NTNU | Norwegian University of Science and Technology

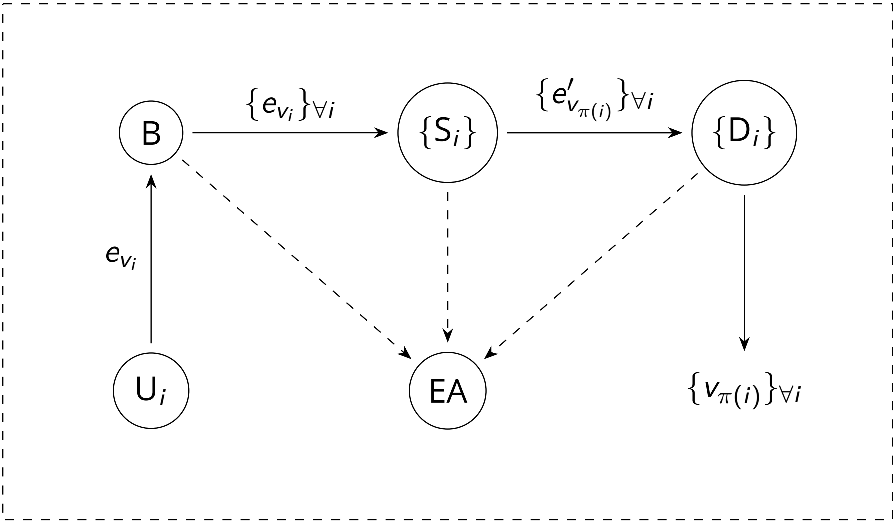
LATTICE-BASED ELECTRONIC VOTING

Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, Patrick Hough, Caroline Sandsbråten, **Tjerand Silde** and Thor Tunge

Contents

Contents

Electronic Voting



Goals

1. Use lattice-based primitives to achieve post-quantum security
2. Build a zero-knowledge protocol to prove correct shuffle of messages
3. Extend the shuffle to handle ciphertexts instead of messages
4. Build a sequential mixing network from the extended shuffle
5. Extend the encryption scheme to support verifiable distributed decryption
6. Combine everything to construct systems for electronic voting

Contents

Commitment

Algorithms:

$\texttt{Com}$: samples randomness $\boldsymbol{r}_m$ and commits to m as $[m] = \texttt{Com}(m; \boldsymbol{r}_m)$.

$\texttt{Open}$: takes as input $([m], m, \boldsymbol{r}_m)$ and verifies that $[m] \stackrel{?}{=} \texttt{Com}(m; \boldsymbol{r}_m)$.

Properties:

$\texttt{Binding}$: it is hard to find $m \neq \hat{m}$ and $\boldsymbol{r}_m \neq \hat{\boldsymbol{r}}_{\hat{m}}$ s.t. $\texttt{Com}(m; \boldsymbol{r}_m) = \texttt{Com}(\hat{m}; \hat{\boldsymbol{r}}_{\hat{m}})$.

$\texttt{Hiding}$: it is hard to distinguish $\texttt{Com}(m; \boldsymbol{r}_m)$ from $\texttt{Com}(0; \boldsymbol{r}_0)$ when given m.

Here we can use the BDLOP18 lattice-based commitment scheme.

Proof of Linearity

Let

$$[x] = \mathtt{Com}(x; \boldsymbol{r}) \quad \text{and} \quad [x'] = [\alpha x + \beta] = \mathtt{Com}(x'; \boldsymbol{r}').$$

Then the protocol Π_{Lin} is a sigma-protocol to prove the relation $x' = \alpha x + \beta$, given the commitments $[x]$, $[x']$ and the scalars α, β.

Here we can use the BDLOP18 proof of linear relations.

Amortized Proof of Shortness

Let

$$[x_1] = \mathtt{Com}(x_1; \boldsymbol{r}_1), \quad [x_2] = \mathtt{Com}(x_2; \boldsymbol{r}_2), \quad \ldots, \quad [x_n] = \mathtt{Com}(x_n; \boldsymbol{r}_n),$$

for bounded norm values x_i. Let Π_A be a sigma-protocol for this relation.

We have approximate proofs by BBCdGL18 and exact proofs by BLNS20.

BGV Encryption

`KeyGen` samples random $a \xleftarrow{\$} R_q$, short $s \leftarrow R_q$ and noise $e \leftarrow \mathcal{N}_{\sigma_E}$.
The algorithm outputs $\mathtt{pk} = (a, b) = (a, as + pe)$ and $\mathtt{sk} = s$.

`Enc` samples a short $r \leftarrow R_q$ and noise $e_1, e_2 \leftarrow \mathcal{N}_{\sigma_E}$, and outputs
$(u, v) = (ar + pe_1, br + pe_2 + m)$.

`Dec` outputs $m \equiv v - su \mod q \mod p$ when noise is bounded by $\lfloor q/2 \rfloor$.

For more details about the encryption scheme see the BGV12 paper.

Contents

Setting

- Public information: sets of commitments $\{[m_i]\}_{i=1}^{\tau}$ and messages $\{\hat{m}_i\}_{i=1}^{\tau}$.
- P knows the openings $\{(m_i, \boldsymbol{r}_{m_i}, f_i)\}_{i=1}^{\tau}$ of the commitments $\{[m_i]\}_{i=1}^{\tau}$,

  and P knows a permutation γ such that $\hat{m}_i = m_{\gamma^{-1}(i)}$ for all $i = 1, ..., \tau$.
- We construct a $4 + 3\tau$-move ZKPoK protocol to prove the statement:

$$R_{\text{Shuffle}} = \left\{ (x, w) \;\middle|\; \begin{array}{l} x = ([m_1], \ldots, [m_\tau], \hat{m}_1, \ldots, \hat{m}_\tau, \hat{m}_i), \\ w = (\gamma, f_1, \ldots, f_\tau, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\tau), \gamma \in S_\tau, \\ \forall i \in [\tau] : \ \texttt{Open}\left(\left[m_{\gamma^{-1}(i)}\right], \hat{m}_i, \boldsymbol{r}_i, f_i\right) = 1 \end{array} \right\}$$

Linear System

First, the verifier sends a challenge ρ to shift all commitments and messages $M_i = m_i - \rho$ and $\hat{M}_i = \hat{m}_i - \rho$ to ensure that all messages are invertible.

Secondly, P draws θ_i uniformly at random, and computes the commitments:

$$\begin{aligned} [D_1] &= \left[\theta_1 \hat{M}_1\right] \\ \forall j \in \{2, \ldots, \tau - 1\} : \; [D_j] &= \left[\theta_{j-1} M_j + \theta_j \hat{M}_j\right] \\ [D_\tau] &= [\theta_{\tau-1} M_\tau] \,. \end{aligned} \tag{1}$$

Linear System

P receives a challenge β from V and computes s_i such that the following equations are satisfied:

$$\begin{aligned} \beta M_1 + s_1 \hat{M}_1 &= \theta_1 \hat{M}_1 \\ \forall j \in \{2, \ldots, \tau - 1\} : \ s_{j-1} M_j + s_j \hat{M}_j &= \theta_{j-1} M_j + \theta_j \hat{M}_j \\ s_{\tau-1} M_\tau + (-1)^\tau \beta \hat{M}_\tau &= \theta_{\tau-1} M_\tau . \end{aligned} \tag{2}$$

Linear System

We can rewrite these equations as a linear system:

$$\begin{bmatrix} M_1 & \hat{M}_1 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M_{\tau-1} & \hat{M}_{\tau-1} \\ (-1)^\tau \hat{M}_\tau & 0 & 0 & \dots & 0 & M_\tau \end{bmatrix} \begin{bmatrix} \beta \\ s_1 \\ \vdots \\ s_{\tau-2} \\ s_{\tau-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

We observe that the determinant of the matrix is equal to $\prod_{i=1}^{\tau} M_i - \prod_{j=1}^{\tau} \hat{M}_j$. If the statement is false, it follows from the Schwartz–Zippel lemma that this system (with high probability) does not have a solution (over the choice of β).

Linear System

P uses the protocol Π_{Lin} to prove that each commitment $[D_i]$ satisfies the equations (2). In order to compute the s_i values, we can use the following fact:

Lemma

Choosing

$$s_j = (-1)^j \cdot \beta \prod_{i=1}^{j} \frac{M_i}{\hat{M}_i} + \theta_j \tag{3}$$

for all $j \in 1, \ldots, \tau - 1$ *yields a valid assignment for Equation* (2).

Protocol

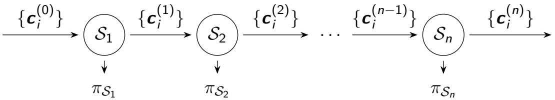
Zero-Knowledge Proof Π_{Shuffle} of Correct Shuffle

Prover, P		Verifier, V
	$\xleftarrow{\rho}$	$\rho \xleftarrow{\$} R_q \setminus \{\hat{m}_i\}_{i=1}^{\tau}$
$\hat{M}_i = \hat{m}_i - \rho$		$\hat{M}_i = \hat{m}_i - \rho$
$M_i = m_i - \rho$		$[M_i] = [m_i] - \rho$
$\theta_i \xleftarrow{\$} R_q, \forall i \in [\tau - 1]$		
Compute $[D_i]$ as in Eq. (1), i.e.		
$[D_1] = [\theta_1 \hat{M}_1], [D_\tau] = [\theta_{\tau-1} M_\tau]$,		
$[D_i] = [\theta_{i-1} M_i + \theta_i \hat{M}_i]$ for $i \in [\tau - 1] \setminus \{1\}$	$\xrightarrow{\{[D_i]\}_{i=1}^{\tau}}$	
	$\xleftarrow{\beta}$	$\beta \xleftarrow{\$} R_q$
Compute $s_i, \forall i \in [\tau - 1]$ as in (3).	$\xrightarrow{\{s_i\}_{i=1}^{\tau-1}}$	
		Use Π_{Lin} to prove that
		(1) $\beta[M_1] + s_1 \hat{M}_1 = [D_1]$
		(2) $\forall i \in [\tau - 1] \setminus \{1\} :\ s_{i-1}[M_i] + s_i \hat{M}_i = [D_i]$
		(3) $s_{\tau-1}[M_\tau] + (-1)^\tau \beta \hat{M}_\tau = [D_\tau]$
		i.e. all equations from (2)

Contents

Extending the Shuffle

- We extend the shuffle to ciphertext vectors instead of single messages
- We create a mix-net as follows:
  1. Re-randomize the ciphertexts
  2. Commit to the randomness
  3. Permute the ciphertexts
  4. Prove that shuffle is correct
  5. Prove that the randomness is short
- Integrity follows from the ZK-proofs
- Privacy if at least one server is honest



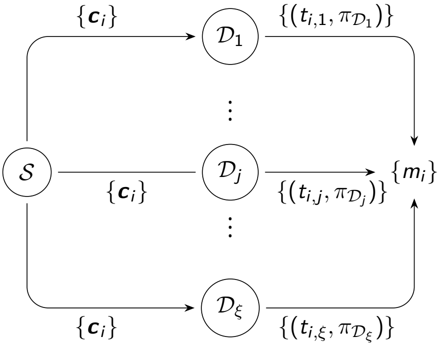
Contents

Distributed Decryption

Verifiable distributed decryption protocol:

- On input key s_j and ciphertext (u, v), sample large noise E_j, output $t_j = s_j u + pE_j$.
- We use Π_{Lin} to prove correct computation.
- We use Π_{A} to prove that E_j is bounded.

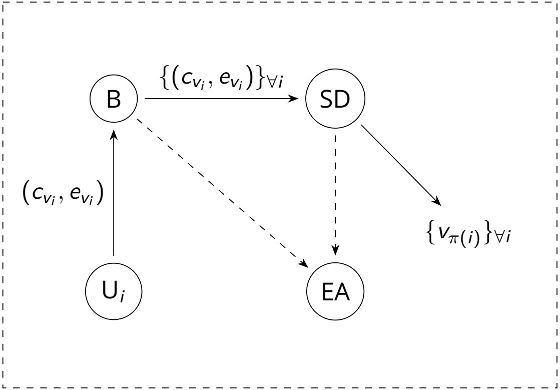
We obtain the plaintext as $m \equiv (v - t \mod q) \mod p$, where $t = t_1 + t_2 + \ldots + t_\xi$.



Contents

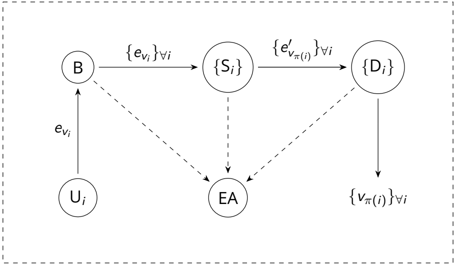
Verifiable Shuffle-Decryption

- SD both shuffle and decrypt the votes.
- Integrity follows from the ZK-proof.
- Privacy if B and SD do not collude.



Verifiable Mix-Net and Distributed Decryption

- $\{\mathcal{S}_i\}$ may consist of many shuffle-servers.
- $\{\mathcal{D}_i\}$ consists of many decryption-servers.
- Integrity follows from the ZK-proofs.
- Privacy holds if the following is true:
  1. at least one shuffle-server is honest and
  2. at least one decryption-server is honest.



Contents

Proof of Shuffle [CT-RSA'21]

- Optimal parameters for the commitment scheme is $q \approx 2^{32}$ and $N = 2^{10}$.
- The prover sends 1 commitment, 1 ring-element and 1 proof per message.
- The shuffle proof is of total size $\approx 22\tau$ KB for τ messages.
- The shuffle proof takes $\approx 27\tau$ ms to compute for τ messages.

Verifiable Mixing and Decryption [CCS'23]

- Optimal parameters for the system is $q \approx 2^{78}$ and $N = 2^{12}$.
- Commitments and ciphertexts are of size ≈ 80 KB each.
- The mixing proof is of size $\approx 370\tau$ KB and takes $\approx 134\tau$ ms.
- The decryption proof is of size $\approx 157\tau$ KB and takes $\approx 101\tau$ ms.

Contents

NTRU Encryption

Key Generation $\mathsf{KeyGen}_{\mathsf{NTRU}}(\mathsf{sp})$. Given input $\mathsf{sp} = (d, p, q, \sigma_{\mathsf{NTRU}}, t, \nu)$:

1. Sample f from $D_{\sigma_{\mathsf{NTRU}}}$; if $(f \mod q) \notin R_q^\times$ or $f \not\equiv 1 \in R_p$, resample.
2. Sample g from $D_{\sigma_{\mathsf{NTRU}}}$; if $(g \mod q) \notin R_q^\times$, resample.
3. If $\|f\|_2 > t \cdot \sqrt{d} \cdot \sigma_{\mathsf{NTRU}}$ or $\|g\|_2 > t \cdot \sqrt{d} \cdot \sigma_{\mathsf{NTRU}}$, restart.
4. Return the secret key $\mathsf{sk} = f$, $\mathsf{pk} = h := g/f \in R_q$.

Encryption $\mathsf{Enc}_{\mathsf{NTRU}}(m, \mathsf{pk})$. Given message $m \in R_p$ and public key $\mathsf{pk} = h$:

1. Sample encryption randomness $s, e \leftarrow S_\nu$.
2. Return ciphertext $c = p \cdot (hs + e) + m \in R_q$.

Decryption $\mathsf{Dec}_{\mathsf{NTRU}}(c, \mathsf{sk})$. Given ciphertext c and secret key $\mathsf{sk} = f$:

1. Compute $m = (f \cdot c \mod q) \mod p$.
2. Return the plaintext message m.

Fig. 1. The encryption scheme `NTRUEncrypt` adapted from [SS13].

NTRU Encryption

NTRU ciphertexts consist of one ring element instead of two. We also wanted to decrease the dimension and moduli to reduce ciphertext sizes, but this was not possible based on current security analysis on ternary secrets.

We analysed the concrete security of NTRU for arbitrary standard deviations σ, and we found that the "fatigue point" for NTRU is $q = 0.0058 \cdot \sigma^2 \cdot d^{2.484}$.

We combined this with exact zero-knowledge proofs of boundedness to get tighter bounds and smaller parameters (but most expensive proofs).

NTRU Mixing Network [ePrint'23]

- Optimal parameters for the overall system is $q \approx 2^{59}$ and $N = 2^{11}$.
- Commitments are ≈ 30 KB and ciphertexts are ≈ 15 KB each.
- The mixing proof is $\approx 130\tau$ KB and decryption proof is $\approx 85\tau$ KB.
- Ciphertexts are $5.3\times$ smaller and the overall system is $2.6\times$ smaller.
- We expect everything to be at least $2\times$ faster (currently benchmarking).

- *Lattice-Based Proof of Shuffle and Applications to Electronic Voting*, Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, Tjerand Silde, and Thor Tunge, published at CT-RSA 2021, eprint.iacr.org/2023/1318
- *Verifiable Mix-Nets and Distributed Decryption for Voting from Lattice-Based Assumptions*, Diego F. Aranha, Carsten Baum Kristian Gjøsteen, and Tjerand Silde, accepted at ACM CCS 2023, eprint.iacr.org/2022/422
- *Concrete NTRU Security and Advances in Practical Lattice-Based Electronic Voting*, Patrick Hough, Caroline Sandsbråten, and Tjerand Silde, available at IACR ePrint 2023/993, eprint.iacr.org/2023/933

Thank you! Questions?

Slides are available at `https://tjerandsilde.no/talks`