NTNU | Norwegian University of Science and Technology

Verifiable Mix-Nets and Distributed Decryption for Voting from Lattice-Based Assumptions

Diego F. Aranha (AU), Carsten Baum (DTU / AU),
Kristian Gjøsteen (NTNU), **Tjerand Silde (NTNU)**

Overview

Mixing Networks

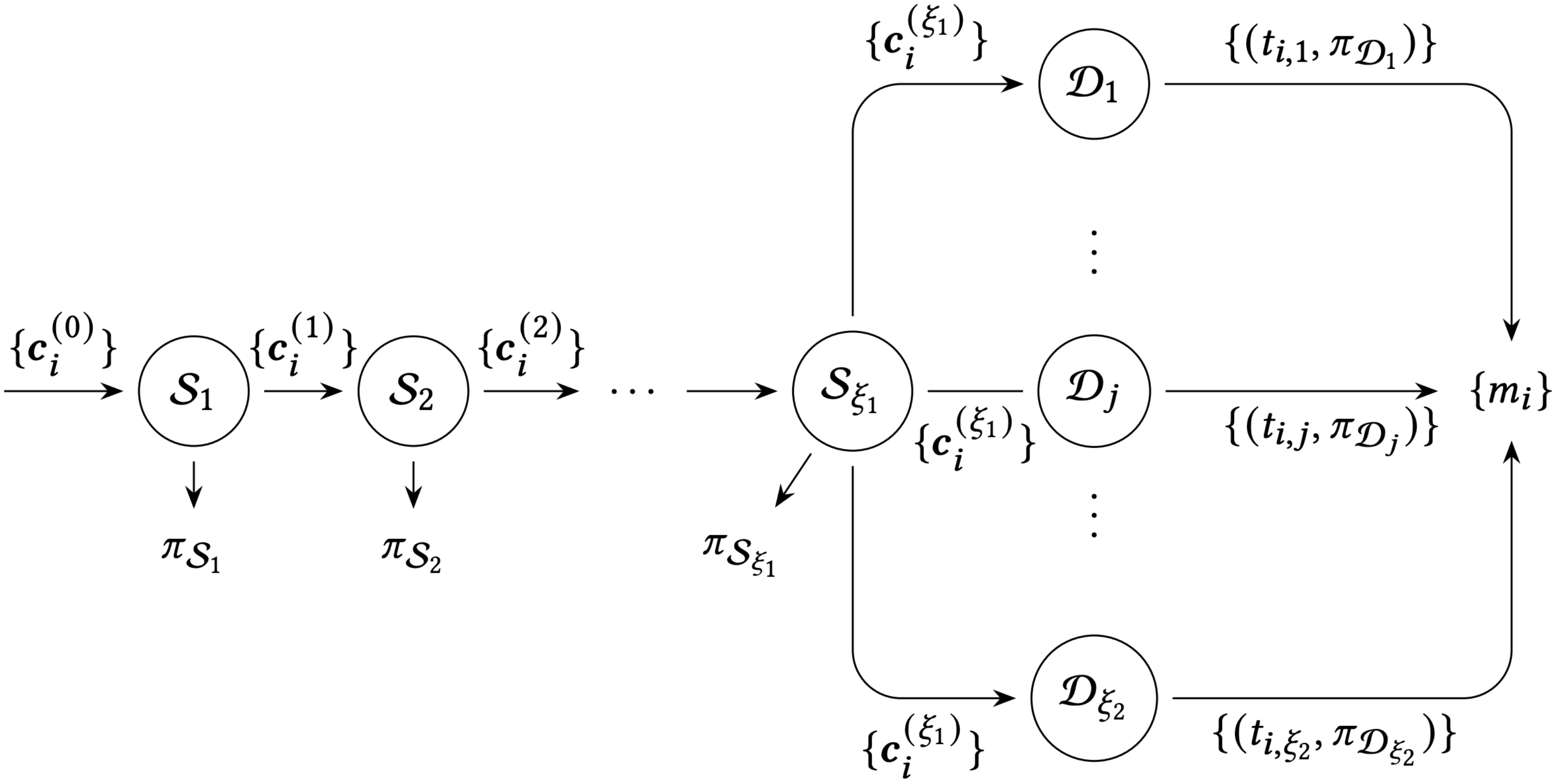
Distributed Decryption

Performance

Overview

1. Clients submit their votes as signed ciphertexts
2. Ciphertexts are re-encrypted and then shuffled
3. Ciphertexts are decrypted in a distributed way
4. Partial decryptions are combined into the votes

Overview



$\{c_i^{(0)}\}$
$\mathcal{S}_1$
$\{c_i^{(1)}\}$
$\mathcal{S}_2$
$\{c_i^{(2)}\}$
$\cdots$
$\mathcal{S}_{\xi_1}$
$\pi_{\mathcal{S}_1}$
$\pi_{\mathcal{S}_2}$
$\pi_{\mathcal{S}_{\xi_1}}$
$\{c_i^{(\xi_1)}\}$
$\mathcal{D}_1$
$\mathcal{D}_j$
$\mathcal{D}_{\xi_2}$
$\{c_i^{(\xi_1)}\}$
$\{c_i^{(\xi_1)}\}$
$\{(t_{i,1}, \pi_{\mathcal{D}_1})\}$
$\{(t_{i,j}, \pi_{\mathcal{D}_j})\}$
$\{(t_{i,\xi_2}, \pi_{\mathcal{D}_{\xi_2}})\}$
$\{m_i\}$

Overview

1. Ciphertexts are based on the LWE assumption
2. Commitments are based on LWE and SIS
3. Everything is of the form $\boldsymbol{A} \cdot \boldsymbol{s} = \boldsymbol{t}$ for short $\boldsymbol{s}$
4. We need four different zero-knowledge proofs

Mixing Networks

1. The server receives a vector of ciphertexts $\{ \boldsymbol{c}_i \}$
2. Creates a vector of encryptions of zero $\{ \underline{\boldsymbol{c}}_i \}$
3. Commits to zero-encryptions as $\underline{\boldsymbol{C}}_i = Com (\underline{\boldsymbol{c}}_i)$
4. Sums each $\bar{\boldsymbol{c}}_i = \boldsymbol{c}_i + \underline{\boldsymbol{c}}_i$, output permuted $\{ \bar{\boldsymbol{c}}_{\pi(i)} \}$

Mixing Networks

We need to prove the following in zero-knowledge:

1. $\{\, \underline{\boldsymbol{C}}_i \,\}$ are commitments to encryptions of zero
   - Need to prove many equations $\boldsymbol{A} \cdot \boldsymbol{s}_i = \boldsymbol{t}_i$ for short $\boldsymbol{s}_i$
2. $\{\, \underline{\boldsymbol{C}}_i + \boldsymbol{c}_i \,\}$ commits to the permuted set $\{\, \bar{\boldsymbol{c}}_{\pi(i)} \,\}$
   - Need to give a proof of shuffle for a set of vectors

Distributed Decryption

1. The servers receive a vector of ciphertexts $\{ \boldsymbol{c}_i \}$
2. Each server holds a uniform secret key-share $\boldsymbol{s}_i$
3. Samples large but bounded noise values $\boldsymbol{E}_i$
4. Finally outputs partial decryptions $\boldsymbol{t}_i = \boldsymbol{c}_i \cdot \boldsymbol{s}_i + \boldsymbol{E}_i$

Distributed Decryption

We need to prove the following in zero-knowledge:

1. The norm of noise $\boldsymbol{E_i}$ is bounded by a bound B
   - Different from the shortness proof for a larger bound
2. Decryptions $\boldsymbol{t_i}$ are computed as given linear eq.
   - We have efficient proofs of committed linear relations

Performance

$\boldsymbol{c}_i^{(k)}$	$[\![R_q^{l_c}]\!]$	$\pi_{\textsc{Shuf}}$	$\pi_{L_{i,j}}$	$\pi_{\textsc{Small}}$	$\pi_{\textsc{Bnd}}$	$\pi_{\mathcal{S}_i}$	$\pi_{\mathcal{D}_j}$
80 KB	$40(l_c + 1)$ KB	150τ KB	35 KB	20τ KB	2τ KB	370τ KB	157τ KB

Table 2: Size of the ciphertexts, commitments, and proofs.

Performance

Protocol	$\Pi_{\text{LIN}} + \Pi_{\text{LINV}}$	$\Pi^{l_c}_{\text{SHUF}} + \Pi^{l_c}_{\text{SHUFV}}$
Time	$(43.4 + 6.4)\tau$ ms	$(44.9 + 7.9)\tau$ ms

Protocol	$\Pi_{\text{BND}} + \Pi_{\text{BNDV}}$	$\Pi_{\text{SMALL}} + \Pi_{\text{SMALLV}}$
Time	$(92.7 + 23.9)\tau$ ms	$(214.4 + 10.0)\tau$ ms

Table 4: Timings for cryptographic protocols, obtained by computing the average of 100 executions with $\tau = 1000$.

Conclusions

We present the first lattice-based voting scheme based on the shuffle-and-decrypt paradigm.

We give parameters, sizes, and timings, improving the performance compared to other building blocks.

The full paper is available at https://ia.cr/2022/422.

THANK YOU! QUESTIONS?