3] as well. The election authorities also need to display the result in secret ballot form. When voting, the election system shows a cyclic clock from 1 to m which is the number of candidates. If a certain voter wishes to read her result on the k^{th} row in the candidate list, she input her choice by clicking the DRE screen when the cyclic clock shows k. Therefore, the individual verifiability property is achieved while maintaining receipt-freeness, only the voter can verify her own result.

References

[1] M. Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. *Advances of Eurocrypt'98*, LNCS 1403:437–447, 1998.

[2] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[3] D. Chaum, P. Ryan, and S. Schneider. A practical voter-verifiable election scheme. *Proceedings of the tenth European Symposium on Research in Computer Science (ESORICS'05)*, pages 118–139, 2005.

[4] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. *Advances of Cryptology-Crypto'86*, pages 186–199, 1986.

[5] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. *Advances of Auscrypt'92*, LNCS 718:244–251, 1992.

[6] J. Furukawa and K. Sako. An efficient scheme for proving a shuffle. *An efficient scheme for proving a shuffle*, LNCS 2139:368–387, 2001.

[7] P. Golle, S. Zhong, D. Boneh, M. Jakobsson, and A. Juels. Optimistic mixing for exit-polls. *Advances of Asiacrypt'02*, LNCS 2501:451–465, 2002.

[8] M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. *Advances of Eurocrypt'00*, LNCS 1807:539–556, 2000.

[9] M. Jakobsson. Flash mixing. *Proceedings of the eighteenth annual ACM symposium on Principles of Distributed Computing (PODC'98)*, pages 83–89, 1998.

[10] M. Jakobsson. A practical mix. *Advances of Eurocrypt'98*, LNCS 1403:449–461, 1998.

[11] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. *Proceedings of the eleventh USENIX Security Symposium*, pages 339–353, 2002.

[12] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. *Advances of Eurocrypt'96*, LNCS 1070:143–154, 1996.

[13] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, pages 61–70, 2005.

[14] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. *Proceedings of ICISC'03*, LNCS 2971:245–258, 2003.

[15] B. Lee and K. Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. *ICISC 2002*, LNCS 2587:389–406, 2002.

[16] E. Magkos, M. Burmester, and V. Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channel. *The first IFIP Conference on E-commerce/E-business/E-government*, pages 683–693, 2001.

[17] J. Markus and J. Ari. Millimix: Mixing in small batches. *DIMACS technical report 99-33*, 1999.

[18] C. A. Neff. A verifiable secret shuffle and its application to e-voting. *Proceedings of the eighth ACM conference on Computer and Communications Security (CSS'01)*, pages 116–125, 2001.

[19] M. Ohkubo, F. Miura, M. Abe, A. Fujioka, and T. Okamoto. An improvement on a practical secret voting scheme. *Information Security'99*, LNCS 1729:225–234, 1999.

[20] T. Okamoto. An electronic voting scheme. *Proceedings of IFIP'96*, pages 21–30, 1996.

[21] T. Okamoto. Receipt-free electronic voting schemes for large scale elections. *Proceedings of the fifth International Workshop on Security Protocols*, LNCS 1361:25–35, 1998.

[22] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, pages 161–174, 1991.