Unconditional Privacy in Remote Electronic Voting

Theory and Practice

Philipp Locher

2016
Traditional Paper-Based Voting
Traditional Paper-Based Voting
Traditional Paper-Based Voting
Remote E-Voting

BLUE: 23%
RED: 31%
YELLOW: 46%
Remote E-Voting

- Blue: 23%
- Red: 31%
- Yellow: 46%

Diagram showing the process of remote e-voting with binary codes representing votes.
End-to-End Verifiability

\[ f(\text{INPUT}) = \text{OUTPUT} \]
Verifiable E-Voting
Verifiable E-Voting
Mix-Net

Mixer 1  Mixer 2  Mixer 3
Properties of an E-Voting System

Verifiability  The result can be verified (combination of individual and universal verifiability)

Privacy  Voter’s privacy is guaranteed, if possible in an everlasting or unconditional manner

Coercion-Resistance  A briber or coercer does not succeed in trying to influence the vote of a voter
Current E-Voting Schemes

- Verifiability is a must requirement
- Privacy is a must requirement, however it relies either on some computational intractability assumptions or on a number of trusted authorities
- There are approaches for receipt-freeness and coercion-resistance, however most are lacking in usability and/or performance
Contributions

Theoretical:
- A new e-voting scheme offering unconditional privacy
- Further development of the scheme to provide receipt-freeness and coercion-resistance

Practical:
- Developing UniVote, an e-voting system for student board elections
- Implementing a shuffle proof, an important but complex building block in many e-voting schemes
Theoretical Contributions
Cryptographic Preliminaries

- **One-way functions:** \( y = f(x) \) can be computed efficiently but there is no algorithm known to compute \( x = f^{-1}(y) \) efficiently (e.g. \( y = g^x \mod p \))
Cryptographic Preliminaries

- **One-way functions:** $y = f(x)$ can be computed efficiently but there is no algorithm known to compute $x = f^{-1}(y)$ efficiently (e.g. $y = g^x \mod p$)

- **Commitments:** commit oneself to a particular value without revealing the value right away but maybe once in the future (e.g. Pedersen commitment $c = com(r, v) = g^r h^v$)
Cryptographic Preliminaries

- **One-way functions:** \( y = f(x) \) can be computed efficiently but there is no algorithm known to compute \( x = f^{-1}(y) \) efficiently (e.g. \( y = g^x \mod p \))

- **Commitments:** commit oneself to a particular value without revealing the value right away but maybe once in the future (e.g. Pedersen commitment \( c = \text{com}(r, v) = g^r h^v \))

- **Public-key encryptions:** encrypt a message using a publicly known key \( pk \) such that the message can be decrypted only with the knowledge of a secret key \( sk \) (e.g. ElGamal encryption: \( e = \text{enc}_{pk}(r, m) = (g^r, pk^r m) \) with \( pk = g^{sk} \))
Cryptographic Preliminaries

- **One-way functions:** \( y = f(x) \) can be computed efficiently but there is no algorithm known to compute \( x = f^{-1}(y) \) efficiently (e.g. \( y = g^x \mod p \))

- **Commitments:** commit oneself to a particular value without revealing the value right away but maybe once in the future (e.g. Pedersen commitment \( c = \text{com}(r, v) = g^r h^v \))

- **Public-key encryptions:** encrypt a message using a publicly known key \( pk \) such that the message can be decrypted only with the knowledge of a secret key \( sk \) (e.g. ElGamal encryption: \( e = \text{enc}_{pk}(r, m) = (g^r, pk^r m) \) with \( pk = g^{sk} \))

- **Non-interactive zero-knowledge proofs of knowledge:** prove knowledge without revealing anything about the knowledge (e.g. \( \text{NIZKP}[(x) : y = g^x] \))
The Basic Scheme

- **Registration:** the voter selects a private credential \((\alpha, \beta)\) and sends the public credential \(u = \text{com}(\alpha, \beta)\) over an authentic channel to the Bulletin Board.

- **Vote casting:** the voter computes an election credential \(\hat{u} = \hat{g}^{\beta}\), two commitments \(c = \text{com}(r, u)\) and \(d = \text{com}(s, \alpha, \beta)\), and sends the ballot containing the vote, \(\hat{u}, c, d\) and the proof over an anonymous channel to the Bulletin Board.

- **Public Tallying:** all data is retrieved from the Bulletin Board and the final tally is derived from the votes with valid proofs.
The Basic Scheme

- **Registration**: the voter selects a private credential \((\alpha, \beta)\) and sends the public credential \(u = \text{com}(\alpha, \beta)\) over an authentic channel to the Bulletin Board

- **Vote casting**: the voter computes
  - an election credential \(\hat{u} = \hat{g}^\beta\)
  - two commitments \(c = \text{com}(r, u)\) and \(d = \text{com}(s, \alpha, \beta)\)
  - a NIZKP proving that \(u\) committed to in \(c\) is a registered credential, that \((\alpha, \beta)\) committed to in \(d\) is the corresponding private credential and that the same \(\beta\) has been used for \(\hat{u}\)

and sends the ballot containing the vote, \(\hat{u}, c, d\) and the proof over an anonymous channel to the Bulletin Board
The Basic Scheme

- **Registration:** the voter selects a private credential \((\alpha, \beta)\) and sends the public credential \(u = \text{com}(\alpha, \beta)\) over an authentic channel to the Bulletin Board.

- **Vote casting:** the voter computes
  
  - an election credential \(\hat{u} = \hat{g}^\beta\)
  - two commitments \(c = \text{com}(r, u)\) and \(d = \text{com}(s, \alpha, \beta)\)
  - a NIZKP proving that \(u\) committed to in \(c\) is a registered credential, that \((\alpha, \beta)\) committed to in \(d\) is the corresponding private credential and that the same \(\beta\) has been used for \(\hat{u}\)

  and sends the ballot containing the vote, \(\hat{u}, c, d\) and the proof over an anonymous channel to the Bulletin Board.

- **Public Tallying:** all data is retrieved from the Bulletin Board and the final tally is derived from the votes with valid proofs.
The Basic Scheme

- Almost no central infrastructure, only a Bulletin Board
- No trusted authorities (except for fairness)
- Computational intractability assumptions are only required to guarantee correctness during vote casting
- Performance: ballot generation and verification require a logarithmic number of exponentiations and a linearithmic number multiplications
- The Tor network based on onion routing is a practical anonymous channel
The Receipt-Free Scheme

- A voter is allowed to cast multiple ballots
- The sum of all cast votes represents voter’s final vote
- The votes and the election credentials must be encrypted
- A voter gets a receipt for each cast ballot, however the voter cannot prove not to have cast any other ballot
- The votes and the election credentials are mixed before all votes with the same election credential are summed up under encryption
- The summed up votes are decrypted and the final tally determined
The Coercion-Resistant Scheme

- A voter may cast multiple ballots, but only the last vote is included in the final tally.
- Under coercion, the voter follows exactly coercer’s instructions.
- A coercer is unable to recognize whether or not a voter has cast another ballot after coercion.
- This principle is called *deniable vote updating*.
The Coercion-Resistant Scheme

- The votes and the election credentials must be encrypted:
  \[ E = \text{enc}(\hat{h}^\beta, \rho), \quad F = \text{enc}(\text{vote}, \sigma) \]

- To make sure, the information whether or not a vote has been updated is not lost during mixing, the mix-net must be applied to a quadratic number of input encryptions.

- To render the scheme practical for large scale elections, it must be further improved.
The Coercion-Resistant Scheme

The expensive mixing process consists of two steps:

1. Compute the lists $E_i$ and apply to each list an exponential shuffle $E'_i = \text{shuffle}_{\text{exp}}(E_i)$

\[
\begin{pmatrix}
E_1 \\
E_2 \\
E_3 \\
\vdots \\
E_n
\end{pmatrix}
= 
\begin{pmatrix}
E_2/E_1 & E_3/E_1 & E_4/E_1 & \ldots & E_n/E_1 \\
E_1 & E_3/E_2 & E_4/E_2 & \ldots & E_n/E_2 \\
E_1 & E_2 & E_4/E_3 & \ldots & E_n/E_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
E_1 & E_2 & E_3 & \ldots & E_{n-1}
\end{pmatrix}
\]

2. Apply to the list $F = ((F_1, E'_1), \ldots, (F_n, E'_n))$ a re-encryption shuffle $F' = \text{shuffle}_{\text{reEnc}}(F)$
Outline

Introduction

Theoretical Contributions

Practical Contributions

Conclusion
UniCrypt

- Cryptographic library providing the cryptographic building blocks used to implement e-voting systems
- Intended to bridge the gap between cryptography and software development
- Offers type safety on a mathematical level
- Contains an implementation of a shuffle proof
- Implemented in Java
Proof System

- **Proof System**
  - **Sigma**
    - Or
    - Double Discrete Log
    - Poly. Evaluation
    - Poly. Set Membership
  - **Shuffle**
    - ReEncryption
    - Identity
    - Permutation Commitment
    - Inequality of Preimages
  - Preimage
    - Plain
    - And
    - Equality
  - Validity
    - ElGamal Encryption
    - Pedersen Commitment
Wikström/Terelius’s Shuffle Proof

Two steps:

1. Commit to a permutation matrix and prove that the resulting commitment indeed contains a permutation matrix

2. Shuffle the input batch according to the permutation matrix committed to in step 1 and prove additionally that the shuffle function has been correctly applied
An $N \times N$ - matrix $M$ is a permutation matrix if there is exactly one non-zero element in each row and column and if this non-zero element is equal to one

Example:

\[
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
= 
\begin{pmatrix}
x_3 \\
x_1 \\
x_2
\end{pmatrix}
\]
An $N \times N$ - matrix $M$ is a permutation matrix if there is exactly one non-zero element in each row and column and if this non-zero element is equal to one

Example:

$$
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
\end{pmatrix}
= 
\begin{pmatrix}
x_3 \\
x_1 \\
x_2 \\
\end{pmatrix}
$$

Theorem (Permutation Matrix) [TW10]:

$$
\prod_{i=1}^{N} x_i' = \prod_{i=1}^{N} x_i \quad \text{and} \quad M\mathbf{1} = \mathbf{1}
$$

With $X = (x_1, \ldots, x_N)$ a vector of $N$ independent variables and $X' = (x_1', \ldots, x_N') = MX$
UniVote

- An e-voting system for student board elections at Swiss universities
- Mix-Net based approach offering participation privacy
- Requirement of late registration
- Kind of a prototype to demonstrate verifiable e-voting
- Not a perfect system, some strong assumptions and cutbacks
- Verification software by a student project
- The project started in 2012 and UniVote2 in 2014
# UniVote

<table>
<thead>
<tr>
<th>Election</th>
<th>Electorate</th>
<th>Turnout</th>
<th>Turnout %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB StudentInnenratswahl 2013</td>
<td>11'249</td>
<td>1'008</td>
<td>9.0%</td>
</tr>
<tr>
<td>VSBFH Studierendenratswahl 2013</td>
<td>5'720</td>
<td>269</td>
<td>4.7%</td>
</tr>
<tr>
<td>VSUZH-Ratswahl 2013</td>
<td>26'186</td>
<td>3'138</td>
<td>12.0%</td>
</tr>
<tr>
<td>SOL StudRat Wahlen 2013</td>
<td>2'715</td>
<td>276</td>
<td>10.2%</td>
</tr>
<tr>
<td>University of Lucerne: Best Teacher Award 2013</td>
<td>2'723</td>
<td>137</td>
<td>5.0%</td>
</tr>
<tr>
<td>VSBFH Studierendenratswahl 2014</td>
<td>6'662</td>
<td>137</td>
<td>2.1%</td>
</tr>
<tr>
<td>University of Lucerne: Best Teacher Award 2014</td>
<td>2'832</td>
<td>40</td>
<td>1.4%</td>
</tr>
<tr>
<td>SUB StudentInnenratswahl 2015</td>
<td>11'679</td>
<td>1'934</td>
<td>16.6%</td>
</tr>
<tr>
<td>VSUZH-Ratswahl 2015</td>
<td>25'707</td>
<td>2'273</td>
<td>8.8%</td>
</tr>
<tr>
<td>VSBFH Studierendenratswahl 2015</td>
<td>6'431</td>
<td>148</td>
<td>2.3%</td>
</tr>
<tr>
<td>SKUBA Urabstimmung 12. - 16. Oktober 2015</td>
<td>9'880</td>
<td>1'202</td>
<td>12.2%</td>
</tr>
<tr>
<td>University of Lucerne: Best Teacher Award 2015</td>
<td>2'878</td>
<td>116</td>
<td>4.0%</td>
</tr>
<tr>
<td>SOL StudRat Wahlen 2015</td>
<td>2'878</td>
<td>435</td>
<td>15.1%</td>
</tr>
<tr>
<td>VSBFH Studierendenratswahl 2016</td>
<td>6'108</td>
<td>148</td>
<td>2.4%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>123'648</strong></td>
<td><strong>11'261</strong></td>
<td><strong>9.1%</strong></td>
</tr>
</tbody>
</table>

Table: Elections and referendums held with UniVote until mid-2016.
Outline

Introduction

Theoretical Contributions

Practical Contributions

Conclusion
Conclusion

Don’t let e-voting undermine voter’s privacy through the back door!

- The secret ballot longs for unconditional vote privacy
- The public understanding for the problems and challenges in e-voting must be increased
Publications

Theoretical Work:

VOTE-ID 2015  Verifiable Internet Elections with Everlasting Privacy and Minimal Trust; with R. Haenni

FC 2016  Coercion-Resistant Internet Voting with Everlasting Privacy; with R. Haenni und R. E. Koenig

AoT 2016  Receipt-Free Remote Electronic Elections with Everlasting Privacy; with R. Haenni

Practical Work:

INFORMATIK 2013  Verifizierbare Internet-Wahlen an Schweizer Hochschulen mit UniVote; with E. Dubuis, S. Fischli, R. Haenni, S. Hauser, R. E. Koenig and J. Ritter

INFORMATIK 2014  A Lightweight Implementation of a Shuffle Proof for Electronic Voting Systems; with R. Haenni