Coercion-Resistant Internet Voting with Everlasting Privacy

Rolf Haenni (Philipp Locher, Reto E. Koenig)

FC'16, Bridgetown, Barbados, February 26, 2016

Outline

Coercion-Resistance

- Strategy 1: Fake Credentials
  - First proposed by Juels, Catalano, Jakobsson (WPES'05)
  - Under coercion, use (indistinguishable) fake credential
  - Submit real vote at any time during the voting period
- Strategy 2: Deniable Vote Updating
  - First proposed by Achenbach et al. (JETS, 2:26–45, 2015)
  - Under coercion, follow the coercer's instructions
  - Update vote shortly before the end of the voting period

Everlasting Privacy

- Strategy 1: Everlasting Privacy Towards the Public
  - First proposed by Demirel et al. (EVT/WOTE'12)
  - Publish perfectly hiding commitments to allow public verifiability
  - Send decommitment values privately to trusted authorities
- Strategy 2: Efficient Set Membership Proof
  - First proposed by Locher and Haenni (VoteID'15)
  - Submit vote over anonymous channel
  - Prove eligibility using perfectly hiding commitment and zero-knowledge proofs

Adversaries

- Present adversary ...
  - tries to manipulate the election outcome, e.g. by coercing voters
  - acts before, during, or shortly after an election
  - is polynomially bounded
- Future adversary ...
  - tries to break vote privacy
  - acts at any point in the future
  - has unlimited computational power

Outline

- Introduction
- Protocol Overview
- Cryptographic Preliminaries
- Detailed Protocol Description
- Properties and Performance
- Conclusion

Involved Parties

- Election administration
- Voters
- Public bulletin board
- Trusted authorities (threshold decryption, mixing)
- Verifiers (the public)

Step 1: Registration

The voter ...

- creates a pair of private and public credentials
- sends the public credential to the election administration (over an authentic channel)

Step 2: Election Preparation

The election administration . . .

- sends the list of public voter credentials to bulletin board

Step 3: Vote Casting

The voter ...

- creates ballot consisting of
  - commitment to public credential
  - commitment to private credential
  - encrypted 'election credential' (used to detect duplicates)
  - encrypted vote
  - Non-interactive zero-knowledge proofs that commitments and encryptions have been formed properly
- sends ballot to bulletin board (over an anonymous channel)

Step 4: Tallying

The trusted authorities . . .

- retrieve ballots from bulletin board
- drop ballots with invalid proofs
- detect and eliminate updated votes
- threshold decrypt remaining encrypted votes
- drop ballots with invalid votes
- compute election result

in a verifiable manner

Outline

- Introduction
- Protocol Overview
- Cryptographic Preliminaries
- Detailed Protocol Description
- Properties and Performance
- Conclusion

Cryptographic Setup

- Group $\mathcal{G}_p$ of prime order p
- Sub-group $\mathbb{G}_q \subset \mathbb{Z}_p^*$ of prime order $q \mid (p-1)$
- Independent generators $g_0, g_1 \in \mathcal{G}_p$ and $h_0, h_1, h_2 \in \mathbb{G}_q$
- Assume that DL is hard in $\mathcal{G}_p$ and DDH is hard in $\mathbb{G}_q$

Set Membership Proof

- Goal: prove that a committed value belongs to a given set

$$NIZKP[(u, r) : C = com(u, r) \wedge u \in \mathbf{U}]$$

- Secret inputs
  - $u, r \in \mathbb{Z}_p$
- Public inputs
  - Commitment $C = com(u, r) \in \mathcal{G}_p$
  - Set $\mathbf{U} = \{u_1, \ldots, u_N\}$ of values $u_i \in \mathbb{Z}_p$

Polynomial Evaluation Proof

- Let $P(X) = \prod_{i=1}^{N}(X - u_i)$ satisfying $P(u_i) = 0$ for all $u_i \in U$

$$NIZKP[(u, r) : C = com(u, r) \wedge u \in \mathbf{U}]$$
$$\Longleftrightarrow$$
$$NIZKP[(u, r) : C = com(u, r) \wedge P(u) = 0]$$

- Efficient protocol by Bayer and Groth (2013)

DL-Representation Proof

- Goal: prove that a commitment contains a DL-representation of another committed value

$$NIZKP[(u, r, v_1, \ldots, v_n, s) : \bigwedge \begin{pmatrix} C = com(u, r) \\ D = com(v_1, \ldots, v_n, s) \\ u = h_1^{v_1} \cdots h_n^{v_n} \end{pmatrix}]$$

- Secret inputs
  - $u, r \in \mathbb{Z}_p$
  - $v_1, \ldots, v_n, s \in \mathbb{Z}_q$
- Public inputs
  - Values $h_1, \ldots, h_n \in \mathbb{G}_q$
  - Commitment $C = com(u, r) \in \mathcal{G}_p$
  - Commitment $D = com(v_1, \ldots, v_n, s) \in \mathbb{G}_q$
- For $n = 2$, efficient protocol by Au, Susilo, Mu (2010)

Verifiable Shuffle

- General verifiable shuffle: $(\mathbf{E}', \pi) = \mathit{shuffle}_f^{\phi}(\mathbf{E}, k_1, \ldots, k_n)$
  - Input list $\mathbf{E} = (E_1, \ldots, E_n)$
  - Random permutation ϕ
  - Keyed one-way function f
  - Keys $k_1, \ldots, k_n$
  - Output list $\mathbf{E}' = (E'_1, \ldots, E'_n)$, where $E'_{\phi(i)} = f(E_i, k_i)$
  - Proof of shuffle π
- In our protocol, we use two shuffle instances
  - Exponentiation: $f(E, k) = E^k$
  - Re-encryption: $f(E, k) = \mathit{reEnc}_{pk}(E, k)$

Outline

- Introduction
- Protocol Overview
- Cryptographic Preliminaries
- Detailed Protocol Description
- Properties and Performance
- Conclusion

Step 1: Registration

The voter . . .

- creates a pair of private and public credentials
- sends the public credential to the election administration (over an authentic channel)

Step 1: Registration

The voter ...

- creates a pair of private and public credentials

$$\alpha, \beta \in_R \mathbb{Z}_q$$
$$u = h_1^\alpha h_2^\beta \in \mathbb{G}_q$$

- sends the public credential u to the election administration (over an authentic channel)

Step 2: Election Preparation

The election administration . . .

- sends the list of public voter credentials to bulletin board

Step 2: Election Preparation

The election administration ...

- defines the list of public voter credentials

$$\mathbf{U} = \{(V_1, u_1), \ldots, (V_N, u_N)\}$$

- computes coefficients $\mathbf{A} = (a_0, \ldots, a_N)$ of polynomial

$$P(X) = \prod_{i=1}^{N}(X - u_i) = \sum_{i=0}^{N} a_i X^i$$

- selects fresh independent election generator $\hat{h} \in \mathbb{G}_q$
- publishes $(\mathbf{U}, \mathbf{A}, \hat{h})$ on bulletin board

Step 3: Vote Casting

The voter . . .

- creates ballot consisting of
  - commitment to public credential
  - commitment to private credential
  - encrypted 'election credential' (used to detect duplicates)
  - encrypted vote
  - Non-interactive zero-knowledge proofs that commitments and encryptions have been formed properly
- sends ballot to bulletin board (over an anonymous channel)

Step 3: Vote Casting

The voter . . .

- creates ballot $B = (C, D, E, F, \pi_1, \pi_2, \pi_3)$ consisting of
  - commitment to public credential $C = com(u, r)$
  - commitment to private credential $D = com(\alpha, \beta, s)$
  - encryption of 'election credential' $E = enc_{pk}(\hat{h}^{\beta}, \rho)$
  - encrypted vote $F = enc_{pk}(v, \sigma)$
  - Non-interactive zero-knowledge proofs π_1, π_2, π_3 (see next slide)
- sends ballot B to bulletin board (over an anonymous channel)

Step 3: Vote Casting

- Polynomial evaluation proof:

$$\pi_1 = NIZKP[(u, r) : C = com(u, r) \wedge P(u) = 0]$$

- DL-Representation proof:

$$\pi_2 = NIZKP[(u, r, \alpha, \beta, s) : \bigwedge \begin{pmatrix} C = com(u, r) \\ D = com(\alpha, \beta, s) \\ u = h_1^{\alpha} h_2^{\beta} \end{pmatrix}]$$

- Standard pre-image proof:

$$\pi_3 = NIZKP[(\alpha, \beta, s, \rho, v, \sigma) : \bigwedge \begin{pmatrix} D = com(\alpha, \beta, s) \\ E = enc_{pk}(\hat{h}^{\beta}, \rho) \\ F = enc_{pk}(v, \sigma) \end{pmatrix}]$$

Step 4: Tallying

The trusted authorities . . .

- retrieve ballots from bulletin board
- drop ballots with invalid proofs
- detect and eliminate updated votes
- threshold decrypt remaining encrypted votes
- drop ballots with invalid votes
- compute election result

in a verifiable manner

Step 4: Tallying

The trusted authorities ...

- retrieve ballots **B** from bulletin board
- drop ballots with invalid proofs π_1, π_2, or π_3 (retain order)

$$(B_1, \ldots, B_n) \subseteq \mathbf{B} \quad \Rightarrow \quad \mathbf{E} = ((E_1, F_1), \ldots, (E_n, F_n))$$

- detect and eliminate updated votes (see next slide)
- threshold decrypt remaining encrypted votes
- drop ballots with invalid votes
- compute election result

in a verifiable manner

Detecting Updated Votes

Step 1: Preparation

- Compute

$$\begin{pmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \\ \mathbf{E}_3 \\ \mathbf{E}_4 \\ \vdots \\ \mathbf{E}_n \end{pmatrix} = \begin{pmatrix} E_1 & E_2/E_1 & E_3/E_1 & E_4/E_1 & \cdots & E_n/E_1 \\ E_1 & E_2 & E_3/E_2 & E_4/E_2 & \cdots & E_n/E_2 \\ E_1 & E_2 & E_3 & E_4/E_3 & \cdots & E_n/E_3 \\ E_1 & E_2 & E_3 & E_4 & \cdots & E_n/E_4 \\ \vdots & & & & & \\ E_1 & E_2 & E_3 & E_4 & \cdots & E_n \end{pmatrix}$$

- Note that $\mathbf{E}_i$ may contain one or multiple encryptions of 1
- If this is the case, then ballot B_i has been updated and must be dropped

Detecting Updated Votes

Step 2: Row-Wise Exponentiation Shuffle

- Compute

$$\begin{aligned}(\mathbf{E}'_1, \pi_1) &= \mathit{shuffle}_{exp}^{\phi_1}(\mathbf{E}_1) \\ \vdots \quad & \qquad \quad \vdots \\ (\mathbf{E}'_n, \pi_n) &= \mathit{shuffle}_{exp}^{\phi_n}(\mathbf{E}_n)\end{aligned}$$

- Note that $\mathbf{E}'_i$ may still contain one or multiple encryptions of 1

Detecting Updated Votes

Step 3: Re-Encryption Shuffle

- Let $\mathbf{F} = ((F_1, \mathbf{E}'_1), \ldots, (F_n, \mathbf{E}'_n))$
- Compute

$$(\mathbf{F}', \pi) = \mathit{shuffle}^{\phi}_{\mathit{reEnc}_{pk}}(\mathbf{F})$$

- Note that $\mathbf{E}''_i$ in $(F'_i, \mathbf{E}''_i) \in \mathbf{F}'$ may still contain one or multiple encryptions of 1

Detecting Updated Votes

Step 4: Decryption

- Decrypt each $\mathbf{E}''_i$ until encryption of 1 is found
- If this is the case for $E''_{ij} \in \mathbf{E}''_i$, ...
  - compute

$$\pi_{ij} = NIZKP[(sk) : dec_{sk}(E''_{ij}) = 1 \wedge pk = h^{sk}]$$

  - drop F'_i
- If this is not the case for $\mathbf{E}''_i$, ...
  - decrypt F'_i
  - prove correctness of decryptions
- Send everything to bulletin board

Outline

Security Properties

- Correctness
  - Find representation (α', β') for some $u \in U$ is equivalent to DL
  - Simulate π_1, π_2, π_3 without (α', β') is equivalent to DL
- Privacy
  - C and D are perfectly hiding
  - π_1, π_2, π_3 are zero-knowledge
  - The future adversary can compute β from $E = enc_{pk}(\hat{h}^{\beta}, \rho)$, but (α', β) satisfying $u' = h_1^{\alpha'} h_2^{\beta}$ can be found for every $u' \in U$
- Coercion-resistance
  - The coercer gets no conclusive receipt that a ballot has not been updated by the voter
  - Checking if $\mathbf{E}_i$ contains an encryption of 1 is equivalent to DL
  - Linking $\mathbf{E}'_i$ to $\mathbf{E}_i$ is equivalent to DL

Performance

- Parameters: N eligible voter, n submitted ballots
- Vote casting
  - $O(\log N)$ exponentiations
  - $O(N \log N)$ multiplications
- Tallying
  - $O(n^2)$ exponentiations
- Verification
  - $O(n^2 + n \log N)$ exponentiations

Outline

Summary

- First protocol that offers everlasting privacy and coercion-resistance simultaneously
- Cryptographic tool
  - Set membership proof (polynomial evaluation proof)
  - DL-representation proof
  - Exponentiation shuffle
  - Re-encryption shuffle
- Limitations
  - Anonymous channel required for vote casting
  - Quadratic tallying and verification
- Application areas: organizations such FIFA, IOC, ICRC, ...