NTNU

Norwegian University of Science and Technology

# Electronic Voting Using Lattice-Based Commitments and Verifiable Encryption

Norsk Kryptoseminar 2020, Kjeller

Carsten Baum, Kristian Gjøsteen, **Tjerand Silde** and Thor Tunge

**Department of Mathematical Sciences,**
**NTNU Trondheim**

NTNU

# Security Definitions I

If a voting system gives the correct answer, relative to some ideally determined collection of ballots and some counting function, we have *integrity*.

If it is hard to determine what ballot a given voter cast, up to what can be deduced from the election outcome, we have *privacy*.

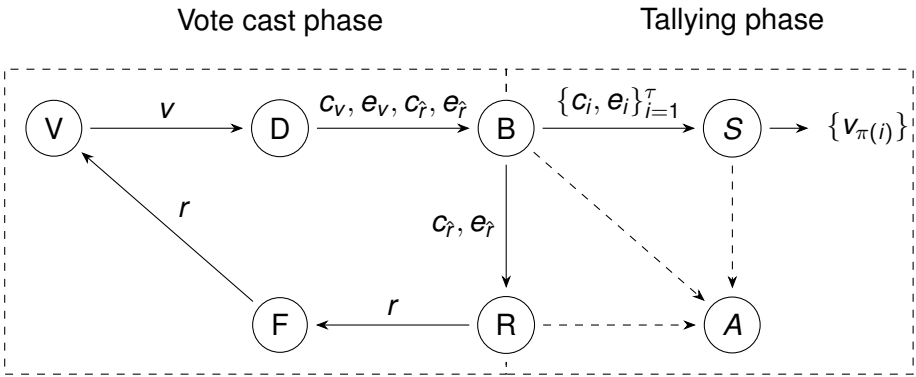
NTNU

# Security Definitions II

If voters can ensure that the ballot of their choice is counted, even when an adversary can control the voter during ballot casting, we have *coercion resistance*.

If the ballot casting and the tally phase produce transcripts that allow voters to verify that the count included their ballots, we have *verifiability*.

# The Protocol Architecture



NTNU

# The Shuffle Protocol I

1. The prover $\mathcal{P}$ and verifier $\mathcal{V}$ receives a set of commitments $\{c_i\}_{i=1}^{\tau}$,
2. $\mathcal{P}$ also receives the set of openings $\{m_i\}_{i=1}^{\tau}$ of the commitments,
3. $\mathcal{V}$ picks a random element $\rho$ and sends $\rho$ to $\mathcal{P}$,
4. $\mathcal{P}$ and $\mathcal{V}$ shifts the commitments to get $M_i = m_i - \rho$,

NTNU

# The Shuffle Protocol II

5. $\mathcal{P}$ picks a random permutation $\pi$ of $\tau$ elements,
6. $\mathcal{P}$ shuffles the messages by defining $\hat{M}_i := M_{\pi^{-1}(i)}$,
7. $\mathcal{P}$ sends the set of shuffled messages $\{\hat{M}_i\}_{i=1}^{\tau}$ to the verifier $\mathcal{V}$,
8. $\mathcal{P}$ proves to $\mathcal{V}$ that $\{\hat{M}_i\}_{i=1}^{\tau}$ is the openings of $\{c_i - \rho\}_{i=1}^{\tau}$.

# The Shuffle Protocol III

The shuffle can be represented as a matrix-equation, where $\theta_i$ is committed randomness, $\beta$ is the challenge and $s_i$ is the response to the challenge:

$$\underbrace{\begin{bmatrix} \hat{M}_1 & 0 & \dots & 0 & 0 \\ M_2 & \hat{M}_2 & \dots & 0 & 0 \\ 0 & M_3 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & M_{\tau-1} & \hat{M}_{\tau-1} \\ 0 & 0 & \dots & 0 & M_\tau \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} s_1 - \theta_1 \\ s_2 - \theta_2 \\ s_3 - \theta_3 \\ \vdots \\ s_{\tau-2} - \theta_{\tau-2} \\ s_{\tau-1} - \theta_{\tau-1} \end{bmatrix}}_{\mathbf{s}} = \underbrace{\begin{bmatrix} -\beta M_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ (-1)^{\tau-1}\beta\hat{M}_\tau \end{bmatrix}}_{\mathbf{b}}.$$

NTNU

# The Shuffle Protocol IV

**Theorem**

*The shuffle protocol is complete.*

**Theorem**

*The shuffle protocol in is sound for any prover $\mathcal{P}^*$ that wins with probability $> \frac{\tau^\delta}{p^N}$.*

**Theorem**

*The shuffle protocol honest-verifier zero-knowledge.*

NTNU

# Proof of Encrypted Opening

To make sure that the vote correct and well-formed, we encrypt the opening of the commitment and prove that:

1. we know underlying the plaintext,
2. the plaintext is the opening of the commitment,
3. the opening is short, and hence, well-formed.

NTNU

# Return Codes I

Assume that the voters have $\omega$ different options in the election. Further, let

— $v_1, v_2, \ldots, v_\omega$ be possible ballots,

— $a$ be a blinding-key for a voter $V$,

— $\mathtt{PRF}_k$ be a pseudo-random function with key $k$,

NTNU

# Return Codes II

The *pre-code* $\hat{r}_j$ corresponding to the ballot $v_j$ is $\hat{r}_j = v_j + a$.

The *return code* $r_j$ corresponding to the ballot $v_j$ is $r_j = \mathrm{PRF}_k(\hat{r}_j)$.

Let $c_a$, $c_{v_j}$ and $c_{\hat{r}}$ be commitments to the blinding key, the ballot and the pre-code.

We can prove in zero-knowledge that $\hat{r}_j = v_j + a$ is correct.

NTNU

# Parameters and Efficiency

A vote $(c_j, e_j, c_{\hat{r}}, e_{\hat{r}}, \Pi_{\hat{r}})$ is of total size $\approx$ 400 KB.

For $\tau$ voters, the ballot box $\mathcal{B}$ receives $\approx 400\tau$ KB of data.

The shuffle proof is of total size $\approx 21\tau$ KB.

| Commitments | Encryption | Verification | Shuffle Proof |
|---|---|---|---|
| 0.12ms | 60ms | 4ms | $12\tau$ms |

NTNU

# Improvements and Future Work

Can we...

- extend the shuffle to handle arbitrary ring elements?
- extend this into a mix-net with more than one shuffle-server?
- extend the return-code mechanism to handle re-voting?
- aggregate zero-knowledge proofs for all equations in the shuffle?

NTNU

# *Thank You! Questions?*

Email: tjerand.silde@ntnu.no
Slides: `www.tjerandsilde.no/talks`