A Robust Electronic Voting Scheme
Against Side Channel Attack

YI-NING LIU1, WEI GUO1, CHI CHENG2, CHINGFANG HSU3, JUN-YAN QIAN1 AND CHANG-LU LIN4

1Guangxi Key Lab of Trusted Software
Guilin University of Electronic Technology
Guilin, 541004 P.R. China
2School of Computer Science
China University of Geosciences
Wuhan, 430074 P.R. China
3Computer School
Central China Normal University
Wuhan, 430079 P.R. China
4College of Mathematics and Computer Science
Fujian Normal University
Fuzhou, 350117 P.R. China
E-mail: yniliu@guet.edu.cn

A challenging task in the design of secure e-voting system is that a voter could verify whether his vote is recorded and counted correctly, while he has no evidence to prove his ballot in order to prevent a malicious voter from selling his vote. Bingo Voting scheme has been proposed to achieve the verifiability and coercion-resistance; however, it has a weakness, which could still lead to vote selling due to a potential of side channel attack. A malicious voter could take a picture of the fresh random number displayed in the voting booth using a secret camera, and then prove to others which candidate he has selected. In this paper, we propose an improved voting scheme, which does not require a trusted random number generator and eliminates the side channel attack, while our scheme still retains the properties of verifiability and coercion-resistance. It also allows the voter to check whether the receipt is correct even after he has left the voting booth.

Keywords: e-voting, coercion-resistant, side channel attack, off-site inspection, verifiable random number

1. INTRODUCTION

Voting has played a vital role in the modern society, since it guarantees that the successor of a government is selected in a democratic way. The traditional voting scheme employing the paper and ballot boxes cannot guarantee the security, and the destructions of votes are reported all over the world. In recent years, the electronic voting (e-voting) scheme has attracted much attention due to the feature of its cryptographic security. Generally, a secure e-voting scheme should be correct and universally verifiable. Correctness means that the voter’s intention should be recorded and tallied correctly, while universal verifiability ensures that any observer can verify the operation of voting process completely. In addition, a voting scheme in practice should be coercion-resistant, which means nobody can learn the content of a voter’s ballot [1].
To achieve these security properties, various cryptographic techniques are used such as blind signature [2], zero knowledge proof (ZKP), and homomorphic encryption, etc. Based on the Mixnet technique [3], e-voting schemes have been proposed to protect the privacy of voters if at least one of the mix servers is honest [4, 5]. In [6, 7], homomorphic encryption is used to aggregate all votes, while the individual vote is still kept secretly. In [8], a secure anonymous e-voting system is presented using the hardness of Discrete Logarithm (DL) problem, and in [9] the e-voting scheme is based on secret sharing and secure multi-parties computation.

A lot of progresses have also been achieved in the area of paper-based cryptographic voting scheme [10-13].

Bingo Voting scheme, in which a trusted random number generator (TRNG) with a display is used to guarantee the universal verifiability and coercion-resistance, has been proposed [14-16]. In Bingo Voting, two kinds of random numbers are generated to mask the voters’ intention and guarantee the verifiability of voters’ choices. One is called the dummy random numbers, which are generated during the pre-voting phase, the other is the fresh random numbers, which are used to denote voters’ selections, and are generated by TRNG during the voting process.

However, there are two problems that Bingo Voting cannot solve. First, Bingo Voting is vulnerable to the side channel attack. Since in Bingo Voting, the fresh random numbers are generated in front of the eyes of voters, and displayed as a human readable form in the voting booth, a malicious voter could take a picture of the fresh random number using a secret camera. Therefore, the malicious voter can prove to others which candidate was selected. This is referred to the side channel attack in e-voting. Second, the public acceptance is also important for a practical voting scheme. In Bingo Voting, once leaving the voting booth, the voter has no way to verify whether the receipt reflects his intention due to the fact that the fresh random number cannot be repeated again. Furthermore, considering that the random numbers in practice is often set long enough to ensure the security, the minor difference between the number printed on the receipt and the number displayed on screen of voting machine may be neglected; therefore may confuses the voter and constitutes a psychological obstacle for the public acceptance.

In this paper, an improved Bingo Voting scheme is proposed. Instead of employing a TRNG to generate the fresh random numbers, we make utilization of all candidates’ information to generate a fresh number. Compared with the original Bingo Voting, the proposed scheme achieves two additional goals:

G1. Side channel attack-resistance

In the proposed scheme, no private information is displayed any more in the voting booth. A malicious voter can obtain nothing except the receipt, which avoids the side channel attack from a malicious voter.

G2. Off-site inspection

The voter can still check if his receipt reflects his intention, even if he has left the voting booth. The knowledge on the receipt is enough for a voter’s verification, whereas not enough to prove to others which candidate is selected.

The rest of the paper is organized as follows. We introduce the cryptographic primitives in Section 2, then describe the trust assumption of a practical e-voting scheme in
Section 3. The improved Bingo Voting scheme is given in Section 4, followed by the security analysis in Section 5. Finally the conclusion is given in Section 6.

2. CRYPTOGRAPHIC PRIMITIVES

In this section, we describe the cryptographic primitives that are basic blocks to build both Bingo Voting and the improved version.

2.1 Pedersen’s Commitment

The employment of commitment scheme is often used to allow someone to commit to a chosen value while keeping the value secretly to others, and later the committed value can be revealed. The Pedersen Commitment [17] is based on the hardness of DL problem. Suppose that $q_1$, $q_2$ are both primes with $q_1 | (q_2 - 1)$, and $q_1$ is at least 1024 or 2048 bits long. Let $G_{q_1}$ be the subgroup of $\mathbb{Z}_{q_1}$ of order $q_1$, and set $g$ a generator of $G_{q_1}$. Then for the element $h \in G_{q_1}$, it is computational infeasible to compute

$$\alpha = \log_g h \mod q_1. \quad (1)$$

The Pedersen’s commitment protocol consists of two stages.

Commit Stage: If Alice wishes to commit a value $r$ to Bob, then Alice first randomly selects a number $t$, and computes $C = g^r h^t$, which is then sent to Bob.

Reveal Stage: Alice sends $r$, $t$ to Bob, then Bob can verify whether $C$ is the commitment of $r$.

The Pedersen’s commitment achieves two goals:

1. The commitment $C = g^r h^t$ reveals no information about $r$, and the committer Alice cannot reveal $C$ to another value $r' \neq r$.
2. The same value $r$ can be committed to different commitments $C = g^r h^t$ and $C' = g^{r'} h^{t'}$, where $t \neq t'$. By revealing $t - t'$, Alice can prove to Bob that $C$ and $C'$ are the commitments of the same value without revealing $r$.

2.2 Zero-Knowledge Proof

Based on Pedersen’s commitment, zero-knowledge proof (ZKP) is introduced. Assume that there are $n$ values $r_1, \ldots, r_n$, Alice commits them by computing

$$C_i = g^{r_i} h^{t_i}, \quad (2)$$

for $i = 1, \ldots, n$, and where $t_1, \ldots, t_n$ are random numbers. Alice proves to Bob that each element of the set $S_C = \{C_1, \ldots, C_n\}$ is really the commitment of a corresponding element of the set $S_r = \{r_1, \ldots, r_n\}$ without releasing the relation between the elements of two sets.

As shown in Fig. 1, the steps are described as follows.
Step 1: Alice commits \( r_1, \ldots, r_n \) again using \( t'_1, \ldots, t'_n \), and shuffles the commitments to obtain a set \( S'_C = \{ C'_1, \ldots, C'_n \} \), then Alice publishes \( S'_C \).

Step 2: The above step is repeated to yield another set \( S''_C = \{ C''_1, \ldots, C''_n \} \) using \( t''_1, \ldots, t''_n \), then Alice also publishes \( S''_C \).

Step 3: Alice opens \( S''_C \) to get a set that should be equal to \( S_r = \{ r_1, \ldots, r_n \} \) by revealing \( (t''_i - t'_i)(1 \leq i \leq n) \), which guarantees that \( S''_C \) is indeed the commitments of \( S_r \).

Step 4: Bob selects a random bit \( b \) and sends it to Alice. If \( b = 0 \), Alice publishes the link between \( S'_C \) and \( S''_C \) by revealing \( (t_i - t'_i)(1 \leq i \leq n) \). Otherwise, Alice releases the relation between \( S'_C \) and \( S''_C \) by revealing \( (t''_i - t'_i)(1 \leq i \leq n) \).

It is not difficult to see that the probability that Alice cheats Bob is \( 1/2 \). If the above process is executed \( k \) times, the probability that Bob can be cheated becomes \( 1 - \frac{1}{2^k} \).

### 3. TRUST ASSUMPTION

The participants in an e-voting system include Election Authority, Helper Organizations, and Voters. For a practical voting scheme, some trust assumptions are essential.

**Election Authority:** Its responsibilities include: the distribution of ballots, votes aggregation, information publishing, and results announcement. The election authority is not always trusted, which may coerce the voter to influence the result, or destroy the privacy of the ballot.

**Helper Organizations:** The role of the helper organizations is to help voters correctly follow the protocol. We assume that at least one helper organization is honest. If a corrupted helper organization executes incorrectly, the voter can turn to another organization for help. Thus, the dishonest organization will be punished.

**Voters:** Each voter casts the vote according his intention, in fact the voter is also a potential adversary when it comes to coercion, therefore the scheme should be designed to
prevent a malicious voter from selling the vote.

Furthermore, Voting Machine, Voting Booth, Bulletin, and Verification Device are important for implementing the protocol.

**Voting Machine:** The voting machine receives the ballot from a voter, and generates the corresponding receipt. Then the voting machine tallies the ballots, and publishes the result together with the corresponding proof.

The security threats for the voting machine can be classified into two categories: subliminal channel attacks that the election equipment and data maybe violate the security, and side channel attacks that some external equipment such as a cell camera is used to violate the privacy of the voting. We assume that the subliminal channel attack can be avoided by the audition and inspection of the help organizations and the voters. The work in [16] gives a detailed description of the implementation of the voting machine, which guarantees this assumption is reasonable. Moreover, “software independence” is also the principle in designing a voting machine, a voting machine is software-independent if an undetected change or error in its software cannot cause an undetectable change or error in an election outcome [18].

**Voting Booth:** The voting booth is designed to guarantee the voter’s privacy since other people cannot learn what happens in the voting booth.

**Bulletin Board:** The bulletin board is a platform for authenticated content from the voting machines, which requires that nobody can alter the published information in the bulletin board.

**Verification Device:** The verification device helps any participant to verify the receipt, which can be equipped in the voting booth or anywhere else.

### 4. THE IMPROVED BINGO VOTING SCHEME

#### 4.1 Review of Bingo Voting

Bingo Voting relies on a fresh random number generated by TRNG to reflect and mask the voter’s intention in the voting booth, which consists of Pre-Voting Phase, Voting Phase, and Post-Voting Phase.

In the pre-voting phase, \( l \) dummy votes are generated for a candidate \( P_i (i = 1, \ldots, n) \) where \( l \) is the number of the eligible voters, their commitments are shuffled and published, meanwhile the dummy votes are unknown for all since it is computational infeasible to obtain the dummy votes with the commitments.

In the voting phase, when \( P_i \) is selected by a voter \( V_t \), TRNG generates and displays a fresh random number \( R_t \) in the voting booth. If \( R_t \) is correctly associated to \( P_i \) on the receipt, \( V_t \) ensures the receipt reflects his intention. At the same time, each unselected candidate \( P_j (j \neq i) \) is assigned a dummy vote, nobody can distinguish the difference between the dummy vote and the fresh random number, the receipt-freeness is achieved.
Then, the receipt is published on the bulletin, voter can easily check whether the published information is same as the receipt.

In the post-voting phase, the result is published together with the proof that consists of: (1) the commitments of the unused dummy votes are opened, 2) each unopened dummy vote is proved that has been used in the voting phase, but nobody knows which receipt a dummy vote has been used on. When \( P_i \) received a vote, a dummy vote of \( P_i \) was not needed for the receipt, therefore, the tallying result equals to the number of unused dummy votes for \( P_i \). With the proof, the voter ensures his ballot is counted correctly.

In Bingo Voting, if a malicious voter takes the picture of the display of TRNG in the voting booth, he can prove to the vote buyer which candidate he has voted, which makes the scheme vulnerable to the side channel attack. In this work, an improved Bingo Voting is proposed, the main contribution is to use a verifiable random number (VRN) to reflect and mask the voter’s intention instead of the fresh random number displayed in the voting booth. Nobody can obtain additional knowledge except the receipt if the scheme is functionally executed, which prevents the malicious voter from getting the evidence of selling votes.

4.2 An Example

We introduce an example to illustrate our basic idea, and all computations are performed over a finite field \( F_p \), where \( p \) is a secure prime. Assume that there are four candidates \( P_1, P_2, P_3, P_4 \) and five voters \( V_1, V_2, V_3, V_4, V_5 \), and the identifier of the candidates \( P_1, P_2, P_3, P_4 \) are also elements of \( F_p \).

Pre-Voting Phase: For simplicity, assume that the voting machine generates the same number of dummy votes for each candidate according the number of the eligible voters, which consists of the candidate’s identifier and a random number. As shown in Fig. 2, there are five dummy votes \((P_1, r_{11}), (P_1, r_{12}), (P_1, r_{13}), (P_1, r_{14}), (P_1, r_{15})\) in the dummy votes pool of \( P_1 \). All dummy votes are committed, shuffled and published, which assures that it is computational infeasible to learn the relation between the dummy votes and the published commitments.

Voting Phase: Assume that the candidate \( P_2 \) is selected by the voter \( V_1 \), the voting machine assigns each unselected candidate a dummy vote, listed as \((P_1, r_{11}), (P_2, r_{12}), (P_3, r_{13}), (P_4, r_{14})\). Then, the voting machine generates a polynomial \( A(x) \) passing through these three pairs, and computes \( R_1 = A(P_2) \). Then, the receipt of the voter \( V_1 \) is \((P_1, r_{11}), (P_2, R_1), (P_3, r_{13}), (P_4, r_{14})\), which is printed and published.

Similarly, assume that \( P_3, P_2, P_4, P_1 \), are selected by voters \( V_2, V_3, V_4, V_5 \) respectively, their corresponding receipts are also generated and printed, which is shown in Fig. 3. Simultaneously, the used dummy votes are marked as “used” in the list. As shown in Fig. 4, the used dummy votes are marked with shadow in the dummy vote pool.

The voters and any other observers can check the validity of the printed receipts. For example, the voter \( V_1 \) verifies his receipt by recovering \( A(x) = a_2x^2 + a_1x + a_0 \) from \((P_1, r_{11}), (P_3, r_{13}), (P_4, r_{14})\), and checking if \( R_1 = a_2P_2^2 + a_1P_2 + a_0 \) holds. Meanwhile, the observer can also verify if the receipt of \( V_1 \) is legitimate by checking if the same polynomial can be generated with any three pairs from \((P_1, r_{11}), (P_2, R_1), (P_3, r_{13}), (P_4, r_{14})\).
### Dummy votes pool

<table>
<thead>
<tr>
<th>P₁, r₁</th>
<th>P₂, r₂</th>
<th>P₃, r₃</th>
<th>P₄, r₄</th>
<th>P₅, r₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁, r₁</td>
<td>P₂, r₂</td>
<td>P₃, r₃</td>
<td>P₄, r₄</td>
<td>P₅, r₅</td>
</tr>
</tbody>
</table>

#### Commit

<table>
<thead>
<tr>
<th>C₁</th>
<th>C₂</th>
<th>C₃</th>
<th>C₄</th>
<th>C₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₁</td>
<td>C₂</td>
<td>C₃</td>
<td>C₄</td>
<td>C₅</td>
</tr>
<tr>
<td>C₁</td>
<td>C₂</td>
<td>C₃</td>
<td>C₄</td>
<td>C₅</td>
</tr>
<tr>
<td>C₁</td>
<td>C₂</td>
<td>C₃</td>
<td>C₄</td>
<td>C₅</td>
</tr>
</tbody>
</table>

#### Shuffle

[C₅'C₂C₄C₃, C₂'C₅C₃C₄, C₃'C₄C₅C₂, C₄'C₃C₂C₅, C₅'C₁C₄C₃, C₁'C₄C₅C₂, C₃'C₅C₁C₄, C₅'C₂C₃C₁, C₂'C₅C₃C₄, C₃'C₄C₂C₅, C₄'C₃C₁C₄, C₅'C₁C₄C₃, C₁'C₄C₂C₅, C₄'C₃C₅C₁, C₅'C₁C₃C₄, C₁'C₃C₄C₅, C₃'C₂C₅C₁, C₅'C₁C₂C₃, C₂'C₅C₁C₄, C₃'C₂C₄C₅, C₄'C₁C₃C₂, C₅'C₁C₂C₃, C₂'C₅C₁C₄, C₃'C₁C₄C₂, C₄'C₁C₃C₂, C₅'C₁C₂C₃, C₂'C₅C₁C₄, C₃'C₁C₄C₂, C₄'C₁C₃C₂, C₅'C₁C₂C₃].

### Bulletin board

**Fig. 2.** The pre-voting phase.

### Voter's receipts

<table>
<thead>
<tr>
<th>V₁</th>
<th>V₂</th>
<th>V₃</th>
<th>V₄</th>
<th>V₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁</td>
<td>P₁</td>
<td>P₁</td>
<td>P₁</td>
<td>P₁</td>
</tr>
<tr>
<td>R₁</td>
<td>r₂</td>
<td>r₃</td>
<td>r₄</td>
<td>r₅</td>
</tr>
<tr>
<td>P₂</td>
<td>P₂</td>
<td>P₂</td>
<td>P₂</td>
<td>P₂</td>
</tr>
<tr>
<td>r₁</td>
<td>r₂</td>
<td>r₃</td>
<td>r₄</td>
<td>r₅</td>
</tr>
<tr>
<td>P₃</td>
<td>P₃</td>
<td>P₃</td>
<td>P₃</td>
<td>P₃</td>
</tr>
<tr>
<td>P₄</td>
<td>P₄</td>
<td>P₄</td>
<td>P₄</td>
<td>P₄</td>
</tr>
</tbody>
</table>

**Fig. 3.** The list of voter’s receipts.

### Dummy vote pool

<table>
<thead>
<tr>
<th>P₁, r₁</th>
<th>P₂, r₂</th>
<th>P₃, r₃</th>
<th>P₄, r₄</th>
<th>P₅, r₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁, r₁</td>
<td>P₂, r₂</td>
<td>P₃, r₃</td>
<td>P₄, r₄</td>
<td>P₅, r₅</td>
</tr>
</tbody>
</table>

**Fig. 4.** The dummy vote pool in the post-voting phase.
These verification is based on the correctness of the used dummy votes, which will be proved in the next phase.

Since \((P_2, R_i)\) is indistinguishable from \((P_1, r_1^i), (P_3, r_2^i), (P_4, r_4^i)\), the voter \(V_1\) cannot prove to others that \(R_i\) is the fresh random number and \(P_2\) has been selected.

**Post-Voting Phase:** This phase tallies the votes and proves to the result be correct, which includes three steps:

**Step 1:** From the voting phase, we know each time the voter has been voted, only the corresponding dummy votes of the unselected candidates are marked as “used”. Therefore, the number of the votes that a candidate has got equals to the number of each candidate’s unused dummy votes. In Fig. 4, there is only one unmarked dummy vote in the first row, which means that the candidate \(P_1\) gets one vote. Similarly, we know that candidates \(P_2\) and \(P_4\) have got one vote, and the candidate \(P_3\) obtains two votes.

**Step 2:** The commitments \(C_1^1, C_2^1, C_3^1, C_4^1\) are revealed since the dummy votes \((P_1, r_1^1), (P_2, r_1^2), (P_3, r_2^1), (P_4, r_3^1)\) never appears in one receipt, which have nothing to do with the privacy of the published receipts.

**Step 3:** For each published receipt, the voting machine proves that it contains the correct number of dummy random numbers. The voting machine commits \((P_1, R_1), (P_2, R_2), (P_3, R_3), (P_4, R_4)\) to obtain \(C_{R_1}, C_{R_2}, C_{R_3}, C_{R_4}\). The voting machine proves that each element of \(\{C_1^0, C_2^0, C_3^0, C_4^0\}\) is the commitment to an element of \((P_1, r_1^1), (P_2, r_1^2), (P_3, r_1^3), (P_4, r_1^4)\) without leaking the link between these elements. We show the correctness of all the receipts in Fig. 5.

---

**Receipt**

\[
\begin{align*}
(P_1, r_1^1), (P_2, R_1), (P_3, r_3^1), (P_4, r_4^1) & \iff C_1^1, C_R^1, C_1^3, C_1^4 \\
(P_1, r_2^1), (P_2, r_1^2), (P_3, R_2), (P_4, r_4^1) & \iff C_2^1, C_2^2, C_R^2, C_2^4 \\
(P_1, r_3^1), (P_2, r_2^1), (P_3, r_3^2), (P_4, r_4^1) & \iff C_3^1, C_3^3, C_R^3, C_3^4 \\
(P_1, r_4^1), (P_2, r_3^1), (P_3, r_4^2), (P_4, R_4) & \iff C_4^1, C_4^4, C_R^4, C_4^5 \\
(P_1, R_1), (P_2, r_1^2), (P_3, r_3^1), (P_4, r_4^1) & \iff C_R^1, C_1^2, C_1^3, C_1^4 \\
(P_1, r_2^1), (P_2, r_2^2), (P_3, r_3^1), (P_4, r_4^1) & \iff C_R^2, C_2^2, C_2^3, C_2^4 \\
(P_1, R_1), (P_2, r_2^2), (P_3, r_3^1), (P_4, r_4^1) & \iff C_R^3, C_3^3, C_3^4, C_3^5 \\
(P_1, R_1), (P_2, r_1^2), (P_3, r_3^1), (P_4, r_4^1) & \iff C_R^4, C_4^4, C_4^5, C_4^6
\end{align*}
\]

ZKP between the receipt and the corresponding set of commitments is published.

Fig. 5. ZKP for the correctness of the used dummy votes.
We can see from the example that the main difference between Bingo Voting and the proposed scheme is that we use all votes’ information to generate the fresh random number, instead of the employment of TRNG in the Bingo Voting. Therefore, the proposed scheme inherits the benefits of Bingo Voting, while resists against the side channel attack.

4.3 Full Description

The proposed e-voting scheme allows a voter to select one from \( n \) candidates, which is denoted as 1-out-of-\( n \). Similar to the example, the proposed e-voting scheme also consists of three phases: pre-voting phase, voting phase, and post-voting phase.

Pre-voting phase
The election authority selects and publishes a secure prime \( p \), all computations are performed over \( \mathbb{F}_p \).

The voting machine generates the same number of dummy votes for each candidate that equals the number of voters, then commits these dummy votes. The commitments are shuffled and published, whereas the dummy votes are still kept secretly.

Specifically, assume that there are \( n \) candidates \( P_1, \ldots, P_n \), and \( l \) eligible voters \( V_1, \ldots, V_l \). Then, \( l \) dummy votes of \( P_i (1 \leq i \leq n) \) are \((P_i, r_i^1), \ldots, (P_i, r_i^l)\), where \( r_i^1, \ldots, r_i^l \in \mathbb{F}_p \) are random number generated under the supervision of help organization and voters. Next, \((P_i, r_i^1), \ldots, (P_i, r_i^l)\) are respectively committed to \( C_i^1, \ldots, C_i^l \) using Pedersen’s commitment scheme. Finally, the commitments are shuffled [3] and published in the bulletin board.

Voting phase

Assume that the candidate \( P_i \) is selected by \( V_t \), the voting machine generates the corresponding receipt using the following steps:

**Step 1:** The voting machine assigns each unselected candidate \( P_j (j \neq i) \) a dummy vote, and each dummy vote is used only once. These \( n - 1 \) dummy votes are listed as \((P_1, r_1^t), \ldots, (P_i - 1, r_i^t), (P_{i+1}, r_{i+1}^t), \ldots, (P_n, r_n^t)\);

**Step 2:** The voting machine generates \( A_i(x) = a_{n-2}x^{n-2} + \ldots + a_1x + a_0 \) of degree \( n - 2 \) for the voter \( V_t \) with the above \( n - 1 \) pairs. The polynomial can be computed by the corresponding linear equations in the following matrix form

\[
\begin{pmatrix}
P_{i-1}^{-2} & \ldots & P_{i-1} & 1 & a_{n-2} & r_1^t \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
P_{i-2}^{-2} & \ldots & P_{i-2} & 1 & a_{n-3} & r_{i-1}^t \\
P_{i+1}^{-2} & \ldots & P_{i+1} & 1 & a_{n-1} & r_{i+1}^t \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
P_n^{-2} & \ldots & P_n & 1 & a_0 & r_n^t
\end{pmatrix}
\]

(3)
Then, the voting machine substitutes the identifier of \( P_i \) into the polynomial \( A_i(x) \) to get \( R_t = A_i(P_i) \), which is associated with the selected candidate \( P_i \). Obviously, the polynomial \( A_i(x) \) is random due to the random feature of \( r_{1t}, \ldots, r_{it}, \ldots, r_{nt} \). Therefore, \( R_t = A_i(P_i) \) can be used to replace the fresh random number in Bingo Voting;

**Step 3:** \((P_1, r_{1t}^1), \ldots, (P_{i-1}, r_{it}^{i-1}), (P_i, R_t), (P_{i+1}, r_{it}^{i+1}), \ldots, (P_n, r_{nt}^n)\) are printed as the receipt of \( V_t \) and published in the bulletin board.

Whether the receipt reflects the voter’s intention correctly can be checked with the following steps:

**Step 1:** The voter \( V_t \) can use a verification device to verify whether the equation \( R_t = A_i(P_i) \) holds or not with the printed information \((P_1, r_{1t}^1), \ldots, (P_{i-1}, r_{it}^{i-1}), (P_i, R_t), (P_{i+1}, r_{it}^{i+1}), \ldots, (P_n, r_{nt}^n)\). If it holds, the voter \( V_t \) believes the receipt reflects his intention correctly. Certainly, this verification is based on the assumption that the unselected candidates is assigned the corresponding dummy votes. If an adversary tries to modify the data inside the machine, this subliminal channel attack can be detected by Help Organizations.

**Step 2:** Any participant can check if the same polynomial can be reconstructed with arbitrary \( n-1 \) pairs from \((P_1, r_{1t}^1), \ldots, (P_{i-1}, r_{it}^{i-1}), (P_i, R_t), (P_{i+1}, r_{it}^{i+1}), \ldots, (P_n, r_{nt}^n)\). If yes, the receipt is believed to be legitimate.

**Step 3:** Nobody can distinguish \( R_t \) from the dummy votes \( r_{1t}^1, \ldots, r_{it}^{i+1}, \ldots, r_{nt}^n \), which masks the voter’s intention to achieve the receipt-freeness.

The above verification can be executed in voting booth or anywhere else, the off-site inspection eliminates the psychological obstacle of the receipt verification for the voter.

**Post-voting phase**

To guarantee the ballots tallied correctly, it is an essential requirement that each candidate has received the same number of dummy votes, which can be done by publishing all the candidates’ commitments of dummy votes together with a proof. For more details, please refers to [14, 16].

First, the voting machine tallies the ballots, and publishes the result on the bulletin board. Once a candidate has received a vote, the corresponding dummy vote of this candidate was not needed for the receipt. Therefore, the number of the votes that the candidate has got equals to the number of each candidate’s unused dummy votes if no voter is absent.

Next, the voting machine opens the commitments of unused dummy votes.

Finally, the voting machine publishes the ZKP between each receipt and the corresponding set of commitments, which assures that each unopened (used) commitment is indeed printed on one receipt while not destroying the privacy.

**Remark 1:** In case of 1-out-of-2 election, the polynomial cannot be generated with the above method, since there is no unique polynomial passing through one point. Here we give a slight modification to make it available.

In the pre-voting phase, the voting machine selects and publishes a random number
In the voting phase, assume that $P_2$ is selected by the voter $V_t$, there is a unique linear polynomial $A(x) = a_1x + a_0$ which passes through, then $A(P_2)$ is attached to $P_2$.

Moreover, the proposed 1-out-of-$n$ voting scheme can easily be generalized to $t$-out-of-$n$ that $t$ of $n$ candidates are selected by the voter. A polynomial of degree $n - t - 1$ is generated with $n - t$ unselected candidates’ dummy votes, then $t$ selected candidate’s identification are substituted to the polynomial to get $t$ verifiable random numbers, which are attached to the corresponding candidates to create the receipt.

**Remark 2:** The assumption that the voting machine assigns the dummy votes correctly is essential for the security of the voting scheme, which can be implemented by the technical and administrative approaches. For more details, please refer to [19, 20].

A preliminary version of the proposed scheme is given in [21]. However, the claim that the voting scheme does not rely on TRNG is not really true since TRNG is still used to generate the dummy votes, though the fresh trusted random number is not necessary for reflecting and masking the voter’s intention. In fact, the main threat of Bingo voting is the side channel attack from the malicious voter, which must be prevented.

## 5. SECURITY ANALYSIS

The improved voting scheme not only inherits the correctness, universal verifiability and receipt-freeness of Bingo Voting, but also achieves two additional security features: side channel attack-resistance and the receipt’s off-site inspection.

**Correctness**

The voter $V_t$ ensures that the receipt is generated correctly by verifying the equation $R_t = A(P_i)$, ensures that the published information is correct by checking the bulletin board and the receipt, and ensures that the result is correct with the revealed the commitments of the unused dummy votes and the published ZKP that unopened commitment has indeed been used in one receipt.

**Universal Verifiability**

The universal verifiability includes two aspects, individual verifiability and public verifiability.

Individual verifiability means that a voter can verify if the receipt reflects and masks his intention. In fact, the voter $V_t$ is the final determinant of $A(x)$ since $(P_1, r_{t1})$, ..., $(P_{i-1}, r_{ti-1})$, $(P_i, r_{ti})$, ..., $(P_n, r_{tn})$ are determined by the voter $V_t$. With the receipt, the voter $V_t$ can recover the polynomial $A(x)$, and check the equation $R_t = A(P_i)$. If yes, the voter ensures that the receipt reflects and masks his intention.

Public verifiability guarantees that any observer can verify if the published receipt $(P_1, r_1)$, ..., $(P_{i-1}, r_{i-1})$, $(P_i, R_i)$, $(P_{i+1}, r_{i+1})$, ..., $(P_n, r_n)$ is legitimate. Since $R_i = A(P_i)$, the degree of the polynomial passing through $n$ points $(P_1, r_1)$, ..., $(P_{i-1}, r_{i-1})$, $(P_i, R_i)$, $(P_{i+1}, r_{i+1})$, ..., $(P_n, r_n)$ is not $n - 1$, but $n - 2$, which means that arbitrary $n - 1$ pairs can recover the same polynomial. Then, any observer can verify whether the published receipt is legal and integral, and anyone can report the illegal of the receipt to the helper organizations.
Lemma 1: If the dummy votes are really random, $R_t = A_t(P_i)$ is also random.

Proof: With the dummy votes of unselected candidates, we could get $A_t(x) = a_{n+2}x^{n+2} + ... + a_2x + a_0$ passing through $n-1$ points $(P_1, r_1)$, ..., $(P_{i+1}, r_{i+1})$, $(P_{i-1}, r_{i-1})$, ..., $(P_n, r_n)$. With the knowledge of Vandermonde Determinant, we obtain the following equation from Eq. (3) when $P_i \neq P_j$, $(1 \leq i, j \leq n)$

$$
\begin{bmatrix}
    a_{n+2} \\
    \vdots \\
    a_{n-1} \\
    a_{n-1} \\
\end{bmatrix}
= 
\begin{bmatrix}
    P_{i-1}^{n+2} & \ldots & P_{i-1} & 1 & r_i^{\frac{1}{i-1}} \\
    \vdots & \ldots & \vdots & \vdots & \vdots \\
    \vdots & \ldots & \vdots & \vdots & \vdots \\
    P_{i+1}^{n+2} & \ldots & P_{i+1} & 1 & r_i^{\frac{1}{i+1}} \\
\end{bmatrix}
\left(\begin{bmatrix}
    1 \\
    \vdots \\
    1 \\
    1 \\
\end{bmatrix}
\right)
$$

Then, we have

$$
R_i = A_t(P_i) = (P_i^{n+2}, ..., P_i, 1)
$$

Since $r_i^{\frac{1}{i-1}}$, ..., $r_i^{\frac{1}{i+1}}$, ..., $r_n$ are randomly generated under the supervision of the helper organizations and the voters, obviously, $R_i = A_t(P_i)$ is also random.

Theorem 1: It is computational infeasible for a corrupted party to forge the receipt if the scheme is executed functionally.

Proof: Recall that a polynomial $A_t(x)$ is generated using the unselected candidates’ dummy votes $(P_1, r_1)$, ..., $(P_{i-1}, r_{i-1})$, $(P_{i+1}, r_{i+1})$, ..., $(P_n, r_n)$. After substitution, a fresh random number $R_t$ is attached to the selected candidate $P_i$, which reflects and masks the voter’s intention correctly.

First, if one of the dummy votes $(P_1, r_1)$, ..., $(P_{i-1}, r_{i-1})$, $(P_{i+1}, r_{i+1})$, ..., $(P_n, r_n)$ is forged, the forgery would be detected by employing ZKP for the correctness of each used dummy vote.

Next, if $R_t$ is forged, this forgery can be detected with the probability $1-1/p$. We know that $R_t = A_t(P_i)$ is randomly distributed over $F_p$ from Lemma 1, and it is computational infeasible for the adversary to forge $R_t$ due to the fact that $p$ is chosen big enough.

Therefore, the proposed e-voting scheme ensures that the receipt cannot be forged.

Receipt-Freeness

Receipt-freeness means that the receipt leaks nothing about the voter’s intention. If a voter is corrupted to sell the vote, he needs to prove that his ballot has been casted to a
certain candidate in compliance with the adversary’s request. Next, we show that the proposed scheme is receipt-free.

**Theorem 2:** The receipt leaks nothing about which candidate is selected.

**Proof:** Assume that the voter $V_t$ has cast his ballot to the candidate $P_i$ in obedience to the demand of the vote-buyer, and got the receipt $(P_1, r_1), \ldots, (P_{i-1}, r_{i-1}), (P_i, r_i), (P_{i+1}, r_{i+1}), \ldots, (P_n, r_n)$ from the voting machine.

Next we show that, in other people’s eyes, there is no difference between the selected $(P_i, r_i)$ and the unselected $(P_j, r_j)$. Since the correctness of the used dummy votes in the post-voting phase is proved using ZKP, nobody can distinguish the fresh random number $r_i$ from the dummy votes. Then, if the role of $P_i$ is replaced with $P_j$, i.e., the same $A_i(x)$ can be retrieved with all points except $(P_j, r_j)$, and the equation $r_i = A_i(P_j)$ also holds. Even if the voter $V_t$ claims that the candidate $P_i$ is actually selected, the vote-buyer cannot ensure that the selected candidate is not $P_j$, but $P_i$, since each pair of $(P_1, r_1), \ldots, (P_{i-1}, r_{i-1}), (P_i, r_i), (P_{i+1}, r_{i+1}), \ldots, (P_n, r_n)$ is equally involved in recovering the polynomial and verifying the equation.

Therefore, the receipt leaks nothing about which candidate is selected.

**Side Channel Attack Resistance**

The proposed voting scheme can resist side channel attacks from the malicious voter. In Bingo Voting, a TRNG with display is used to guarantee the secrecy and verifiability. If a malicious voter takes a secret camera into the voting booth, he can prove to others which number is the fresh random number by taking a picture of the display, which is the main means of side channel attack in e-voting.

In the proposed scheme, the generation and substitution of interpolation polynomials are all performed inside the voting machine, from the viewpoint of practice, the subliminal channel attack can be avoided, which means that the malicious voter with a secret camera can obtain nothing.

Receipt-freeness guarantees that the receipt leaks nothing about the voter’s intention, and resists the side channel attack from the corrupted voter by proving his vote. These two properties guarantee that the proposed voting scheme is coercion-resistant.

**Off-site Inspection**

In Bingo Voting, the voter must check the number printed on the receipt and the number displayed in the TRNG on the spot. Since the fresh random number cannot be repeated again, the voter has no idea to verify the receipt any more after he left the voting booth.

The proposed scheme ensures that a voter can still verify his receipt even if he has left the voting booth, since the information on the receipt is enough for verification, meanwhile keeping the receipt-freeness.

**6. CONCLUSION**

In this paper, we have proposed the universally verifiable and coercion-resistant
e-voting scheme, which employs all the candidates’ information to generate a fresh random number to reflect and mask the voter’s intention. Security analysis shows that the proposed scheme not only inherits the merits of Bingo Voting, but also achieves two additional security goals: side channel attack-resistance and the receipt’s off-site inspection.

ACKNOWLEDGMENT

The work presented in this paper was supported in part by the National Natural Science Foundation of China under grant Nos. 61363069, 61301166, 61572132, 61363030, 61572146, U1501252, 61662016, 61672029.

REFERENCES


Yi-Ning Liu (刘忆宁) is currently a Professor in Guilin University of Electronic Technology, Guilin, China. He received the B.S. degree in Applied Mathematics from Information Engineering University, Zhengzhou, China, in 1995, the M.S. in Computer Software and Theory from Huazhong University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in Mathematics from Hubei University, Wuhan, China, in 2007. His research interests include the analysis of information security protocol, the smart grid, and e-voting.

Wei Guo (郭巍) is now a graduate in the School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin, China. He received his B.S. degree in Information and Computational Science from Guilin University of Electronic Technology, Guilin, China, in 2015. His research interest focuses on the side channel attack of the protocol.
Chi Cheng (程池) is an Associate Professor in School of Computer Science, China University of Geosciences, Wuhan, P.R. China, and is also an International Research Fellow of the Japan Society for the Promotion of Science (JSPS), Institute of Mathematics for Industry, Kyushu University, Japan. He received the B.S. and M.S. degrees in Mathematics from Hubei University, Wuhan, P.R. China, in 2003 and 2006, respectively, and the Ph.D. degree in information and communication engineering from Huazhong University of Science and Technology, Wuhan, P.R. China, in December 2013. His research interests focus on network and information security.

Chingfang Hsu (许靜芳) received the M.Eng. and the Ph.D. degrees in Information Security from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2010 respectively. From September 2010 to March 2013, she was a Research Fellow at the Huazhong University of Science and Technology. She is currently an Assistant Professor at Central China Normal University, Wuhan, China. Her research interests are in cryptography and network security, especially in secret sharing and its applications.

Jun-Yan Qian (钱俊彦) received the B.S. degree from the Anhui Polytechnic University, China, in 1996, the M.S. degree from the Guilin University of Electronic Technology, China, in 2000, and the Ph.D. degree from the Southeast University of China in 2008. He is a Professor of the School of Computer Science and Engineering, Guilin University of Electronic Technology, China. His research interests include formal verification, optimization algorithm, and reconfigurable VLSI design.

Chang-Lu Lin (林昌露) received the Ph.D. degree in Information Security from the state key laboratory of information security, Graduate University of Chinese Academy of Sciences, P.R. China, in 2010. He was a Visiting Scholar in the Information Security Group at Royal Holloway, University of London from July 2011 to January 2012. He was a Visiting Scholar in the Division of Mathematical Science, School of Physical and Mathematical Sciences, Singapore Nanyang Technological University from February 2015 to February 2016. He is interested in cryptography and network security, and has conducted research in diverse areas, including secret sharing, secure multi-party computation, public key cryptography and their applications.