RSAConference2021

May 17 – 20 | Virtual Experience

SESSION ID: **CRYP-T10A**

Lattice-Based Proof of Shuffle and Applications to Electronic Voting

Tjerand Silde

Ph.D. Student
Math @ NTNU
@TjerandSilde
tjerandsilde.no

NTNU
Norwegian University of Science and Technology

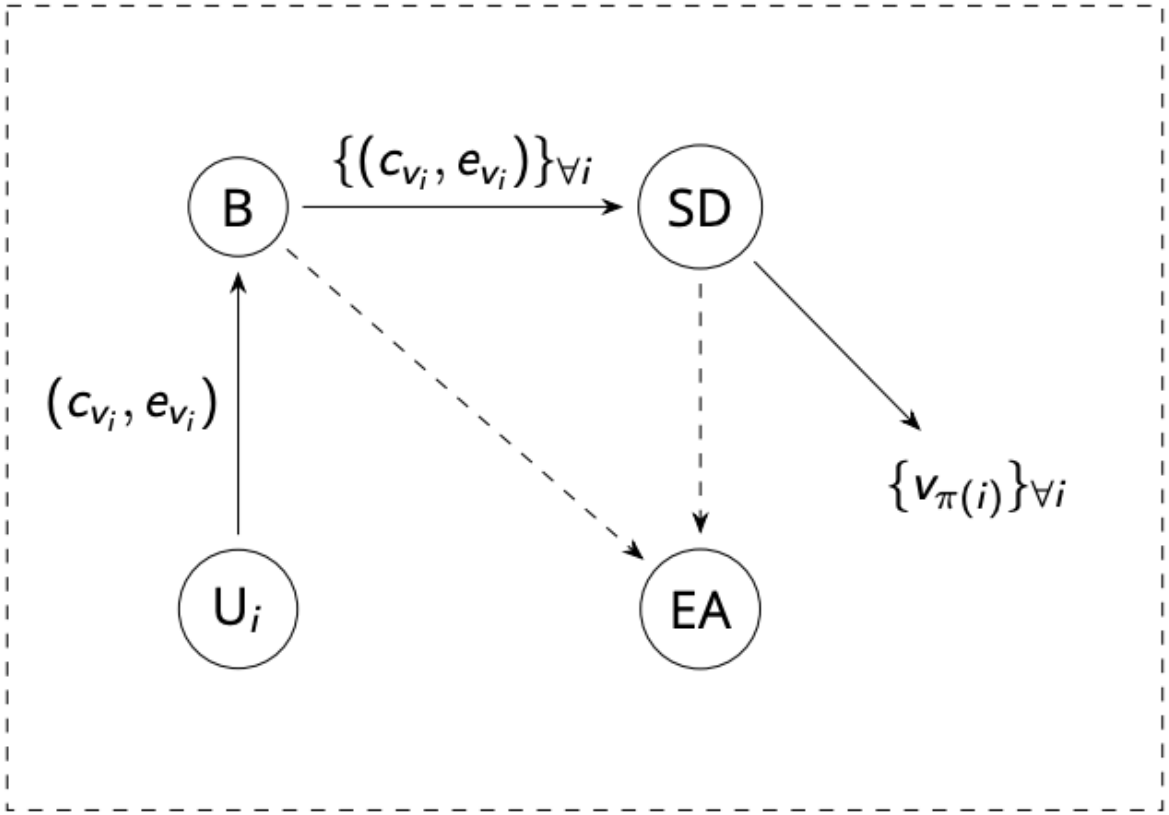
RSAConference2021

Joint work with:

Diego F. Aranha and Carsten Baum, Aarhus University
Kristian Gjøsteen and Thor Tunge, NTNU

Outline

- Overview
- Primitives
- Shuffle Protocol
- Voting System
- Performance



NTNU

Overview

- We present the first practical verifiable shuffle of known values based on lattice-based primitives to ensure long-term privacy
- We construct an electronic voting protocol by combining the verifiable shuffle with lattice-based verifiable encryption
- We also construct a return code mechanism for voter verifiability
- Finally, we implement the protocol and present performance
- Future work: prove the construction secure in QROM

Primitives

- Our constructions are based on the following main primitives:
  - Lattice-based commitments and ZK-proofs of linear relations
  - Lattice-based verifiable encryption
- We use the commitment scheme by Baum et al. from SCN 2018
- The same commitment scheme provides very efficient ZK-proofs
- We adapt the verifiable encryption scheme by Lyubashevsky and Neven from EC 2017 to encrypt openings of the commitments

Shuffle Protocol

- Goal: given a set of messages and a set of commitments, we want to prove that there exists a secret permutation such that the commitments opens to a re-ordering of the set of messages
  - Set of messages $\{ m_i \}$
  - Set of commitments $\{ c_i \}$
  - Permutation π
- Relation $R_{Shuffle} = (\pi, \{ (m_i , r_i) \} : \forall i \text{ Open} (c_{\pi(i)} , m_i , r_i) = 1)$

Shuffle Protocol

- Idea by Neff from CCS 2001 used to create a verifiable shuffle: polynomials are stable under permutation of their roots
- Schwartz-Zippel Lemma: two different polynomials differs with overwhelming probability when evaluated in a random point
- Our protocol: commit to many random linear combinations of commitments and messages to create two large polynomials, evaluate them randomly, and prove in ZK that they are equal

Voting System

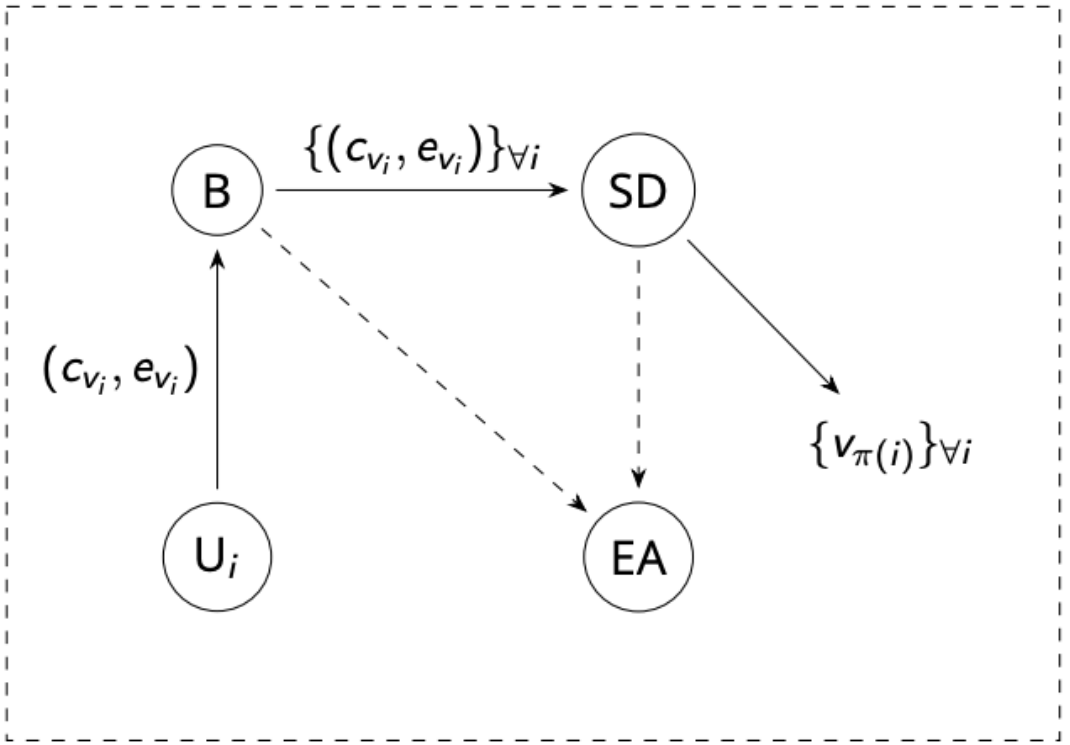
- Players: Users, Ballot Box, Shuffle Server, Election Authorities.
- Users: Commit and encrypt ballot. Prove correctness. Send to BB.
- BB: Receive votes, check proofs, strip information. Send to SS.
- SS: Receive votes, decrypt, shuffle, publish ballots and a proof.
- EA: Ensure that everything went well and the proofs are correct.
- We also have a return code mechanism so that the voter receives a confirmation that the correct vote is submitted.

Voting System

Security

- Integrity of the system follows from the zero-knowledge proofs
- Privacy of votes if ballot box and shuffle server does not collude
- Voter verifiability follows from the return code mechanism

System



Performance

- We are working over the cyclotomic ring $R_q = Z_q[X] / < X^N + 1 >$
- Instantiation: N = 1024 and q is a 32-bit prime $\equiv$ 1 modulo 4
- The shuffle proof consist of one commitment, one ring element and one zero-knowledge proof of linear relation per message
- Decryption + shuffle takes 33 ms and has size 17 KB per vote
- Our voting system is 5 times faster and at least 50 % smaller per vote than the 0/1 voting system by del Pino et al. from CSS 2017

RSAConference2021

Thank you! Questions?

Email: tjerand.silde@ntnu.no

Full version: eprint.iacr.org/2021/338