Design and analysis of electronic voting protocols

Quentin Yang
supervised by
Véronique Cortier and Pierrick Gaudry

LORIA - CNRS, Inria Nancy - Grand-Est, Université de Lorraine

JC2, April 2022
What is electronic voting?
The security goals in electronic voting

Security properties
✓ Eligibility
✓ Cast-as-intended
✓ Vote secrecy
✓ Confidence in the result
Coercion-resistance

voteselling.com

Get 50$!!

Sell your vote!
Easy money!

Fast and secure!

Offer ends in 23:58:37!!!
Petition to stop vote buying in Italia (Libero Futuro)

Article about vote buying in Bulgaria (Le Monde)
Coercion-resistance in the literature: The JCJ scheme

Voters

Registrars

Public board

Talliers

$(sk, pk)$

Coercion-resistance in the literature: The JCJ scheme

Coercion-resistance in the literature: The JCJ scheme

Coercion-resistance in the literature: The JCJ scheme

Coercion-resistance in the literature: The JCJ scheme

Analysis of JCJ - Where does the security come from?

Intuitively:
- Indistinguishability of fake and real credentials.
- Untraceability of the shuffle.

Formally:
- We need a definition of coercion-resistance.
The voter either obeys or evades coercion.

<table>
<thead>
<tr>
<th></th>
<th>Real Game</th>
<th>Ideal game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adversary</td>
<td>Takes part in the protocol</td>
<td>Learns the size of the board</td>
</tr>
<tr>
<td></td>
<td>Learns any public information</td>
<td>Learns the result</td>
</tr>
<tr>
<td></td>
<td>Guesses the behavior of the voter</td>
<td></td>
</tr>
</tbody>
</table>

**Definition (Informal)**

A scheme is coercion-resistant if the adversary cannot guess the voter’s behavior with a better probability in the real game than in the ideal game.
Analysis of JCJ - Is it really coercion-resistant?

Voters
(cred)

\text{genCred}

\text{Enc}_{pk}(v), \text{Enc}_{pk}(\text{cred})

+ proofs

\text{Coercer}

\text{Enc}_{pk}(w), \text{Enc}_{pk}(\text{fakecred})

+ proofs

\text{Registars}

\text{Public board}

\text{Public Roster}

\begin{align*}
C_1, C'_1 \\
\vdots \\
C_c, C'_c \\
\vdots \\
C_n, C'_n
\end{align*}

\text{Talliers}

(\text{sk, pk})

\text{Remove duplicates}

\text{Shuffle ballots}

\text{Remove invalid}

\text{Decrypt}
Analysis of JCJ - Is it really coercion-resistant?

<table>
<thead>
<tr>
<th>Adversary</th>
<th>Real Game</th>
<th>Ideal game</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Learns the #duplicates</td>
<td>Learns the size of the board</td>
</tr>
<tr>
<td></td>
<td>Learns the #invalid ballots</td>
<td>Learns the (size of the) result</td>
</tr>
<tr>
<td></td>
<td>Guesses the behavior of the voter</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{size(board)} - \text{size(result)} = \#\text{duplicates} + \#\text{invalid}
\]

Conclusion:

- The adversary has more information in the real game.
- JCJ is **not** coercion-resistant!
Design a coercion-resistant protocol

Voters
(cred)

Registars

\(\text{genCred}\)

Public board

(\(sk, pk\))

Talliers

Remove duplicates

Shuffle ballots

Remove invalid

Decrypt

Public Roster

\[C_1, C_1', \ldots, C_c, C_c', \ldots, C_n, C_n'\]

Coercer

\(\text{Enc}_{pk}(w), \text{Enc}_{pk}(\text{fakecred})\)

\(\text{Enc}_{pk}(v), \text{Enc}_{pk}(\text{cred})\)

+ proofs

+ proofs

\(\text{secret part} \rightarrow \text{public part}\)
Design a coercion-resistant protocol

- **Voters** (cred)
  - genCred
  - Enc_{pk}(v), Enc_{pk}(cred)
  - + proofs

- **Coercer**
  - Enc_{pk}(w), Enc_{pk}(fakecred)
  - + proofs

- **Registars**
  - Secret part
  - Public part

- **Public board**
  - **Public Roster**
    - C_1, C'_1
    - ... 
    - C_c, C'_c
    - ... 
    - C_n, C'_n

- **Talliers** (sk, pk)
  - Cleanse Decrypt

Q. Yang
Design and analysis of electronic voting protocols
JC2, April 2022 12 / 16
Design a cleansing-hiding protocol

We use the CGate primitive to build an MPC protocol:

\[ X = \text{Enc}_{pk}(x) \]

\[ Y = \text{Enc}_{pk}(y) \]

\[ \text{CGate} \rightarrow Z = \text{Enc}_{pk}(xy) \]

It allows logical operations on encrypted bits.
Design a cleansing-hiding protocol

We use the framework and security proof from a previous work:

A toolbox for verifiable tally-hiding e-voting systems

Véronique Cortier, Pierrick Gaudry, Quentin Yang
Université de Lorraine, Inria, CNRS
March 2022

ABSTRACT
In most verifiable electronic voting schemes, one key step is the tally phase, where the election result is computed from the encrypted ballots. A generic technique consists in first applying (verifiable) mixnets to the ballots and then revealing all the votes in the clear. This however discloses much more information than the result of the election itself (that is, the winners) and may offer the possibility to coerce voters.

In this paper, we present a collection of building blocks for designing tally-hiding schemes based on multi-party computations. As an application, we propose the first tally-hiding schemes with no leakage for three important counting functions: D'Hondt, STV, and Majority Judgment. We prove that they can be used to design a private and verifiable voting scheme. We also unveil unknown flaws or leakage in several previously proposed tally-hiding schemes.

individual votes. For verifiability, each trustee produces a zero-knowledge proof of correct (partial) decryption so that anyone can check that the result indeed corresponds to the encrypted ballots. The second main approach is based on verifiable mixnets. The encrypted ballots are shuffled and re-randomized such that the resulting ballots cannot be linked to the original ones [24, 47]. A zero-knowledge proof of correct mixing is produced to guarantee that no ballot has been removed nor added. Several mixers are successively used and then each (rerandomized) ballot is decrypted, yielding the original votes in clear, in a random order.

Homomorphic tally can only be applied to simple vote counting functions, where voters select one or several candidates among a list and the result of the election is the sum of the votes, for each candidate. We note that even in this simple case, the tally reveals more information than just the winner(s) of the election. Mixnet-based tally can be used for any vote counting function since it

(eprint.iacr.org/2021/491)
**Algorithm 3: Real**

```
Require: \( A, K, n_T, t, n_V, n_A, n_C, B \)
1 \( BB \leftarrow \emptyset \)
2 \( pk, s_1, h_1, \ldots, s_{n_T}, h_{n_T} \leftarrow P_{\text{Setup}}(K, n_T, t) \)
3 \( V \leftarrow A() \)
4 \( \{c_i; i \in [1, n_V]\}, R \leftarrow \text{Register}(K, pk, n_V) \)
5 \( (j, \beta) \leftarrow A(\{c_i; i \in V\}, R) \)
6 if \( |V| \neq n_A \vee j \notin [1, n_V] \setminus V \vee \beta \notin [1, n_C] \cup \{\phi\} \) then
7 \quad Return 0
8 \( B \leftarrow B(n_V - n_A, n_C) \)
9 for \( (i, \ast) \in B, i \notin [1, n_V] \) do
10 \quad \( c_i \leftarrow \text{Fakecred}(c_1) \)
11 \( b \leftarrow \{0, 1\} \)
12 \( c \leftarrow c_j \)
13 if \( b == 1 \) then
14 \quad Remove all \( (j, \ast) \in B \)
15 else
16 \quad Remove all \( (j, \ast) \in B \) but the last, which is replaced by \( (j, \beta) \)
17 \quad \( c \leftarrow \text{Fakecred}(c_j) \)
18 \( A'(c) \)
19 for \( (i, \alpha) \in B \) (in this order) do
20 \quad \( M \leftarrow A'(BB) \)
21 \( BB \leftarrow BB \cup \{m \in M \mid m \text{ is valid}\} \)
22 \( BB \leftarrow \{\text{Vote}(c_i, \alpha, pk)\} \)
23 \( M \leftarrow A'(BB) \)
24 \( BB \leftarrow BB \cup \{m \in M \mid m \text{ is valid}\} \)
25 \( X, \Pi \leftarrow P_{\text{tally}}(BB, R, pk, \{h_i, s_i\}, t) \)
26 \( b' \leftarrow A() \)
27 Return 1 if \( b' == b \) else 0
```

**Algorithm 4: Ideal**

```
Require: \( A, K, n_V, n_A, n_C, B \)
1 \( V \leftarrow A(K) \)
2 \( (j, \beta) \leftarrow A() \)
3 if \( |V| \neq n_A \vee j \notin [1, n_V] \setminus V \vee \beta \notin [1, n_C] \cup \{\phi\} \) then
4 \quad Return 0
5 \( B \leftarrow B(n_V - n_A, n_C) \)
6 \( b \leftarrow \{0, 1\} \)
7 if \( b == 1 \) then
8 \quad Remove all \( (j, \ast) \in B \)
9 else
10 \quad Remove all \( (j, \ast) \in B \) but the last, which is replaced by \( (j, \beta) \)
11 \( (\nu_i)_{i \in V}, \beta' \leftarrow A(|B|) \)
12 if \( (b == 1) \land (\beta' \neq \phi) \) then
13 \quad \( B \leftarrow B \cup \{(j, \beta')\} \)
14 \( B \leftarrow B \cup \{(i, \nu_i) \mid i \in V, \nu_i \in [1, n_C]\} \)
15 \( X \leftarrow \text{result}(\text{cleanse}(B)) \)
16 \( b' \leftarrow A(X) \)
17 Return 1 if \( b' == b \) else 0
```

Fig. 3. Definition of coercion-resistance. \( \kappa \) is the security parameter, \( n_T \) the number of talliers, \( t \) the threshold, \( n_V \) the number of voters, \( n_A \) the number of corrupted voters, \( n_C \) the number of voting options and \( B \) the distribution of the sequence of votes.
Summary of our contributions

- Detect a shortcoming in the JCJ scheme
- Present a cleansing-hiding protocol which does not have this flaw
- Propose a new definition for coercion-resistance
- Prove that our protocol is coercion-resistant
- Described the exact leakage in JCJ
- Explain how to adapt our methodology to other schemes

Our work is available on eprint: https://eprint.iacr.org/2022/430