Coercion-Resistant Electronic Elections with Write-In Candidates

2012-08-06
Carmen Kempka
Election with Write-In Candidates

If the tally is published directly, forced abstention attacks are always possible. \(\Rightarrow\) Coercion-resistance is unachievable with write-in candidates.
If the tally is published directly, forced abstention attacks are always possible. ⇒ Coercion-resistance is unachievable with write-in candidates.
If the tally is published directly, forced abstention attacks are always possible. \(\Rightarrow\) Coercion-resistance is unachievable with write-in candidates.
Election with Write-In Candidates

If the tally is represented in a fuzzy way, coercion resistance becomes possible.
Contribution of this Work

- A formalization of fuzziness
- A construction of election schemes that provide end-to-end verifiability without revealing the exact tally
- Coercion-resistant write-in support for Bingo Voting
Some Notions

- A **tally** $T$ is represented by a **representation** $R$.
- A representation $R$ represents the tallies in a set $T_R$.
- A set $\mathcal{R}$ of representations is **complete**, if every tally has a representation $R \in \mathcal{R}$.
- Two tallies are **$\delta$-neighbourhood**, if each candidate’s numbers of votes in the two tallies do not differ by more than $\delta$. 
μ, δ-fuzzable:
Let $T$ be the set of all possible tallies of a given election. We call this election $μ, δ$-fuzzable if there is a complete set $R$ of representations such that for each representation $R ∈ R$ and its set of represented tallies $T_R ⊆ T$:

1. All elements in $T_R$ are $δ$-neighbored to each other.
2. For each candidate there are at least $μ$ different values for his number of votes in $T_R$. 
**μ, δ-Fuzzy Election Scheme**

- $\mathcal{T}$: set of all possible tallies
- $\mathcal{R}$: complete set of representations
- $T_R$: subset of $\mathcal{T}$ represented by $R \in \mathcal{R}$

**μ, δ-fuzzy:**
An election scheme is **μ, δ-fuzzy**, if applied to a μ, δ-fuzzable election, for each representation $R \in \mathcal{R}$:

1. The scheme’s proof of correctness proves that the tally lies in $T_R$.
2. No other information is revealed by this proof or any published data.
General Construction of a Fuzzy Election Scheme

Preconditions:

- Shuffle-based or homomorphic underlying election scheme
- Existence of a trusted authority
- Votes are cast encrypted
- The trusted authority can open those encrypted votes
Fuzzy Election Scheme

A trusted authority gets the encrypted votes
Fuzzy Election Scheme

The trusted authority computes the tally in secret

Candidate 1    Candidate 2    Candidate 3

Candidate 1    Candidate 2    Candidate 3
Fuzzy Election Scheme

The trusted authority chooses a fuzziness vector $f = (f_1, f_2, \ldots)$, according to which she takes aside a small amount of votes.
The trusted authority opens and publishes the remaining votes, according to the underlying election scheme.

<table>
<thead>
<tr>
<th>Candidate 1</th>
<th>Candidate 2</th>
<th>Candidate 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Votes:</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Fuzz:</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
The trusted authority adds $\mu - f_i$ new votes for each candidate.

<table>
<thead>
<tr>
<th>Candidate 1</th>
<th>Candidate 2</th>
<th>Candidate 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Votes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzz:</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>New:</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Fuzzy Election Scheme

The trusted authority proves that there are $\mu$ for each candidate.

<table>
<thead>
<tr>
<th>Candidate 1</th>
<th>Candidate 2</th>
<th>Candidate 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Votes:</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Fuzz:</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>New:</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Carmen Kempka – Coercion-Resistant Electronic Elections with Write-In Candidates
Fuzzy Election Scheme

The trusted authority proves that there are $\mu$ for each candidate.

<table>
<thead>
<tr>
<th></th>
<th>Candidate 1</th>
<th>Candidate 2</th>
<th>Candidate 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Votes</td>
<td>9</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Fuzz</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>New</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Repr.</td>
<td>8-10</td>
<td>5-7</td>
<td>0-2</td>
</tr>
</tbody>
</table>

Carmen Kempka – Coercion-Resistant Electronic Elections with Write-In Candidates
μ, μ-Fuzzy Bingo Voting

Bingo Voting:
- End-to-end-verifiable voting scheme
- Relies on a trusted random number generator
- The voting machine has to be trusted for secrecy, but not for correctness
- The original scheme does not support write-in candidates
Pre-Voting Phase

For each candidate, dummy votes are created (and kept secret) that are later used on the receipt to conceal the real vote:

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>Write-In</th>
</tr>
</thead>
<tbody>
<tr>
<td>com(213)</td>
<td>com(683)</td>
<td>com(172)</td>
</tr>
<tr>
<td>com(769)</td>
<td>com(579)</td>
<td>com(413)</td>
</tr>
<tr>
<td>com(145)</td>
<td>com(123)</td>
<td>com(756)</td>
</tr>
</tbody>
</table>

Published on a public bulletin board:
- Commitments to those dummy votes
- A proof that each candidate got the same amount of dummy votes
Pre-Voting Phase

For each candidate, dummy votes are created (and kept secret) that are later used on the receipt to conceal the real vote:

| com(213) | com(683) | com(172) |
| com(769) | com(579) | com(413) |
| com(145) | com(123) | com(756) |
| Alice    | Bob      | Write-In |

Published on a public bulletin board:

- Commitments to those dummy votes
- A proof that each candidate got the same amount of dummy votes
Voting Phase

The dummy votes are known to the voting machine:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>683</td>
<td>172</td>
</tr>
<tr>
<td>769</td>
<td>579</td>
<td>413</td>
</tr>
<tr>
<td>145</td>
<td>123</td>
<td>756</td>
</tr>
</tbody>
</table>

Now the voter votes:

- Alice
- Bob
- Write-In
### Voting Phase

The dummy votes are known to the voting machine:

<table>
<thead>
<tr>
<th></th>
<th>Alice</th>
<th>Bob</th>
<th>Write-In</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>769</td>
<td>579</td>
<td>413</td>
</tr>
<tr>
<td>145</td>
<td>123</td>
<td></td>
<td>756</td>
</tr>
</tbody>
</table>

Now the voter votes:

<table>
<thead>
<tr>
<th></th>
<th>Alice</th>
<th>Bob</th>
<th>Write-In</th>
</tr>
</thead>
<tbody>
<tr>
<td>789</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Com (No Write-In)
Voting Phase

The voter can also input names to be included in the representation of the tally without voting for them.

Enter name here...
Post-Voting Phase

The tally is mirrored by the remaining dummy votes:

<table>
<thead>
<tr>
<th></th>
<th>com(769)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>com(145)</td>
<td>com(123)</td>
<td>com(756)</td>
</tr>
<tr>
<td>Alice</td>
<td>Bob</td>
<td>Write-In</td>
</tr>
</tbody>
</table>

Tally of the regular candidates (non-fuzzy):

- Unveil and publish all remaining dummy votes
- Publish all receipts
- Publish a proof that on each receipt, exactly one random number is fresh and the others are dummy votes
Post-Voting Phase

Tally of the write-in candidates (non-fuzzy):

- Mix and open the commitments to the write-in names
- Publish a proof of correct mixing
Post-Voting Phase

Tally of the regular candidates (fuzzy):

1. Step 1: Publish a proof that on each receipt, exactly one random number is fresh and the others are dummy votes.
2. Step 2: Compute and publish a representation of the tally
3. Step 3: Prove the correctness of the tally representation
Post-Voting Phase

Step 2: Publish a fuzzy representation of the tally:

1. Compute tally $T = (t_1, t_2, t_3)$ in secret.
2. Choose vector $f = (f_1, f_2, f_3)$ with:
   - $\sum_{i=1}^{n} f_i \geq \mu$
   - $0 \leq f_i \leq \min t_i, \mu$ for all $i$
3. Publish $R = (t_1 - f_1, t_2 - f_2, t_3 - f_3)$ as the representation of the tally
Step 3: Prove correctness of the fuzzy tally representation:

1. For all \( i \): Open \( t_i - f_i \) unused commitments to dummy votes of the \( i \)th candidate.

2. For all \( i \): Create \( \mu - f_i \) new commitments for the \( i \)th candidate, contents indistinguishable to the dummy votes.

3. Shuffle and open the remaining unopend dummy votes and the new commitments with a proof of correct shuffling. This proves that:
   - There are exactly \( \mu \) commitments for each candidate.
   - Each candidate \( i \) got between \( R_i \) and \( R_i + \mu \) votes.
Post-Voting Phase

Tally of the write-in votes: Analogous to the regular candidates
Coercion resistance with write-in candidates is possible.
What is not covered by this notion of fuzziness:

- Group Coercion
- Pattern Voting

Future work:

- Fuzzy schemes that need no trusted authority
- Relation between fuzziness and database anonymity
Thank you
Paper to this talk: