NTNU

Norwegian University of Science and Technology

# Lattice-Based Verifiable Mix-Net

NTNU Applied Cryptology Lab, March 11, 2020

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

**Department of Mathematical Sciences, NTNU Trondheim**

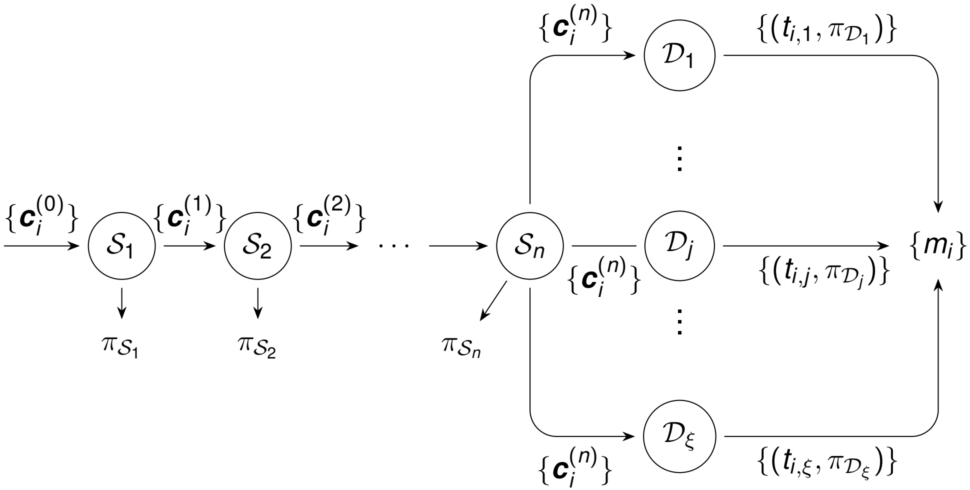
# Goal I

We want to create a mixing network where...

- each element being shuffled is a ciphertext,
- each ciphertext can publicly be re-randomized,
- we can prove the correctness of each shuffle,
- we can prove the correctness of each re-randomization,
- we can decrypt the ciphertexts in a distributed manner,
- we can prove the correctness of the decryption,
- and everything is post-quantum secure using lattices.

NTNU

# Goal II



$\{\boldsymbol{c}_i^{(0)}\}$
$\mathcal{S}_1$
$\{\boldsymbol{c}_i^{(1)}\}$
$\mathcal{S}_2$
$\{\boldsymbol{c}_i^{(2)}\}$
$\cdots$
$\mathcal{S}_n$
$\pi_{\mathcal{S}_1}$
$\pi_{\mathcal{S}_2}$
$\pi_{\mathcal{S}_n}$
$\{\boldsymbol{c}_i^{(n)}\}$
$\mathcal{D}_1$
$\{(t_{i,1}, \pi_{\mathcal{D}_1})\}$
$\mathcal{D}_j$
$\{\boldsymbol{c}_i^{(n)}\}$
$\{(t_{i,j}, \pi_{\mathcal{D}_j})\}$
$\{m_i\}$
$\{\boldsymbol{c}_i^{(n)}\}$
$\mathcal{D}_\xi$
$\{(t_{i,\xi}, \pi_{\mathcal{D}_\xi})\}$

# BGV Encryption Scheme I

The BGV encryption scheme consists of three algorithms: key generation ($\texttt{KeyGen}$), encryption ($\texttt{Enc}$) and decryption ($\texttt{Dec}$), where

- $\texttt{KeyGen}$, samples $a \xleftarrow{\$} R_q$, $s \leftarrow R_q$ such that $\|s\|_\infty = 1$, $e \leftarrow \mathcal{N}_\sigma(R_q)$, and outputs $\texttt{pk} = (a, b) = (a, as + e)$ and $\texttt{sk} = s$.
- $\texttt{Enc}$, on input $m$ in $R_p$, samples $r \leftarrow R_q$ such that $\|r\|_\infty = 1$, $e_1, e_2 \leftarrow \mathcal{N}_\sigma(R_q)$, and outputs the ciphertext $(u, v) = (ar + pe_1, br + pe_2 + m)$.
- $\texttt{Dec}$, on input $(u, v)$, outputs $m' \equiv v - su \mod q \mod p$.

# BGV Encryption Scheme II

Also, we have an algorithm `Rand` for re-randomization of the ciphertexts, where

- `Rand`, on input a ciphertext $(u, v)$ in $R_q^2$, samples $r' \leftarrow R_q$ such that $\|r'\|_\infty = 1$, $e'_1, e'_2 \leftarrow \mathcal{N}_\sigma(R_q)$, and outputs $(u', v') = (u + ar' + pe'_1, br' + pe'_2)$.

# BGV Encryption Scheme III

Further, we have an algorithm `DistDec` for distributed decryption of the ciphertexts, where each decryption server $D_j$, for $1 \leq j \leq \xi$ does the following:

- `DistDec`, on input a secret key-share $s_j$, computes $m_{i,j} = s_j u_i$, samples large $e_{i,j} \leftarrow R_q$ such that $\|e_{i,j}\|_\infty \leq 2^{\texttt{sec}}(B/p\xi)$, then outputs $t_{i,j} = m_{i,j} + pe_{i,j}$.

Then we obtain the full decryption of the ciphertext $(u_i, v_i)$ as

$$m_i \equiv v_i - t_i \mod p, \text{ where } t_i = t_{i,1} + t_{i,2} + ... + t_{i,\xi}.$$

# BGV Encryption Scheme IV

Finally, we have a method to switch the modulus of a ciphertexts, going from a ring $R_q$ to a ring $R_Q$, for two odd moduli $q$ and $Q$, while still being able to decrypt the original message using the original secret key $s$.

Let $(u', v') \leftarrow \texttt{Scale}((u, v), q, Q, p)$, where $\texttt{Scale}((u, v), q, Q, p)$ outputs the pair $(u', v')$ closest to $((Q/q)u, (Q/q)v)$ such that $u' \equiv u \mod p$ and $v' \equiv v \mod p$. Then $(u', v')$ is an encryption of $m$ under the key $s$ for modulus $Q$.

# Zero-Knowledge Proofs I

**Zero-knowledge proof of linearity**: Let $[\boldsymbol{m}_1]$, $[\boldsymbol{m}_2]$ and $[\boldsymbol{m}_3]$ be such that $m_{3,j} = \alpha_{1,j} m_{1,j} + \alpha_{2,j} m_{2,j}$ for public $\alpha_{1,j}, \alpha_{2,j} \in R_q$. We denote by

$$\pi_L \leftarrow \Pi_{\mathrm{Lin}}(([\boldsymbol{m}_1], [\boldsymbol{m}_2]), [\boldsymbol{m}_3], (\alpha_{1,1}, \alpha_{1,2}, \alpha_{2,1}, \alpha_{2,2})), \text{ and}$$

$$0 \vee 1 \leftarrow \Pi_{\mathrm{LinV}}(([\boldsymbol{m}_1], [\boldsymbol{m}_2]), [\boldsymbol{m}_3], (\alpha_{1,1}, \alpha_{1,2}, \alpha_{2,1}, \alpha_{2,2}), \pi_L),$$

the proof and verification protocols of this linear relation, respectively.

# Zero-Knowledge Proofs II

**Amortized zero-knowledge proof of short preimages**: Let $\boldsymbol{A}'$ be a publicly known matrix over $R_q$ and let $\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_l$ be short vectors over $R_q$. We compute $\boldsymbol{A}'\boldsymbol{s}_1 = \boldsymbol{t}_1, \boldsymbol{A}'\boldsymbol{s}_2 = \boldsymbol{t}_2, ..., \boldsymbol{A}'\boldsymbol{s}_l = \boldsymbol{t}_l$, and publish the set $\{\boldsymbol{t}_i\}_{i=1}^{l}$. We denote by

$$\pi_A \leftarrow \Pi_{\text{AZKPoK}}(\boldsymbol{A}', \boldsymbol{S}), \text{ and}$$

$$0 \vee 1 \leftarrow \Pi_{\text{AZKPoKV}}(\boldsymbol{A}', \boldsymbol{T}, \pi_A),$$

the proof and verification protocols of the knowledge of short $\boldsymbol{S}$, respectively.

# Zero-Knowledge Proofs III

**Zero-knowledge proof of shuffle of known content**: Given a list of commitments $[\boldsymbol{m}_1], [\boldsymbol{m}_2], ..., [\boldsymbol{m}_\tau]$ and a list of elements $\hat{\boldsymbol{m}}_1, \hat{\boldsymbol{m}}_2, \ldots, \hat{\boldsymbol{m}}_\tau$, we want to prove that the elements are the underlying messages of the commitments for some secret permutation $\gamma$ of the indices. We denote by

$$\pi_S \leftarrow \Pi_{\text{Shuffle}}(\{\boldsymbol{m}_i\}, \{[\boldsymbol{m}_i]\}, \{\hat{\boldsymbol{m}}_i\}, \gamma), \text{ and}$$

$$0 \vee 1 \leftarrow \Pi_{\text{ShuffleV}}(\{[\boldsymbol{m}_i]\}, \{\hat{\boldsymbol{m}}_i\}, \pi_S)$$

the run of the proof and verification protocols of the shuffle, respectively.

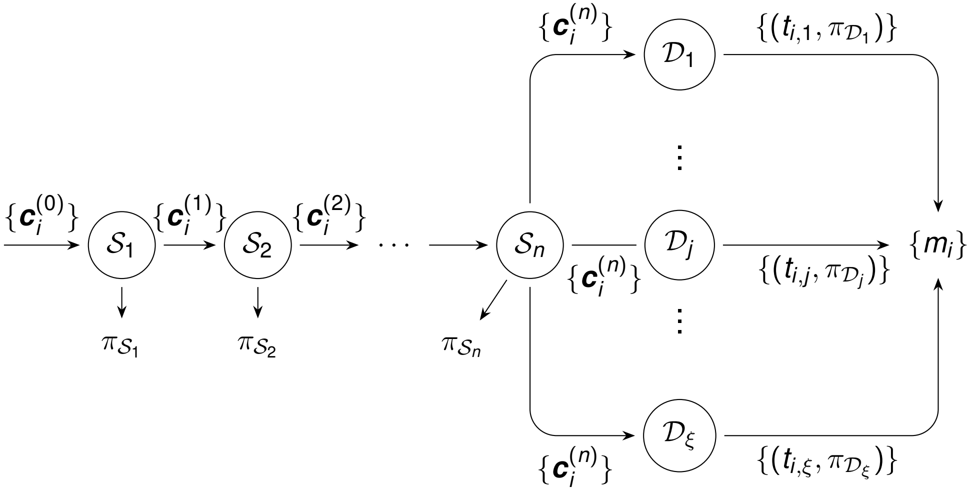
# The shuffle servers

1. receive a set of ciphertexts,
2. randomize the ciphertexts,
3. commits to the new ciphertexts,
4. prove correctness of the commitments,
5. shuffle the new ciphertexts,
6. prove correctness of the shuffle,
7. outputs information.

# The decryption servers

1. receive a set of ciphertexts,
2. switch the ciphertext-modulus,
3. partially decrypt the ciphertexts,
4. prove correctness the partial decryption,
5. prove correct norm of the randomness,
6. outputs information.

# Our protocol



NTNU

# *Thank You! Questions?*

Email: tjerand.silde@ntnu.no

Slides: `www.tjerandsilde.no/talks`