Rigorous Digital Engineering for Secure Voting Systems

SUMMARY REPORT

At a Glance

Sponsor: Tusk Philanthropies

Team: Free & Fair

Project: An open source cryptographic voting protocol and complete assurance artifacts.

Purpose: Create a robust base for future end-to-end verifiable online voting systems with public auditability and ballot secrecy.

Approach: Rigorous digital engineering (RDE).

This is a summary of the complete article, which can be found here.

FREE & FAIR

https://freeandfair.us

Project Overview

Software failures in critical systems are frequent and consequential. In elections, failure is unacceptable: if we cannot trust the systems that count our votes, we cannot trust the results.

In 2024, Tusk Philanthropies asked Free & Fair to develop a secure cryptographic protocol for online voting that would withstand extreme scrutiny. The result is an open, rigorously engineered, end-to-end verifiable Internet voting protocol designed as a secure foundation upon which vendors, jurisdictions, and civic organizations can build complete systems.

A Robust Foundation

This project treats correctness, security, and verifiability as non-negotiable. It employs *rigorous digital engineering (RDE)*, which is a model-driven, proof-oriented methodology with precise models, traceable requirements, formal architecture, verification-centric design, and robust proofs.

What The Protocol Provides

End-to-end verifiable Internet voting (E2E-VIV)

- **Cast-as-intended:** Voters can verify encrypted ballots match their choices.
- **Recorded-as-cast:** Voters can verify their encrypted ballots were posted correctly to a public bulletin board.
- **Counted-as-recorded:** Anyone can verify that posted ballots were tallied correctly.

Software independence: Even if surrounding software is flawed or compromised, undetected outcome changes are prevented by cryptographic proofs and public auditability.

The foundation, not a full voting system: The protocol comprises the secure core that different implementers can build upon while preserving uniform, verifiable security.

Security Posture and Cryptography

Transparency by design: Security does not depend on secrecy of design or code. Protocol, models, and proofs are public; only keys and privacy-critical data in an election remain secret.

Threat model: Threats include compromised devices, network attackers, corrupt insiders, and nation-states. Assumptions are deliberately strong: the adversary is sophisticated, well resourced, and knows the system.

Assurance case: 47 security requirements; 100+ adversary capabilities and scenarios; explicit mitigations and evidence chains linking requirements, models, and proofs.

Cryptographic building blocks: Leverages established algorithms and emerging post-quantum options; threshold cryptography to split trust; mix networks to break links between voters and ballots; zero-knowledge proofs to verify correctness without revealing secrets.

Trust distribution and zero trust: No single component is privileged. Keys are split across trustees; all interactions are mutually authenticated and signed; sensitive operations run in an air-gapped environment.

Call to Action

Election technology vendors and jurisdictions: Build on the open protocol to deliver user-friendly, verifiable systems. Use the assurance artifacts to drive certification and public trust.

Security researchers and auditors: Review the models, proofs, and code; contribute improvements.

Policymakers and funders: Encourage procurement and certification processes that reward rigorous, evidence-backed engineering.

The cryptographic voting protocol and all development artifacts are open source. Explore here: https://github.com/FreeAndFair/VoteSecure

Methodology: Rigorous Digital Engineering (RDE)

RDE builds an unbroken chain from intent to implementation through precise, analyzable models and proofs.

Domain engineering: Establishes an unambiguous, shared vocabulary for the election domain (e.g., ballot, marked ballot, cast ballot; cast vs. spoil events; constraints like “a ballot cannot be both cast and spoiled”). Controlled Natural Language evolves into formal models that can be checked for consistency and edge cases early.

Requirements engineering: Transforms stakeholder, legal, and security needs into precise, structured, traceable, and analyzable specifications decomposed hierarchically, grounded in authoritative sources, and formalized where safety and security demand proof.

Product line engineering: Manages variability across different applications of the protocol (e.g., encryption choice, revoting policies, receipt types). Feature models and constraints are analyzed automatically to ensure only secure, realizable configurations.

Architecture: A formal, verifiable blueprint that defines components, interfaces, and flows with clear alignment to requirements and security properties. Issues are caught and fixed before implementation.

Design: Verification-centric and explicit. Components are decomposed to make verification tractable; interfaces are minimal and precise.

Development: Specification-driven coding. Types and interfaces come first; constructors and validators prevent invalid states; correct-by-construction code generation is used where possible; continuous static analysis and runtime checking enforce discipline.

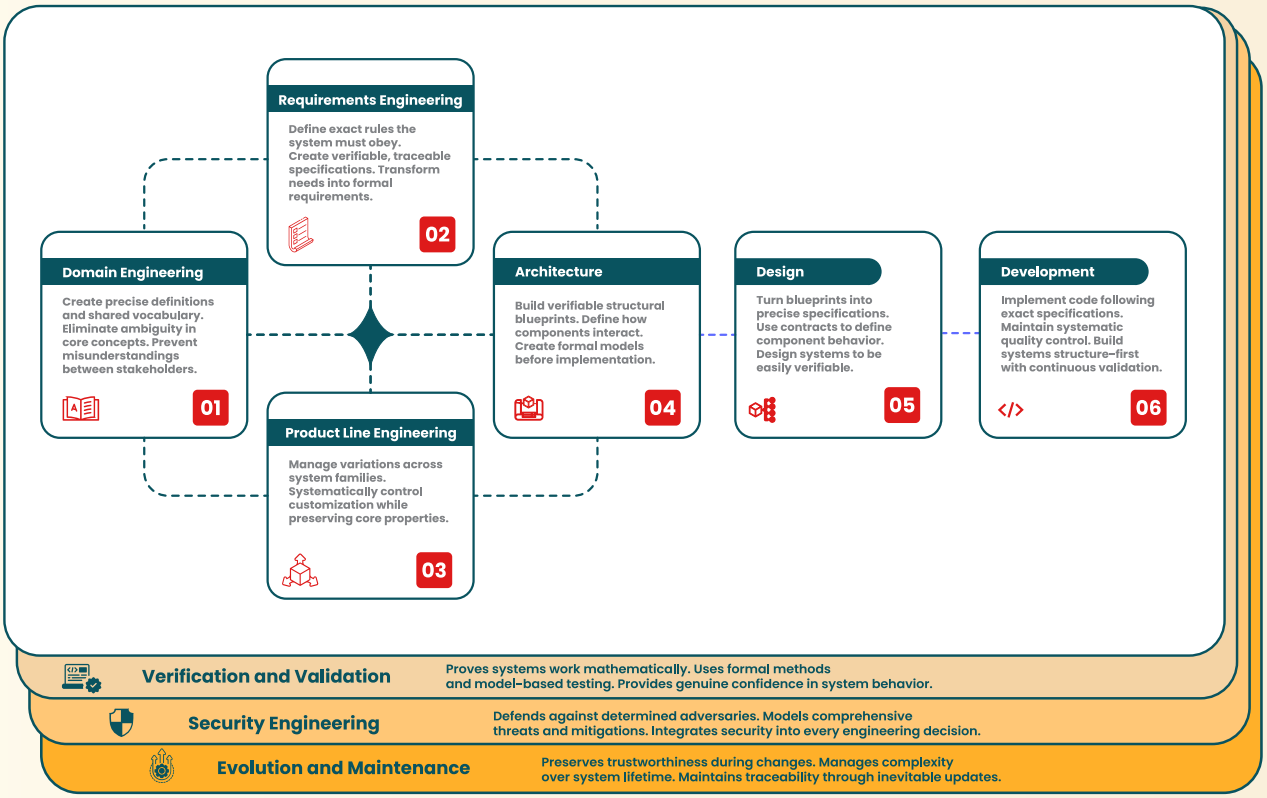
Security engineering: Integrated from day zero. Threat models, security requirements, mitigations, and proofs are first-class artifacts linked across the refinement chain. Zero trust and trust distribution are enforced by architecture and cryptography.

Verification and validation: Models generate tests; static analysis, model checking, and formal methods provide layered assurance. Continuous integration runs checks automatically; proofs and tests are tied to code changes to prevent regressions.

Evolution and maintenance: Compositionality and traceability provide precise change impact analysis. As requirements, platforms, or cryptographic libraries evolve, affected models, proofs, and code are identified and updated systematically, preserving trust over time.

Adoption Landscape for RDE

RDE's principles are moving from research to practice. DoD's Digital Engineering strategy and DARPA's resilience initiatives underscore a broad shift: we can no longer accept cyber risk as the status quo. The cost of rigor is typically far less than the cost of failure. Cultural change, skills development, and better tooling are the remaining barriers, and they are being addressed.



This is a summary of the complete article, which can be found here.