# Unconditional Individual Verifiability with Receipt Freeness

#### WHY E-VOTING FAILS DEMOCRACY?

- Real-World Deployment Gap. Ontario 2022 proves the theory-practice gap—municipalities chose unverifiable systems over E2E-V alternatives, citing complexity.
- **Trust Dependencies.** Even verifiable systems require voters to trust authorities during verification, creating opportunities for deception.
- Threshold Decryption Limits Privacy. Threshold systems require known, liable parties, fundamentally limiting anonymity and creating corruption vulnerabilities.
- Quantum Threat. Cryptographically relevant quantum computers would break computational privacy assumptions. This perception already affects elections today: voters may fear expressing their true choices while authorities can use it to justify withholding election evidence.

All these problems trace back to an unresolved tension in e-voting system design.

#### THE E-VOTING DESIGN CHALLENGE

Every e-voting system must balance three competing requirements:

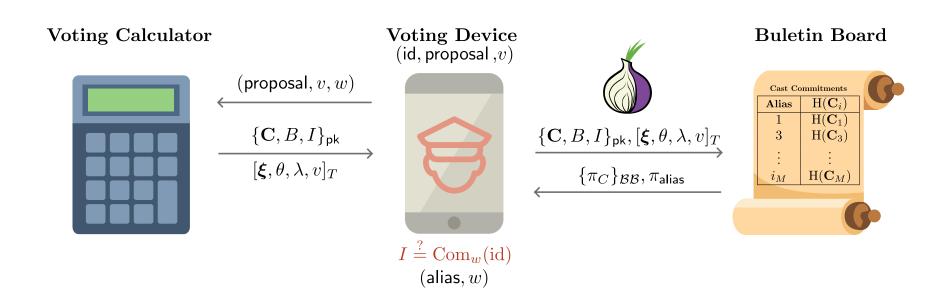
- o Individual Verifiability: Voters can confirm their votes were counted correctly
- Receipt Freeness: Voters cannot prove to others how they voted (prevents coercion/buying)
- Software Independence: No reliance on trusted authorities for verification guarantees

Core Challenge: Can an e-voting system satisfy all three properties simultaneously while offering public auditability and everlasting privacy?

#### Breakthrough: Post-Cast Isolation

The key innovation is **temporal separation**: voters isolate their calculators after voting until verification, breaking the receipt-freeness and verifiability paradox.

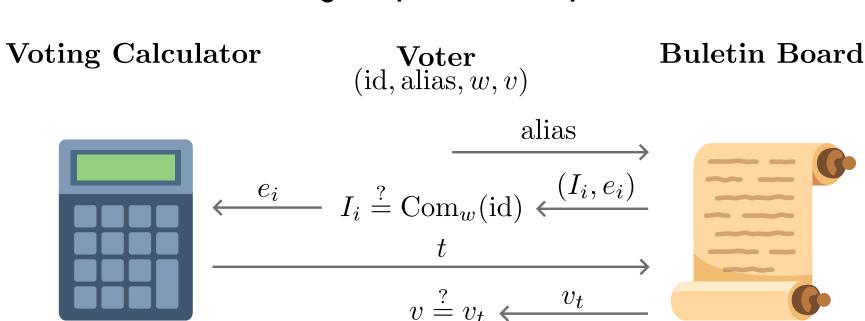
#### 1. Voting Phase - Calculator creates vote commitments



## 2. Isolation Phase - Physical separation from communication



## 3. Verification Phase - Challenge-response with tally board



The verification process can be streamlined through a voting device-generated link to the bulletin board with embedded parameters. Voters enter the link into a chosen browser, obtaining and confirming exclusive ownership of the challenge. They then input this challenge into their isolated voting calculator to compute the tracker, which can be submitted back to the web form to retrieve and verify the vote.

## RELATED WORK

## Pseudonym Braiding

Haenni & Spycher proposed a pseudonym anonymization scheme that constructs exponential mix via ElGamal reencryption shuffle. Voters sign their votes with braided pseudonyms, verified by:

$$\pi_{Braid} = \mathsf{PoK}\{(\chi, x) : \mathsf{g}_{i+1} = \mathsf{g}_i^x \wedge Y_{i+1, j} = Y_{i, \chi(j)}^x\}$$

## Centralized Deployment

Traditional immutable bulletin boards rely on third-party monitors to ensure integrity. PeaceFounder proposed an alternative using history trees, where voter devices maintain lightweight consistency proof chains ensuring votes are neither manipulated nor dropped, enabling secure centralized deployment.

## Tracker-Based Verification

Selene introduced tracker-based verification with post-voting verification phases.

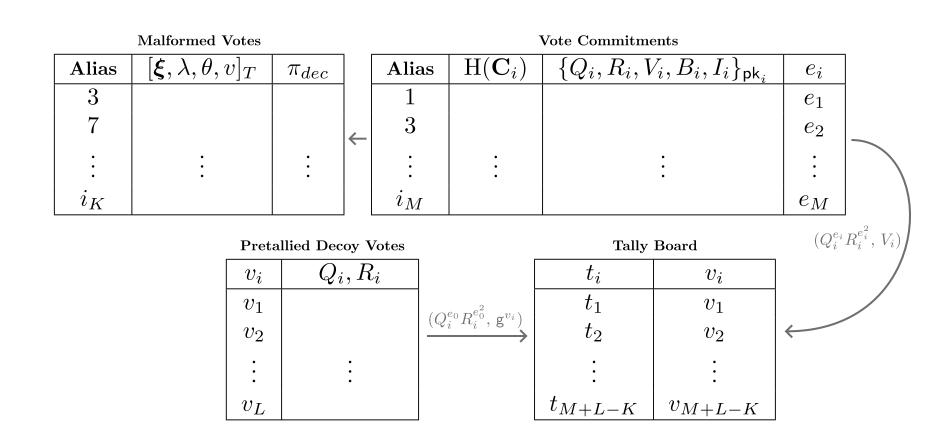
PostCast system differs through **trapdoorless tracker construction**—no secrets exist during voting that could fake verification.

## via Post-Cast Isolation

Dr. Janis Erdmanis

janis@peacefounder.org

#### CRYPTOGRAPHIC FRAMEWORK



#### Vote Construction:

The voting calculator generates a secret tracker preimage  $\theta,\lambda\in\mathbb{Z}_q$  that is signed together with vote commitment on voters choice  $v\in\mathbb{Z}_q$  using homomorphic commitment scheme:

$$V = \operatorname{Com}(v), \quad Q = \operatorname{Com}(\theta), \quad R = \operatorname{Com}(\lambda)$$

#### Challenge Generation:

Once all votes are collected tracker preimage commitments are used to construct a single seed from which tracker challenges  $e_i \in \mathbb{Z}_q$  are derived:

$$seed = Hash(\{Q_i, R_i\}_i)$$

$$\{e_i\}_i \leftarrow PRG(seed)$$

#### **Tracker Computation:**

Tracker challenges are assigned to individual voters, who can then compute their trackers using their calculators. Whereas tracker commitments are computed publicly using homomorphism:

$$T = Q^{e}R^{e^{2}} = \text{Com}(t)$$
$$t = e \theta + e^{2}\lambda \pmod{q}$$

Commitments  $\{T_i,V_i\}_i$  are jointly shuffled with proof:

$$\pi_{PoS} = \mathsf{PoK}\{\psi : T_i = \mathsf{Com}(t_{\psi(i)}) \land V_i = \mathsf{Com}(v_{\psi(i)})\}$$

where the tally board  $\{t_j,v_j\}_j$  is announced immediately.

The release of proofs and challenges can be delayed to enable coercion mitigation, allowing voters to find a tracker with the coercer's preferred choice and configure their voting calculator to use it as a decay.

## SECURITY PROPERTIES

Property	Mechanism	Key Requirement
Individual Verifiability	Information flow control	Observable isolation
Receipt Freeness	Decoy state simulation	Memory protection
Universal Verifiability	Proof of shuffle: $\pi_{PoS}$	Standard crypto assumptions
Vote Anonymity	Shuffle/Braiding	Honest tallier OR one braider
Everlasting Privacy	Pedersen commitments	Honest tallier

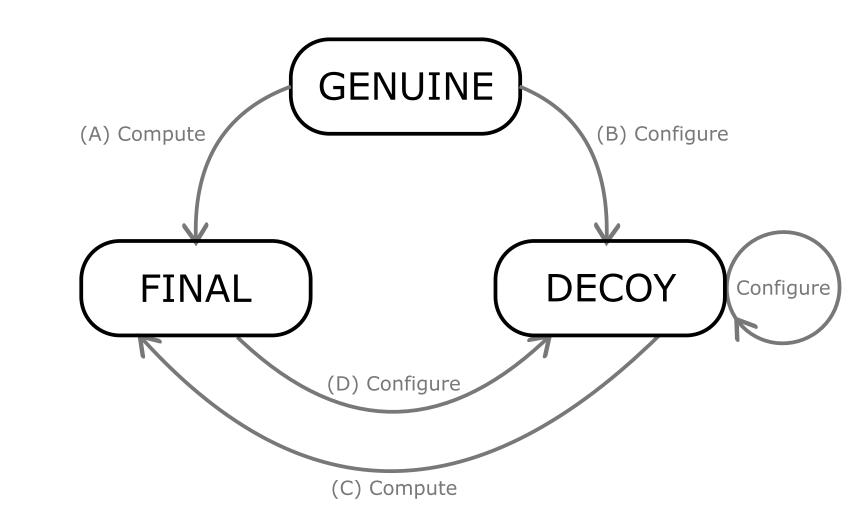
## TRACKER CONSTRUCTION SECURITY OVERVIEW

• Tracker Unpredictability: The tallier might attempt to force predictable tracker assignments by manipulating the last published tracker preimage commitments that enter the seed. This is computationally infeasable under discrete logarithm hardness assumption. Hence the calculator can only guess:

$$\Pr[\mathsf{guess}] = 1/q$$

- Challenge Exclusivity: The system uses unique challenges that are paired
  with voters pseudonyms. This means a corrupt calculator cannot compute
  valid trackers for other voters' votes without knowing their challenges.
- Tracker Extrapolation Protection: A dual-component tracker design and single attempt for computing genuine tracker prevents extrapolation from trackers computed with arbitrary challenges during voting to actual challenges after voting phase closes.
- Pedersen Commitments: Perfectly hiding Pedersen commitments, shuffled by the tallier before public opening, provide information-theoretic protection of the vote-voter relationship. This ensures vote privacy against computationally unbounded future adversaries, hence achieving everlasting privacy.

#### RECEIPT FREENESS THROUGH TRACKER OVERRIDE



The calculator can operate in genuine verification or decoy simulation modes, with the voter controlling which mode is active. The voter can choose to:

- (A) Enter their challenge to compute tracker, locate their vote, then configure decoy tracker via path (D)
- (B) Configure decoy tracker directly along with challenge hash, supporting coercion mitigation strategies as in Selene systems

To prevent  $(\theta,\lambda)$  extraction, the calculator is limited to computing only one tracker. A checksum can be added to the challenge to ensure that voters do not accidentally compromise their ability to verify their vote.

#### Unconditional Individual Verifiability

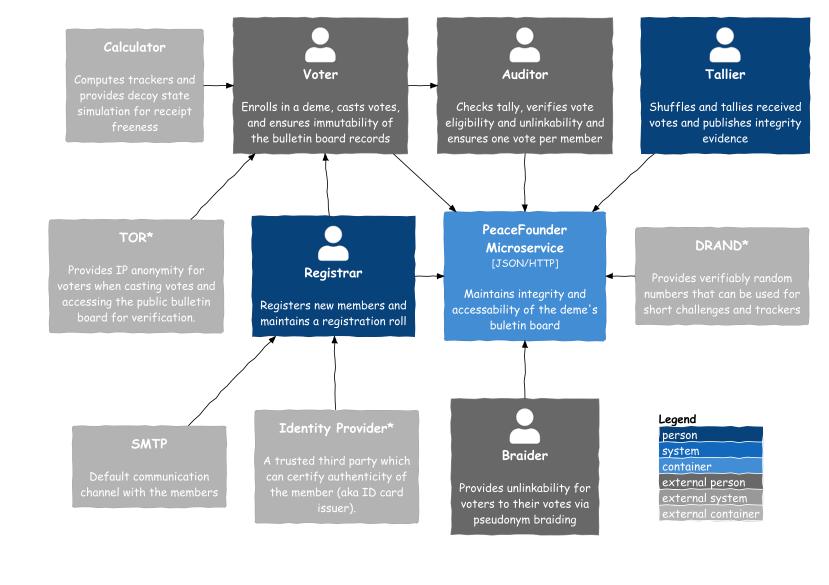
The isolation of the voting calculator creates information-theoretic guarantees that enables individual verifiability without trust dependencies:

- o Calculator is isolated from communication once voter casts their vote
- Calculator cannot access or predict what trackers/votes are on the tally board
- $\circ$  Calculator only learns the voter's exclusive challenge  $e_i$  and hence must compute  $t=e\,\theta+e^2\lambda$  or guess

#### How Can You Verify AND Deny Your Vote?

The tracker override is a local setting you control. You know whether you're looking at your real tracker or a fake one. A coercer cannot tell the difference since they have no control over the information flow in the calculator as long as it remains in your possession. This gives you both genuine verification and plausible deniability.

## DEPLOYMENT BENEFITS



- User-controlled security through isolation
- Market incentives for honest hardware
- O Weaker trust model reliability not integrity
- o **Public auditability** lets anyone to audit resulting tally at any time
- Clear accountability when verification fails

Simple centralized deployment with voter-verifiable results and accessible vote privacy

## FUTURE WORK

- Short trackers with verifiable randomness
- ElGamal mixing with threshold decryption
- Individual tally board views (Hyperion-style)
- Coercion resistance

## REFERENCES

- 1 Erdmanis, J. (2025). Unconditional Individual Verifiability with Receipt Freeness via Post-Cast Isolation. IACR ePrint Archive.
- 2 Goodman, N., Spycher-Krivonosova, I., Essex, A., Brunet, J. (2023). Verifiability experiences in Ontario's 2022 online elections.
- 3 Ryan, P.Y.A., Rønne, P.B., Iovino, V. (2016). Selene: Voting with Transparent Verifiability and Coercion-Mitigation 4 Haenni, R., Spycher, O. (2011). Secure internet voting on limited devices with anonymized DSA public keys.
- Wikström, D. (2005). A sender verifiable mix-net and a new proof of a shuffle.
   Erdmanis, J. (2023). PeaceFounder: Centralized E2E verifiable e-voting via pseudonym braiding and history
- trees. IACR ePrint Archive.

  7 PeaceFounder project website: http://peacefounder.org

janiserdmanis.org EVoteID 2025