

From ElGamal to Lattice-Based

Enhancing Helios Voting with Quantum-Safe Cryptography

João Pedro Cardoso Barbosa Gustavo Zambonin
Thaís Bardini Idalino Ricardo Felipe Custódio

Universidade Federal de Santa Catarina, Florianópolis, Brazil
`joao.c.barbosa@posgrad.ufsc.br`

ACNS, June 25th

Context

- ▶ **E-voting Challenge:** Balancing *verifiability* with *privacy*
 - ▶ homomorphism, mix-nets, blind signatures
- ▶ **Vulnerability of Classical Systems**
 - ▶ Mostly reliant on discrete logarithm or factorization problems broken by Shor's algorithm [Sho99]
- ▶ **Future and imminent threat**
 - ▶ Store now, decrypt later

Our Motivation

- ▶ **Abundance of Theory:** Many theoretical proposals for quantum-safe voting schemes exist
 - ▶ [ABG⁺21, ABGS23, BHM21, CGGI16, dPLNS17]
- ▶ **Lack of Practical Application:**
 - ▶ Few full-scale implementations of these schemes
 - ▶ Little integration into real-world systems
- ▶ **Limitation:** This gap hinders practical adoption and the evolution of e-voting

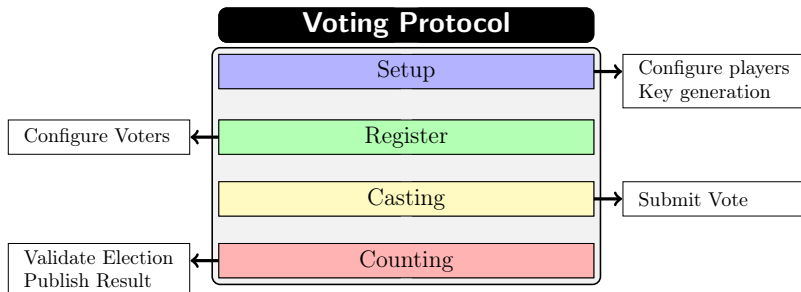
Our Approach

- ▶ **Foundation:** Implementing the lattice-based e-voting protocol by Aranha et al. [ABG⁺21]
 - ▶ Bridge the gap between theory and practice
- ▶ **Why Aranha et al. [ABG⁺21]?**
 - ▶ Considered the closest to a practical instantiation of a quantum-safe e-voting scheme
 - ▶ Supports complex ballot structures

Our Contributions

- ▶ **First Publicly Available Implementation:**
fully-featured implementation of a quantum-safe e-voting protocol
- ▶ **Instantiation and Formalization of Π_{Sum} :** formal definition, proof sketch, and implementation for the Proof for Sum over [BDL⁺18] commitments.
- ▶ **Integration with Helios Voting**

Electronic Voting



Cryptographic Building Blocks

- ▶ **Foundation:** polynomial rings $R = \mathbb{Z}[X]/\langle X^N + 1 \rangle$ and R_p for a $p \in \mathbb{N}$
- ▶ **Key Primitives:**
 - ▶ Commitment Scheme (C) ([BDL⁺18])
 - ▶ Verifiable Encryption (VE) ([ABG⁺21] adapted from [LN17])
 - ▶ Return Codes & PRF
 - ▶ Proof of Shuffle (Π^{Shuffle}) ([ABG⁺21])
 - ▶ Proof for Sum (Π^+) ([BDL⁺18])

Commitment Scheme C

The voter commits to the value of the vote

- ▶ $\text{KEYGEN}_C, \text{COM}, \text{OPEN}, \text{REC}$
- ▶ Committing to a value secretly, with later opening
- ▶ *hiding* and *binding*



$$\llbracket m \rrbracket \leftarrow \text{COM}(m, r_m)$$



$$\text{OPEN}(m, r_m, \llbracket m \rrbracket) \stackrel{?}{=} 1$$



$$m \leftarrow \text{REC}(\llbracket m \rrbracket, r_m)$$

Verifiable Encryption VE

Binds a commitment and an opening to the vote

- ▶ $\text{KEYGEN}_{\text{VE}}, \text{ENC}, \text{DEC}, \text{VER}$
- ▶ Ciphertext includes a proof of knowledge that asserts a property of the message from public information



Interplay: C + VE

► Commit to a vote

$$\llbracket v \rrbracket \leftarrow \mathbf{COM}(v, \mathbf{r}_v)$$

► Encrypt openings

$$e \leftarrow \mathbf{ENC}(\mathbf{r}_v, pk)$$

► Prove that an opening was used to create a commitment

$$\mathbf{VER}(e, pk, \llbracket v \rrbracket) \stackrel{?}{=} 1$$

► Decipher the opening

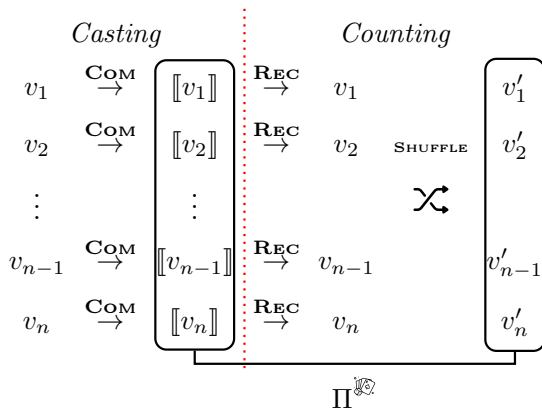
$$\mathbf{r}_v \leftarrow \mathbf{DEC}(e, sk)$$

► Recover the vote

$$v \leftarrow \mathbf{REC}(\llbracket v \rrbracket, \mathbf{r}_v)$$

Proof of Shuffle

Verify the election result



- Π proves that $\llbracket v_1 \rrbracket, \dots, \llbracket v_n \rrbracket$ are valid commitments for $v'_1, \dots, v'_n$ without revealing the permutation

Proof for Sum

Binds the return code to the vote

- ▶ Let $m, m', m'', \alpha, \beta \in R_p$
- ▶ **Purpose:** To prove a linear relationship between committed values ($m'' = \alpha m + \beta m'$) without revealing m, m', m''
- ▶ **Significance:** verify that the return code is derived from the actual vote
- ▶ **Structure:** three-move interactive protocol made non-interactive using Fiat-Shamir with aborts

Proof for Sum

Binds the return code to the vote

Lemma (Simplified)

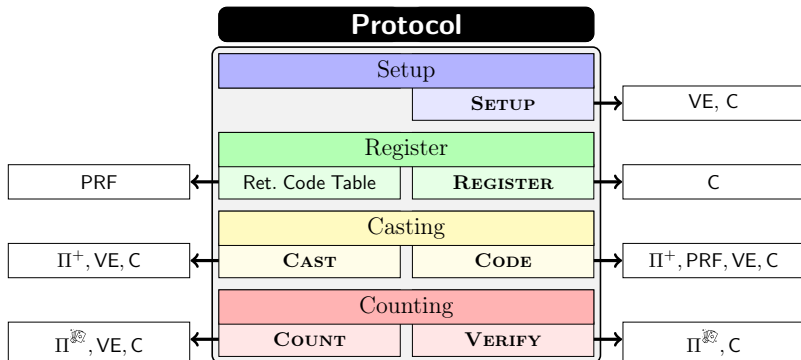
Let $\alpha, \beta \in R_p$ and $\llbracket m \rrbracket, \llbracket m' \rrbracket, \llbracket m'' \rrbracket$ be three commitments with respective messages m, m', m'' and openings $\mathbf{r}, \mathbf{r}', \mathbf{r}''$. Then, Π^+ is an Honest-Verifier Zero Knowledge Proof of the relation

$$R_{\text{Sum}} = \left\{ (q, w) \left| \begin{array}{l} q = (\alpha, \beta, \llbracket m \rrbracket, \llbracket m' \rrbracket, \llbracket m'' \rrbracket), \\ w = (m, \mathbf{r}, m', \mathbf{r}', m'', \mathbf{r}''), \\ \text{OPEN}(\llbracket m \rrbracket, m, \mathbf{r}) \\ \quad = \text{OPEN}(\llbracket m' \rrbracket, m', \mathbf{r}') \\ \quad = \text{OPEN}(\llbracket m'' \rrbracket, \alpha \cdot m + \beta \cdot m', \mathbf{r}'') \\ \quad = 1 \end{array} \right. \right\}.$$

The Quantum-Safe Voting Protocol: Participants

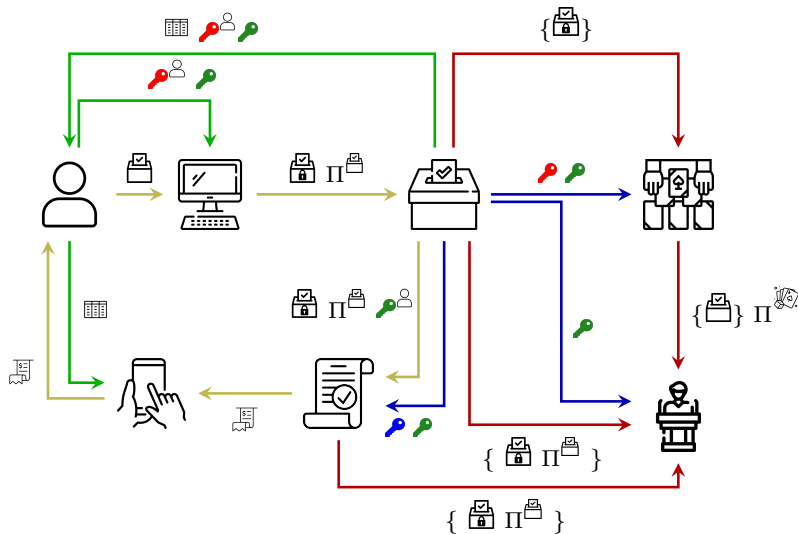
- ▶ **Voters** (): Use voting device () and secondary device () for return codes
- ▶ **Ballot Box** (): Collects encrypted ballots and proofs
- ▶ **Return Code Generator** (): Generates and sends return codes
- ▶ **Shuffler** (): Anonymizes votes via shuffle
- ▶ **Auditors** (): Verify election integrity

The Quantum-Safe Voting Protocol: Phases



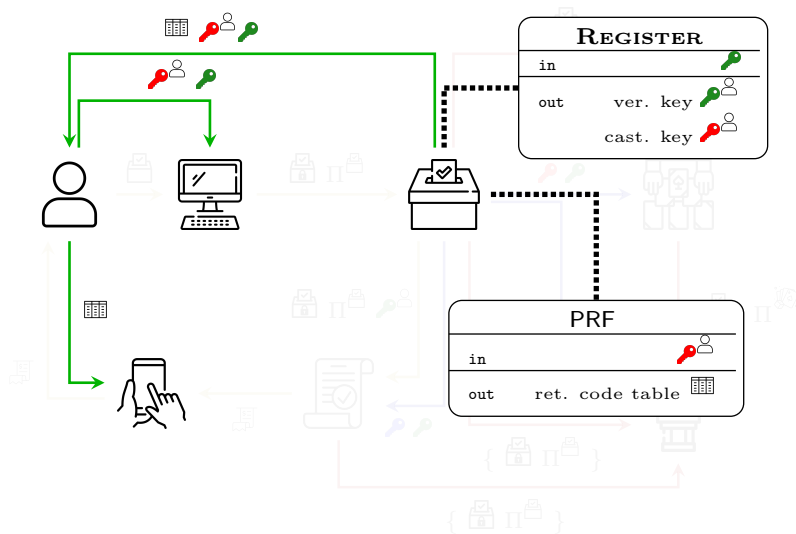
The Quantum-Safe Voting Protocol: Overview

□ Setup □ Registration □ Casting □ Counting



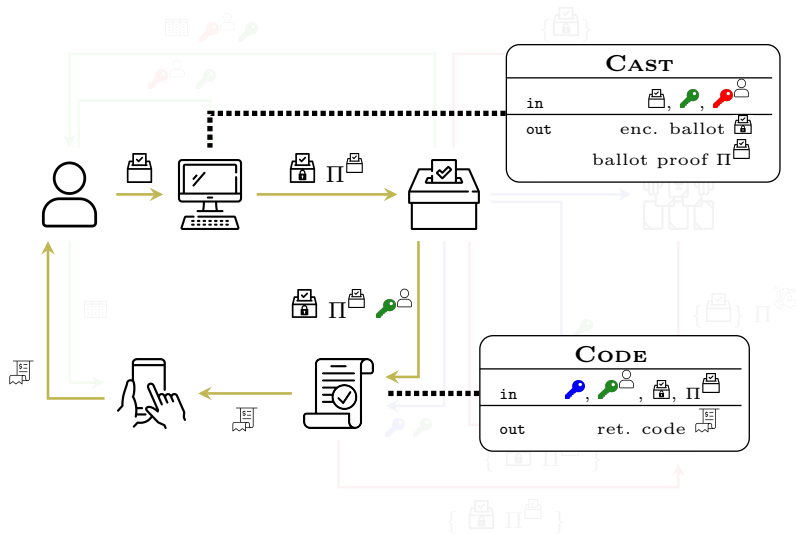
The Quantum-Safe Voting Protocol: Register

□ Setup ■ Registration □ Casting □ Counting



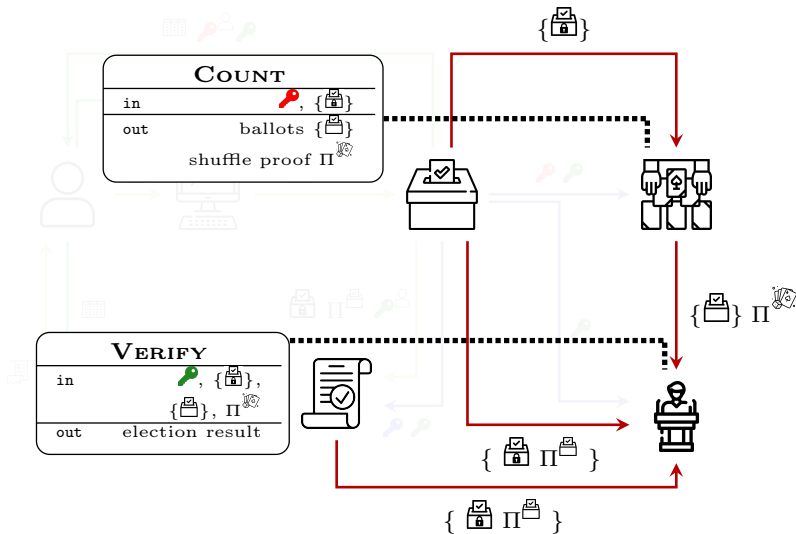
The Quantum-Safe Voting Protocol: Cast

□ Setup □ Registration ■ Casting □ Counting



The Quantum-Safe Voting Protocol: Count

Setup Registration Casting Counting



Implementation and Integration with Helios

- ▶ **Software Architecture**
 - ▶ **Python Library:** High-level protocol logic, phase orchestration
 - ▶ **C Primitives:** Core cryptographic operations
- ▶ **Helios Integration**
 - ▶ Main Helios server acts as the Ballot Box and Trusted entity for *Setup* and *Registration*
 - ▶ Other players as APIs

Properties gained and lost

- ▶ **Properties Gained:**
 - ▶ Quantum Security
 - ▶ Complex Ballots
 - ▶ Cast-as-intended Ballot Verification
- ▶ **Properties Lost and Challenges:**
 - ▶ Universal Verifiability
 - ▶ Browser Compatibility

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
Register	REGISTER	0.014	1.541
	PRF	-	5.926
Casting	CAST	304.600	116.506
	CODE	-	48.340
Counting	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Inexpensive for both (key generation), independent of voter count

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
	REGISTER	0.014	1.541
Register	PRF	-	5.926
	CAST	304.600	116.506
Casting	CODE	-	48.340
	COUNT	0.156	31.879
Counting	VERIFY	59*	5.581
Total		304.770	209.773

Ours is inexpensive (polynomial sampling). Helios registration is negligible.

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
Register	REGISTER	0.014	1.541
	PRF	-	5.926
Casting	CAST	304.600	116.506
	CODE	-	48.340
Counting	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Generation of the return code table. Significant overhead; scales greatly with voting options.

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup Register	SETUP	5*	4*
	REGISTER	0.014	1.541
	PRF	-	5.926
Casting	CAST	304.600	116.506
	CODE	-	48.340
Counting	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Ours is significantly faster than Helios, even using lattice-based primitives.

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
	REGISTER	0.014	1.541
Casting	PRF	-	5.926
	CAST	304.600	116.506
Counting	CODE	-	48.340
	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Additional step for cast-as-intended ballot verification. Verifies Π^+ and VE

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
Register	REGISTER	0.014	1.541
	PRF	-	5.926
Casting	CAST	304.600	116.506
	CODE	-	48.340
Counting	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Ours (shuffle-proof generation) is slower. Helios homomorphic tallying is efficient

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
Register	REGISTER	0.014	1.541
	PRF	-	5.926
Casting	CAST	304.600	116.506
	CODE	-	48.340
Counting	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Ours (shuffle-proof ver) scales with the number of voters.
Helios count proof only scales with the auditor count

Performance Analysis

Phase	Algorithm	Time (ms) per voter	
		Helios	Ours
Setup	SETUP	5*	4*
Register	REGISTER	0.014	1.541
	PRF	-	5.926
Casting	CAST	304.600	116.506
	CODE	-	48.340
Counting	COUNT	0.156	31.879
	VERIFY	59*	5.581
Total		304.770	209.773

Conclusion

- ▶ **Surprising Result:** certain portions of the protocol performed similarly or better than Helios
- ▶ **Viability:** a lattice-based quantum-safe e-voting scheme can be practically competitive
- ▶ **Implementation:** step towards practical, deployable quantum-resistant e-voting
 - ▶ Open-source and benchmarks aid future research and development

Future Works

- ▶ **Enhanced Usability**

- ▶ Achieve broader browser compatibility for seamless integration into the Helios web booth
- ▶ Generalize the return code mechanism to support sending more general message types to voters via secure channels

- ▶ **Explore Other Protocols**

- ▶ Evaluate other quantum-safe e-voting schemes to integrate into Helios



<https://github.com/jp-db/voting-protocol-acns>

From ElGamal to Lattice-Based

Enhancing Helios Voting with Quantum-Safe Cryptography

João Pedro Cardoso Barbosa Gustavo Zambonin
Thaís Bardini Idalino Ricardo Felipe Custódio

Universidade Federal de Santa Catarina, Florianópolis, Brazil
`joao.c.barbosa@posgrad.ufsc.br`

References I

- [ABG⁺21] Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, Tjerand Silde, and Thor Tunge. Lattice-Based Proof of Shuffle and Applications to Electronic Voting. In Kenneth G. Paterson, editor, *Topics in Cryptology – CT-RSA 2021*, volume 12704 of *Lecture Notes in Computer Science*, pages 227–251, May 2021.
- [ABGS23] Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, and Tjerand Silde. Verifiable Mix-Nets and Distributed Decryption for Voting from Lattice-Based Assumptions. In *CCS '23: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1467–1481, November 2023.
- [BDL⁺18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More Efficient Commitments from Structured Lattice Assumptions. In Dario Catalano and Roberto de Prisco, editors, *Security and Cryptography for Networks (SCN 2018)*, volume 11035 of *Lecture Notes in Computer Science*, pages 368–385, September 2018.
- [BHM21] Xavier Boyen, Thomas Haines, and Johannes Müller. Epoque: Practical End-to-End Verifiable Post-Quantum-Secure E-Voting. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 272–291, September 2021.

References II

- [CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. A Homomorphic LWE Based E-voting Scheme. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography (PQCrypto 2016)*, volume 9606 of *Lecture Notes in Computer Science*, pages 245–265, February 2016.
- [dPLNS17] Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. Practical Quantum-Safe Voting from Lattices. In *CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1565–1581, October 2017.
- [LN17] Vadim Lyubashevsky and Gregory Neven. One-Shot Verifiable Encryption from Lattices. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, volume 10210 of *Lecture Notes in Computer Science*, pages 293–323, April 2017.
- [Sho99] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review*, 41(2):303–332, April 1999.

The images have been designed using resources from Flaticon.com