Complex Elections via Threshold (Fully) Homomorphic Encryption

Charlotte Bonte, Georgio Nicolas, Nigel Smart

ZAMA

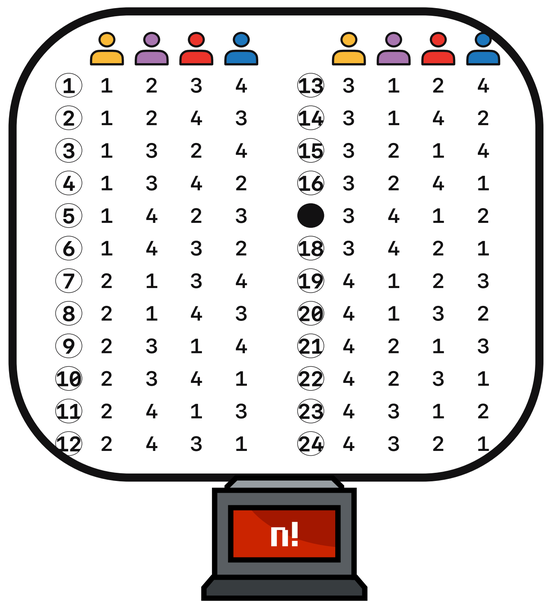
KU LEUVEN

COSIC

E-Voting Phases

1. Setup
   - › Configure Server
   - › Distribute Keys
2. Voting
   - › Encode and Encrypt Ballots
   - › Generate zkProofs
3. Decoding
   - › Verify zkProofs
   - › Decode Ballots
4. Tallying
   - › Compute Homomorphic Tally
   - › Decrypt Result
   - › Eliminate Candidates *: if needed

Complex Elections (AV/RCV)



RCV ballot sizes grow at a factorial rate relative to the number of choices to rank.

# Choices	Size	Voting Time
n = 4	270 kb	7s
n = 5	1.1 mb	40s
n = 6	4.6 mb	273s

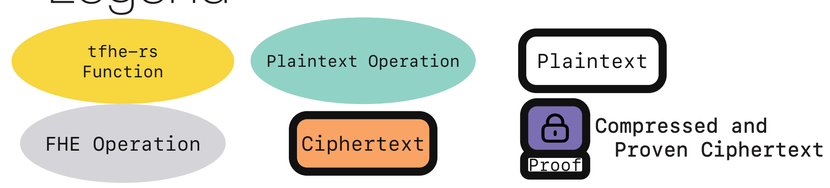
Client-side performance for RCV with Helios

[BMN+09] estimate 10,000h of tallying for an election with 5 candidates and 3 million voters.

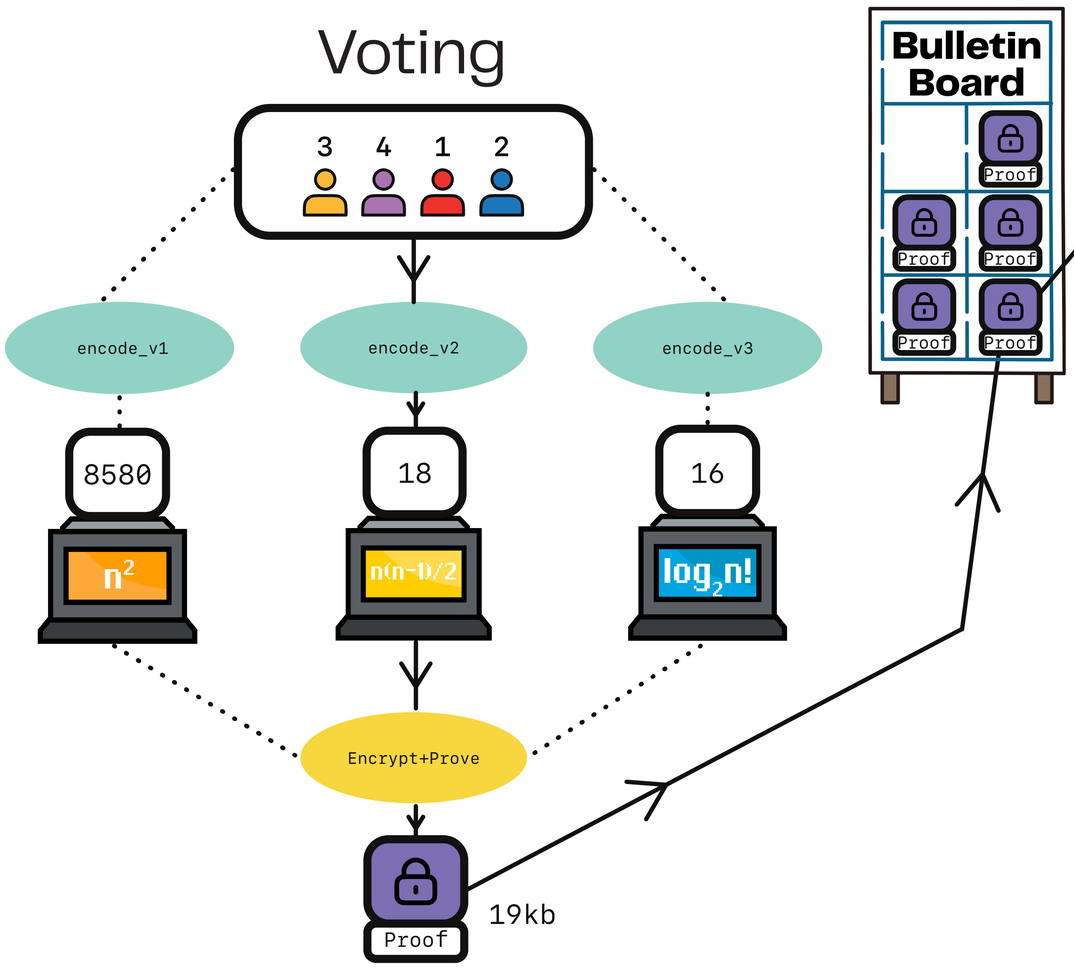
Our Scheme

- › 120x Faster than [BMN+09].
- › Minimal Leakage (Homomorphic Tallying).
- › HNDL Post-Quantum Secure.
- › 3 encoding methods to prioritize speed, proof size, or decoding complexity.
- › Constant client-side performance (Up to 10 choices, 19kb Ballot, 170ms Voting Time).
- › Implemented with tfhe-rs.

Legend

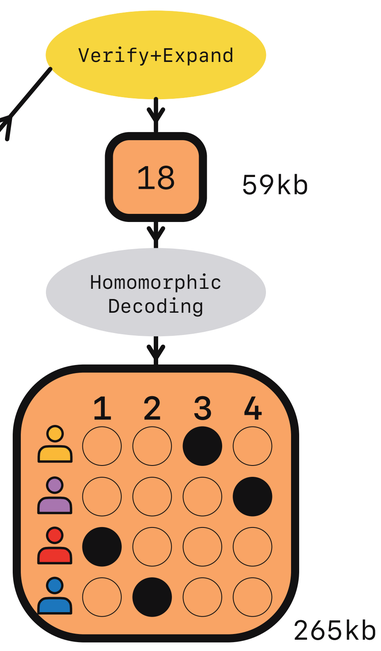


Voting

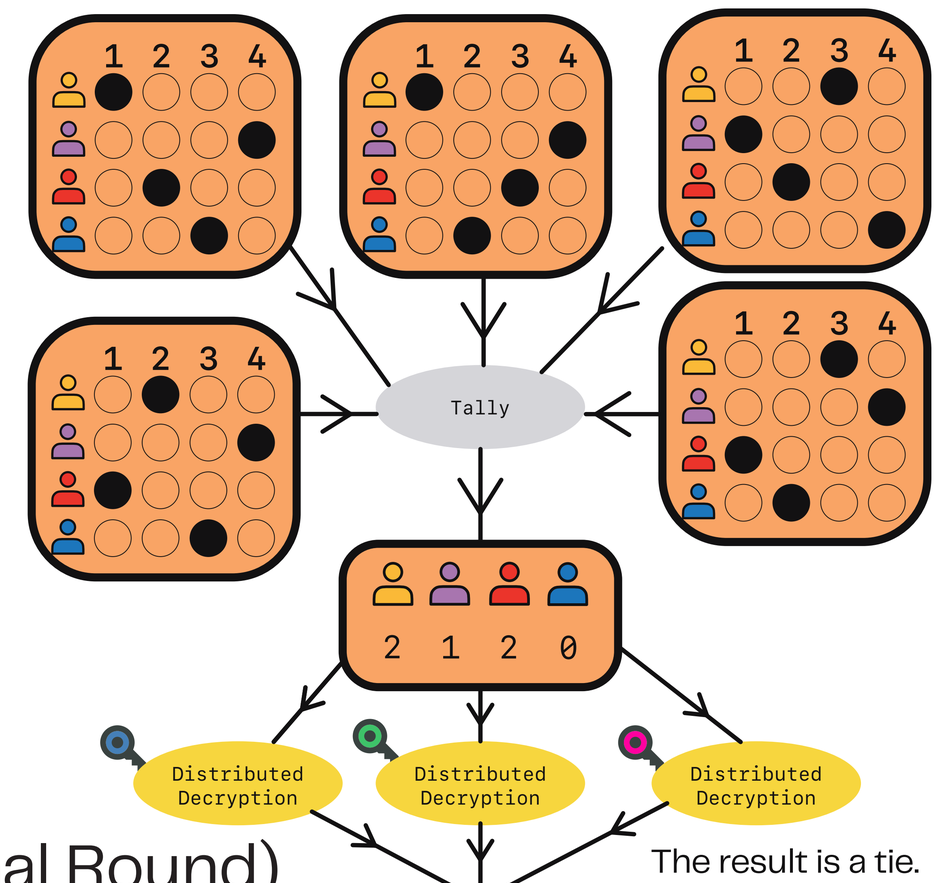


Bulletin Board

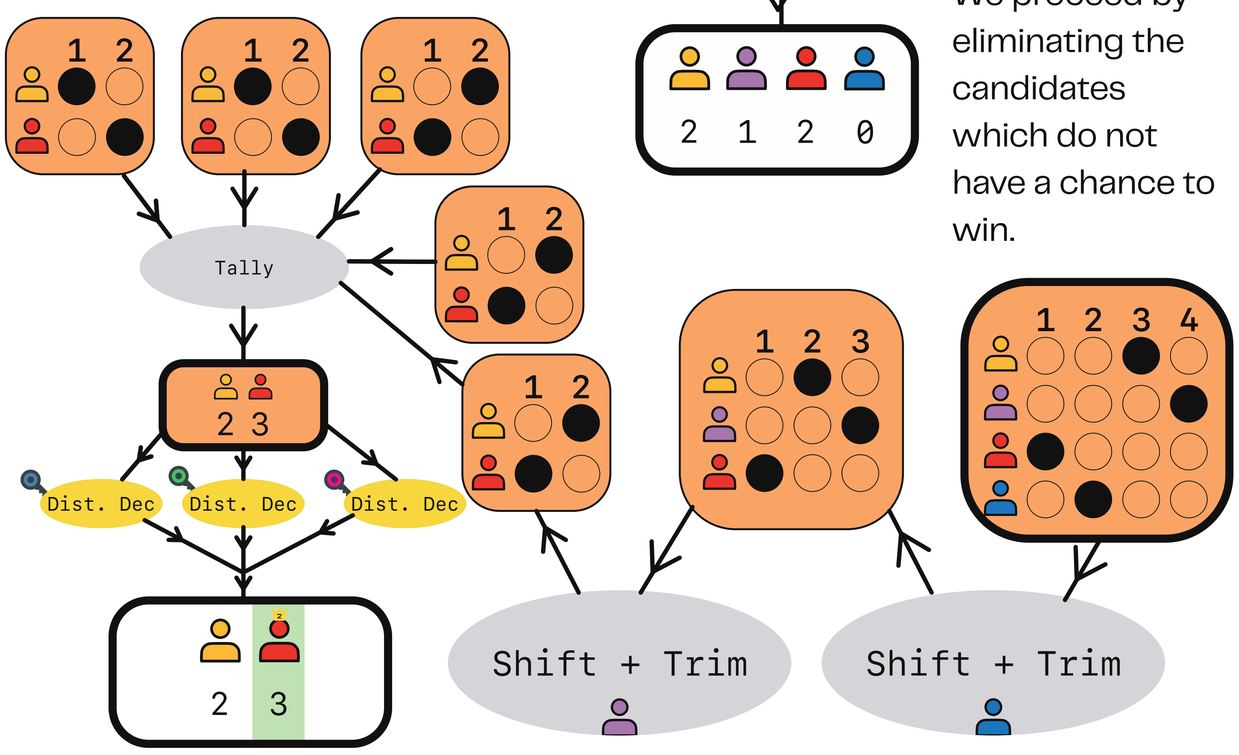
Decoding



Tallying (Round 1)

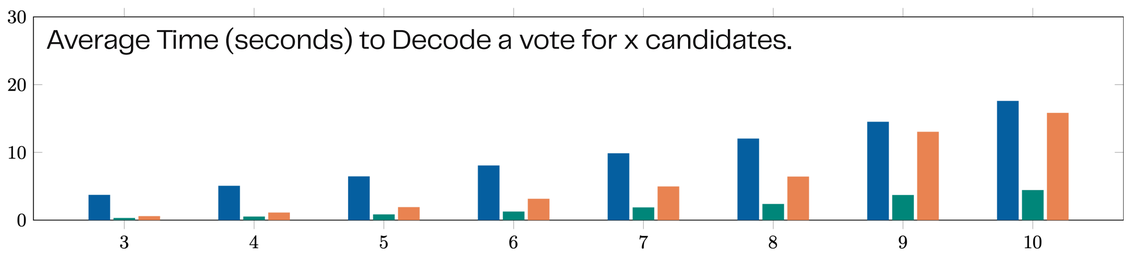


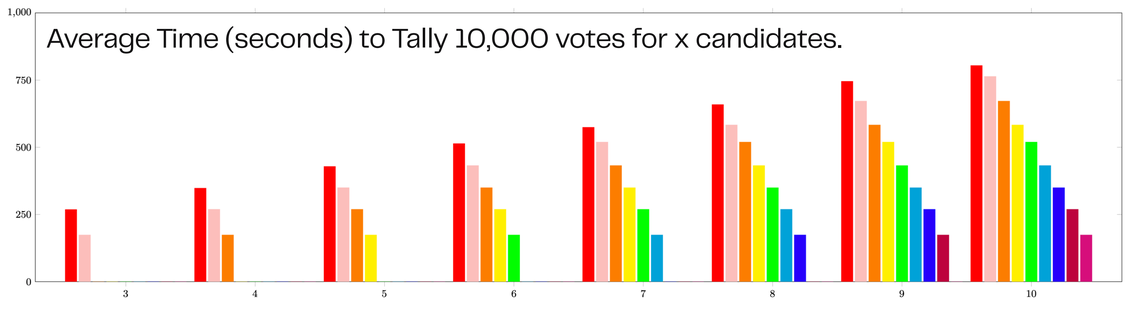
Tallying (Final Round)



The result is a tie. We proceed by eliminating the candidates which do not have a chance to win.

Benchmarks





Resources

ia.cr/2025/1482