Efficient Voter Verifiable E-Voting Schemes with Cryptographic Receipts

Yunho Lee, Kwangwoo Lee, Seungjoo Kim, and Dongho Won

Information Security Group,
Sungkyunkwan University,
300 Chunchun-dong, Suwon, Gyeonggi-do, 440-746, Korea
{leeyh,kwlee,skim,dhwon}@security.re.kr
http://www.security.re.kr

Abstract. Current electronic voting systems are not sufficient to satisfy trustworthy elections as they do not provide any proofs or confirming evidences of their honesty while they provide more efficiency than plain paper voting system. Receipts could assure voters that their intended votes are counted. In this paper, we propose two efficient schemes for issuing receipts in electronic voting. Our schemes do not require any special printers or scanners, nor frequent observations to voting machines. In addition to that, our schemes are more secure than the previous ones.

Key words: E-Voting, Voter Verifiable E-Voting, Receipts

1 Introduction

Current electronic voting systems require voters to trust them. Voters should believe that the voting machines do not cheat even though they do not provide any proofs or confirming evidences of their honesty. This so called *Black-Box Voting* is the greatest obstacle to conduct electronic voting.

Mercuri [6] stated that fully electronic systems do not provide any way that the voter can truly verify that the ballot cast corresponds to that being recorded, transmitted, or tabulated as many other experts did. Moreover, any programmer can write code that displays one thing on a screen, records something else, and prints yet another result. There is no known way to ensure that this is not happening inside of a voting system.

The most effective way to decrease the trust voters must place in voting machine software is to let voters physically verify that their intent is recorded correctly, for e.g., voter verifiable paper trails.

In 2002, Mercuri proposed a method for voter verifiable ballots [8]. In her method, after a voter has finished making selections using a voting machine, the machine prints out a paper ballot that contains the voter's selections for each choice. The printed ballot is kept behind a window to prevent voters from having any opportunity to tamper with it. If voters examine and approve the ballot, the voting machine drops the printed ballot into an opaque ballot box. While voter verifiable paper ballots eliminate the need to trust the voting machine, the need to support printing and collecting of paper ballots increases the maintenance costs and election complexity for the poll workers [14].

In 2004, Chaum proposed a method to provide voters with a coded receipt that reflects their vote but does not reveal it to anyone else [13]. In his scheme, the voting machine prints the coded receipt on the two separable layers using visual cryptography [4]. When laminated together, they reveal the voter's choices, but each separated layer is meaningless dots. The voter verifies the laminated receipt and then selects one of the two layers to retain as receipt. The other layer should be surrendered to a poll worker and shredded. The voting machine could cheat if it knows which layer will be selected by the voter in advance, thus the chance is $\frac{1}{2}$. The costs to implement Chaum's scheme is relatively high as it is required that all voting machines to be equipped with special printers.

Neff *et al.* proposed another method to provide voters with a coded receipt [9]. In their scheme the receipt is printed with a code for each selection which is based on a codebook generated uniquely for the voter's ballot sequence number. The code numbers are meaningless to anyone other than the voter, who sees the code numbers displayed by the voting machine. The validity of the displayed codes can be assured by using observers at randomly selected times throughout the election who act as voters and audit the displayed codes. Thus, the chance that the voting machine could cheat is $\frac{c}{l+c}$ where c and l are the number of observations and the number of voters respectively.

In 2005, Klonowski *et al.* proposed a voting scheme with receipts [15]. They incorporated the ideas of Chaum's voting scheme, printing method of van de Graaf, mixing via re-encryption and a cut-and-choose mechanism that is used to catch cheating parties in a mix-network. Their scheme's cost is low as it does not require any special printers. However, the chance of a voting machine's fraud is $\frac{1}{2}$ which is the same as Chaum's scheme.

An e-voting scheme should be verifiable by voters and observers whether casting and counting ballots are performed correctly or not. A verifiable e-voting involves the following two distinct checks: [9]

- (**Requirement 1**) A voter should be able to satisfy him or herself that the voted ballot is captured correctly (*cast-as-intended*); and
- (**Requirement 2**) Any observer should be able to satisfy him or herself that the voted ballot is counted correctly (*counted-as-cast*).

The second requirement preserving anonymity can be satisfied by using various methods for e.g., a provably secure mix-network. However, in case of the first requirement, it is not easy to satisfy it without voter verifiable receipts because no one can trust voting machines. While a voter verifiable receipts are used for satisfying the first requirement, nobody can prove his or her vote to anyone even to him or herself in order to prevent *vote-buying* and *selling.*

In this paper, we propose two efficient methods for issuing receipt to a voter for efficient voter verifiable e-voting. Our schemes do not require voting machines to be equipped with special printers, observations to voting machines, and secret codebooks as Chaum's scheme or Neff's scheme. Moreover, in most cases, our schemes are more secure than the previous ones. The rest of this paper organized as follows. Section 2 outlines cryptographic primitives for our schemes. Section 3 describes the Neff's e-voting scheme. Section 4 presents our two efficient receipt issuing schemes and we analyze security of the schemes in section 5. Finally, this paper is concluded in Section 6.

2 Cryptographic Primitives

2.1 Probabilistic Encryption

Probabilistic encryption is the use of randomness in an encryption algorithm, so that when encrypting the same message several times it will yield different ciphertexts. Probabilistic encryption is particulary important when the plaintext space is extremely finite for e.g., e-voting. If a deterministic encryption algorithm is used for e-voting, the adversary can simply try encrypting each of his guesses under the recipient's public key, and compare each result to the target ciphertext. To combat this attack, public key encryption schemes must incorporate an element of randomness, ensuring that each plaintext maps into one of a large number of possible ciphertexts. Among several probabilistic encryption schemes such as [2][3][5], we will use ElGamal cryptosystem for our e-voting schemes. We denote ElGamal encryption of message M with arbitrary random number r as $E(M, r)$. Please refer to [3].

2.2 Mix-Net

A mix network or mix-net accepts as input a collection of ciphertexts, and outputs associated plaintexts in a randomly permuted order. A well constructed mix-net makes it infeasible for an adversary to determine which output plaintext corresponds to which input ciphertext more efficiently than by guessing at random. Proposed by Chaum [1] in 1981 as a technique for anonymous e-mail and e-voting, mix-net can be categorized into *decryption mix-nets* and *re-encryption mix-nets*. As ElGamal encryption is a good example for re-encryption mix-nets, we assume that a re-encryption mix-net [7][12] is used for our schemes.

3 Previous Work

3.1 Neff's Scheme

Neff proposed a practical e-voting scheme for issuing voter verifiable receipt [9]. In this section, we briefly introduce Neff's e-voting scheme. For simplicity, we assume that the e-voting is limited to a single precinct, a single race and a single trustee.

Initialization The election trustee selects a large prime p, a generator g and a secret key x, and then publishes p, g, and a public key $h = g^x (\bmod\ p)$.

Let l and n denote a few times larger number than the maximum number of voters and the number of candidates respectively. For each candidate $A_j (1 \leq j \leq n)$, the trustee generates a corresponding mark $\alpha_j \in \langle g \rangle$ by way of a publicly defined pseudorandom process.

The Codebook Commitments For $1 \leq i \leq l$, the trustee creates commitment (γ_i, τ_i) by forming

$$\tau_i = \gamma_i^{\sigma_i} (\bmod\ p), \tag{1}$$

where γ_i is a randomly chosen element of $\langle g \rangle$. The trustee keeps secret the discrete logarithm, $\sigma_i = log_{\gamma_i} \tau_i (\bmod\ p)$.

Each codebook $D_i (1 \leq i \leq l)$ is determined by the following equation

$$C_i(A_j) = H\left(\alpha_j^{\sigma_i} (\bmod p)\right) \tag{2}$$

where $H(\cdot)$ is a public hash function.

Voting After voter identification process, the voting machine displays the verification codes to the voter V_i, each identified by a ballot sequence number. V_i makes her selection, is delivered her receipts, checks the ballot receipt against the verification code and casts her vote. Once voted, the voting machine captures the V_i's ElGamal encrypted choice as $B_i = E(\mu_i, r)$ where μ_i is α_j for some j corresponding to V_i and r is an arbitrary random number.

To ensure that the voting machine does not cheat, an observer can check the verification codes on a voting machine at any time during the voting day. For auditing codebooks, an observer would not vote, but simply takes the committed codebook, checks that any on-screen verification codes were the ones committed, and readies the machine for the next voter or observer.

Codebook Verification Once the polls close, the trustee separates verification codebooks into used ones by voters and unused ones. For each unused verification codebooks, the trustee reveal his secret $\sigma_i (1 \leq i \leq l)$. Anyone can easily check validity of the codebook commitments and verification codes by equations (1) and (2).

Voter Verification of the Ballot Box The entire collection of encrypted ballots $B_i (1 \leq i \leq l)$ are published. The trustee computes $E(B_i, \sigma_i)$ and decrypt it using $\sigma_i x$ as private key, yielding $\mu_i^{\sigma_i} (\bmod\ p)$. Since the computation of $E(B_i, \sigma_i)$ is done in secret, the trustee must post a proof of validity demonstrating that $E(B_i, \sigma_i)$ is computed from B_i.

And then, applying hash function H gives $H\left(\mu_i^{\sigma_i} (\bmod\ p)\right)$ which is exactly the voter V_i's verification code.

Results Verification Counting is performed by of a verifiable re-encryption mix-net. The input and output sets of encrypted ballots are accompanied by a proof of validity that proves that the output set exactly matches the input set. After shuffling by mix-net, the re-encrypted ballots are decrypted and tabulated.

4 Proposed Schemes

A receipt issued to a voter which can be took out of the polling place increases voter's confidence that the ballot was *cast-as-intended* and *counted-as-cast* as she can verify the whole processes of election at any time using her receipt. Though both Chaum's and Neff's schemes give receipts to the voters, both schemes have disadvantages. In Chaum's scheme, receipt issuing cost is high as it is required that every voting machines should be equipped with special printers. In Neff's scheme, the trustee(s) should make a codebook in advance and observer(s) are required in order to audit the voting machines' operation.

Moreover in Chaum's scheme, the chance that a voting machine's cheating would go undetected is one half for each ballot and in Neff's scheme, the chance is $\frac{c}{l+c}$ where c is the number of observations and l is the number of voters.

In this paper, we propose two different schemes for issuing receipts which do not require observations, any special printers and are secure not less than the both schemes. Our schemes are based on the Neff's scheme and the whole e-voting procedure is depicted in Fig. 1.

A voter can be assured that her vote is *cast-as-intended* by her own computation of ElGamal encryption or by using a public verification server. Also a voter can verify that her vote is *counted-as-cast* by comparing her encrypted choice with the result registered to the public bulletin board and verifying ballot shuffling by mix-net.

4.1 The Proposed Scheme 1

In Neff's scheme, to ensure that the voting machine does not cheat, it is important that the codebook be committed before you vote. Furthermore, an observer should check the codebooks on a voting machine during the voting day to make sure that the voting machine is not displaying erroneous verification codes.

We propose a more efficient scheme for issuing receipt in e-voting which does not require observers and prior commitment of codebooks. Let n and $E(\cdot,\cdot)$ denote the number of candidates and ElGamal encryption. The scheme 1 can be described as follows.

1. Voting machine displays n encrypted code pairs
$$(e_j, e'_j) = (E(j,r), E(j,r')),$$
where r and r' are random numbers for each $j = 1, ..., n$.
2. For $j = 1, ..., n$, Voter randomly selects e_j or e'_j, and voting machine prints n selected codes and their corresponding random numbers on receipt as proofs.

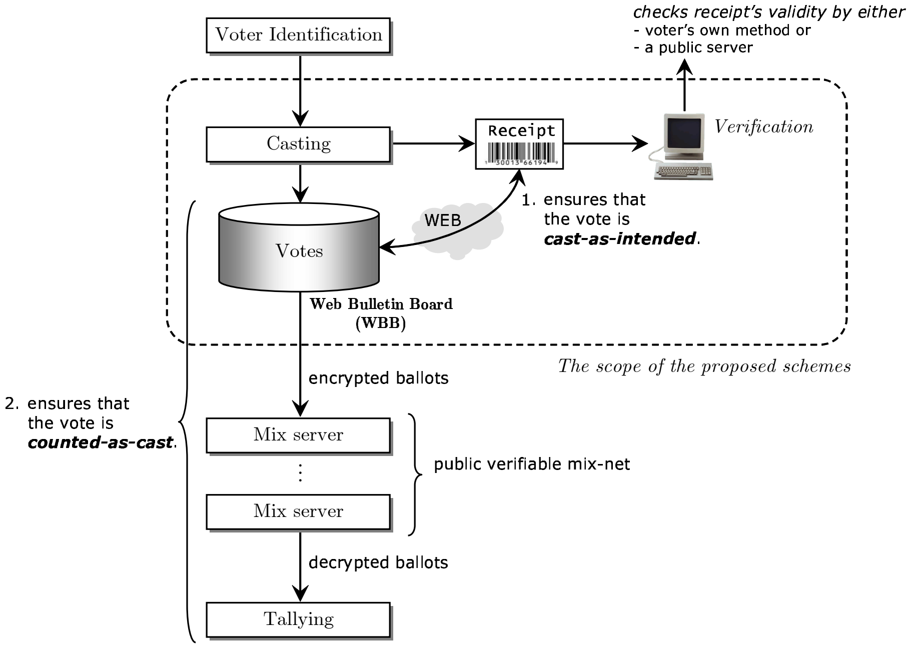
Voter Identification
checks receipt's validity by either
- voter's own method or
- a public server
Casting
Receipt
Verification
Votes
WEB
1. ensures that the vote is *cast-as-intended*.
Web Bulletin Board (WBB)
The scope of the proposed schemes
encrypted ballots
2. ensures that the vote is *counted-as-cast*.
Mix server
⋮
Mix server
public verifiable mix-net
decrypted ballots
Tallying

Fig. 1. Overall Procedure for E-Voting and The Scope of Our Proposed Schemes

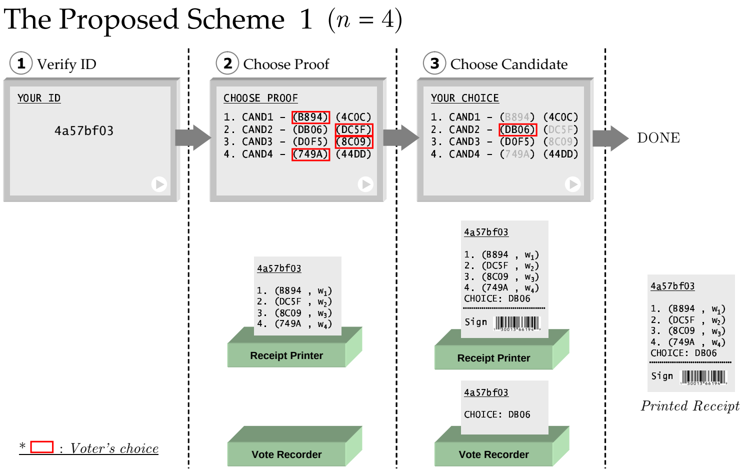

Fig. 2. Proposed Scheme 1. $(n = 4)$

3. Let e_j and w_j be the n selected codes and their corresponding random numbers. Let $e_j^*(j = 1, ..., n)$ be the n unselected ones.
4. Voter casts her ballot by choosing $v_i(1 \leq v_i \leq n)$.
5. Voting machine stores $e_{v_i}^*$ and prints it on the receipt with digital signature σ_i for the receipt. Voter should verify that the printed $e_j(j = 1, ..., n)$ and $e_{v_i}^*$ are the same as on the screen.
6. After leaving the polling place, voter checks the validity of receipt using the voting machine's signature σ_i. Then voter should check the proof whether the following equation holds by encrypting j with corresponding random number w_j.
$$\forall j = 1, ..., n, \ \ E(j, w_j) = e_j.$$
This check is for *cast-as-intended* and can be done by *public verification server*.
7. Voter also checks the public bulletin board that the unique ID s_i and the encrypted her choice $e_{v_i}^*$ is posted correctly.

Please note that candidates' representations $j(1 \leq j \leq n)$ can be replaced by unique random numbers that is generated by way of a publicly defined pseudorandom process as Neff's scheme. For this scheme, the chance that the voting machine's cheating is undetected is 2^{-n}. Fig. 2. depicts the scheme 1 in case of $n = 4$.

4.2 The Proposed Scheme 2

Obviously the scheme 1's security is dependent to the number of candidates. The scheme 2 is enhanced version of the scheme 1 introducing a security parameter t in order to make it more secure. The scheme 2 can be described as follows.

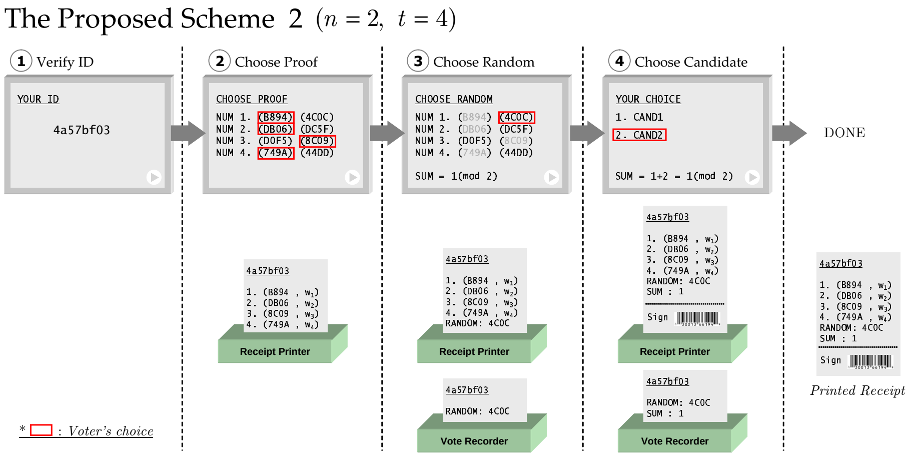

Fig. 3. Proposed Scheme 2. $(n = 2, t = 4)$

1. For a predefined security parameter t, voting machine displays t encrypted code pairs
$$(e_j, e'_j) = (E(j, r), E(j, r')),$$
where r and r' are random numbers for each $j = 1, ..., t$.
2. For $j = 1, ..., t$, voter randomly selects e_j or e'_j, and voting machine prints t selected codes and their corresponding random numbers on receipt as proofs.
3. Let e_j and w_j be the t selected codes and their corresponding random numbers. Let $e^*_j (j = 1, ..., t)$ be the t unselected codes.
4. Voter selects a random number $r_i \in \{1, ..., t\}$ and voting machine stores $e^*_{r_i}$ and prints it on the receipt.
5. Voter casts her ballot by choosing $v_i (1 \leq v_i \leq n)$.
6. Voting machine computes $m_i = r_i + v_i (\mathrm{mod}\ n)$.
7. Voting machine stores m_i and prints it on the receipt with digital signature σ_i for the receipt. Voter should verify that the printed $e_j (j = 1, ..., t)$, $e^*_{r_i}$ and m_i are the same as on the screen.
8. After leaving the polling station, voter checks the validity of receipt using σ_i. Then voter should check the proof whether the following equation holds by encrypting j with corresponding random number w_j.
$$\forall j,\ E(j, w_j) = e_j (1 \leq j \leq t).$$
This check is for *cast-as-intended* and can be done by *public verification server.*
9. Voter also checks the public bulletin board that the unique ID s_i, $e^*_{r_i}$, and m_i are posted correctly.

The security parameter t should be multiple of n. If it is not, then for m_i, the following equation does not hold.

$$\forall j,\ \Pr[v_i = j] = \frac{1}{n} (1 \leq j \leq n). \tag{3}$$

For example, if we choose $n = 2,\ t = 3$ and the computed $m_i = 0$, then

$$\Pr[v_i = 1] = \frac{2}{3},\ \Pr[v_i = 2] = \frac{1}{3}. \tag{4}$$

As in the scheme 1, please note that candidates' representations $j (1 \leq j \leq n)$ and t random numbers can be replaced by unique random numbers that is generated by way of a publicly defined pseudorandom process.

4.3 Tallying

While tallying for the scheme 1 is exactly the same as the Neff's scheme, for the scheme 2, additional modulo arithmetic is required to tabulate in addition to decryption. The v_i can be computed by the following equation using $e^*_{r_i}$ and m_i.

$$v_i = m_i - E^{-1}(e^*_{r_i})(\mathrm{mod}\ n),\ \text{where } 1 \leq v_i \leq n \tag{5}$$

For example, suppose $n = 2$, $t = 4$, $r_i = 3$, and $v_i = 2$, then the ballot record will be $(E(3), 1)$ as $m_i = 3 + 2 = 1 (\text{mod } n)$. The decryption is performed as follows.

$$\begin{aligned} v_i &= 1 - E^{-1}(E(3))(\text{mod } 2) && (6) \\ &= 1 - 3(\text{mod } 2) && (7) \\ &= 0(\text{mod } 2) = 2(\text{mod } 2) && (8) \end{aligned}$$

The v_i should be 2 because v_i ranges from 1 to n.

5 Security Analysis

The security analysis of a voter verifiable receipt can be measured by the chance of detection of a voting machine's fraud and the receipt-freeness of a receipt. Receipt-freeness means that no one can obtains or is able to construct a receipt proving the content of a ballot. More formally, if we denote the number of candidates by n and the printed value on the receipt by α, then the probability that each candidate is mapped to α should be exactly $\frac{1}{n}$.

Theorem 1. *(Fraud detection of the scheme 1) Let $n (\geq 2)$ denotes the number of candidates. For each ballot, the chance of detection of a voting machine's fraud is at least $\frac{3}{4}$.*

Proof. The only way that a voting machine could change a voter's correctly casted ballot without detection is to predict a voter's choice of n codes for proof. For each $l (1 \leq l \leq n)$, there is only two codes to choose and the probability of successful prediction is exactly 2^{-n}. Considering $2 \leq n$, the probability of detecting the voting machine's cheating δ is at least $frac34$.

$$\frac{3}{4} \leq \delta = 1 - \frac{1}{2^n}, \text{ where } 2 \leq n \quad \Box \tag{9}$$

Similarly, in case of the scheme 2, the δ is determined by

$$\frac{15}{16} \leq \delta = 1 - \frac{1}{2^t}, \text{ where } 4 \leq t \tag{10}$$

where $t (\geq n)$ is predefined security parameter.

Obviously, receipt-freeness of the scheme 1 is satisfied because there are only encrypted values of a voter's choice and proofs and there is no relations between a voter's choice and one of proofs.

Theorem 2. *(Receipt-freeness of the scheme 2) In the scheme 2, let n be the number of candidates. For a code m_i printed on a voter i's receipt,*

$$\forall j, \ \Pr[v_i = j] = \frac{1}{n}(1 \leq j \leq n),$$

iff $n \mid t$ with $n \leq t$.

Proof. (Sketch) For two finite sets μ_1, μ_2 and a constant c, we define a operation $\odot$ as (11) and (12).

$$\mu_1 \odot \mu_2 = \{\epsilon_1 + \epsilon_2 \mid \forall\, \epsilon_1, \epsilon_2, \text{ where } \epsilon_1 \in \mu_1 \text{ and } \epsilon_2 \in \mu_2\}. \tag{11}$$

$$\mu_1 \odot c = \{\epsilon_1 + c \mid \forall\, \epsilon_1, \text{ where } \epsilon_1 \in \mu_1\}. \tag{12}$$

Let ϕ_j^s be the number of occurrence of an element j in a set s, for e.g., for $s = \{0, 1, ..., n-1\}$, the following equations hold.

$$\forall j,\ \phi_j^s = 1(0 \le j \le n-1) \ \text{ and} \tag{13}$$

$$\sum_{j=0}^{n-1} \phi_j^s = n. \tag{14}$$

Now, we can consider $\odot$ as an identical operation for a set s and a constant c in modulo arithmetic as

$$s \odot c \pmod n = \tag{15}$$

$$\{0, ..., n-1\} \odot c \pmod n = \tag{16}$$

$$\{c, ..., n+c-1\} \pmod n = \tag{17}$$

$$\{0, ..., n-1\} \pmod n = s. \tag{18}$$

Thus, $\phi_j^{s \odot c (\mathrm{mod}\ n)} = \phi_j^s = 1\ (0 \le j \le n-1)$. Suppose that $u = \{0, ..., t-1\}$ with $n \le t$, then $\phi_j^{s \odot u (\mathrm{mod}\ n)} = t\ (0 \le j \le n-1)$.

For a given m_i $(0 \le m_i \le n-1)$ and if we choose r_i from u uniformly and $n \nmid t$, then v_i from s satisfying $m_i = v_i + r_i \pmod n$ is not uniform because $|u| = t$ and $\phi_{m_i}^{s \odot u (\mathrm{mod}\ n)} = t\ (0 \le m_i \le n-1)$. □

Table 1 illustrates a comparison of fraud detection probabilities of the proposed schemes, Chaum's scheme and Neff's scheme. Efficiency comparison of these four schemes is presented in Table 2.

Table 1. The Detection Probabilities of Voting Machine's Cheating

Scheme 1	Scheme 2	Chaum's Scheme	Neff's Scheme	Klonowski's Scheme
$1 - 2^{-n}$	$1 - 2^{-t}$	2^{-1}	$1 - \frac{c}{l+c}$	2^{-1}

n : The number of candidates(≥ 2).
t : The security parameter with $n \le t$ and $n \mid t$.
l : The number of voters.
c : The number of observations.

Let δ be the detection probability of a voting machine's fraud for a voter. For Neff's scheme, in order to make $\delta \ge \frac{1}{2}$, the number of observations c should be greater than or equal to the number of voters l which is very inefficient. In case of Chaum's scheme, $\delta = \frac{1}{2}$ regardless of the number of voters l.

Suppose that $n = 2$ which is the minimum number of candidates and $t = 4$, then δ for the scheme 1 is $1 - 2^{-n} = \frac{1}{4}$ and δ for the scheme 2 is $1 - 2^{-t} = \frac{15}{16}$.

Table 2. Efficiency Comparison

	Proposed Scheme 1,2 and Klonowski's Scheme	Chaum's Scheme	Neff's Scheme
Secret codebooks	×	×	⊙
Observers	×	×	⊙
Special printers	×	⊙	×

⊙ : Required.
× : Not required.

6 Conclusions

As many experts said, voter verifiable audit trail is the most effective way to make trustworthy electronic elections. In this paper, we have presented two efficient schemes for issuing cryptographic receipts in electronic voting. Our schemes do not require any special printers or scanners nor frequent observations to voting machines. In addition to that, our schemes are more secure than the previous ones, Chaum's scheme, Klonowski's scheme, and Neff's scheme.

References

1. D.Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Comm. of the ACM*, vol. 24, no. 2, pages 84–88, Feb. 1981.
2. S.Goldwasser and S.Micali, "Probabilistic Encryption," *Journal of Computer System Sciences(JCSS)*, vol. 28, no. 2, pages 270–299, Apr. 1984.
3. T.ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. on Information Theory*, vol. IT-31, no. 4, pages 469–472, 1985.
4. M.Naor and A.Shamir, "Visual Cryptography," *Proc. of Advances in Cryptology(Eurocrypt'94)*, LNCS 950, pages 1–12, 1995.
5. P.Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *Proc. of Advances in Cryptology(Eurocrypt'99)*, LNCS 1592, pages 223–238, 1999.
6. R.Mercuri, "Rebecca Mercuri's Statement on Electronic Voting," `http://www.notablesoftware.com/RMstatement.html`, 2001.

7. C.A.Neff, "A Verifiable Secret Shuffle and Its Application to E-Voting," *Proc. of the 8th ACM Conference on Computers and Communications Security(CCS-8)*, pages 116–125, 2001.

8. R.Mercuri, "A Better Ballot Box?," *IEEE Spectrum Online*, pages 46–50, Oct. 2002.

9. C.A.Neff and J.Adler, "Verifiable e-Voting: Indisputable Electronic Elections at Polling Places," `http://www.votehere.net/vhti/documentation/VH_VHTi_WhitePaper.pdf`, VoteHere Inc., 2003.

10. C.A.Neff, "Practical High Certainty Intent Verification for Encrypted Votes," `http://www.votehere.net/documentation/vhti`, VoteHere Inc., 2004.

11. D.Chaum, P.Y.A.Ryan, and S.Schneider, "A Practical, Voter-Verifiable Election Scheme," `Technical Report CS-TR-880`, University of Newcastle upon Tyne, 2004.

12. P.Golle, M.Jakobsson, A.Juels, and P.Syverson, "Universal Re-encryption for Mixnets," *CT-RSA 2004*, LNCS 2964, pages 163–178, 2004.

13. D.Chaum, "Secret-Ballot Receipts: True Voter-Verifiable Elections," *IEEE Security and Privacy Magazine*, vol. 2, no. 1, pages 38–47, Jan. 2004.

14. D.Evans and N.paul, "Election Security: Perception and Reality," *IEEE Security and Privacy Magazine*, vol. 2, no. 1, pages 24–31, Jan. 2004.

15. M.Klonowski, M.Kutyłowski, A.Lauks and F.Zagórski, "A Practical Voting Scheme with Receipts," *Proc. of 8th International Conference, ISC 2005*, LNCS 3650, pages 490–497, 2005.