National University of Computer and Emerging Sciences
Islamabad Campus

Project Proposal
CY4001
Secure Software design

Submitted by: Javeriah Faheem , Sabreena Azhar , Ayesha Areej

Roll number: 22I-7421, 22I-1751, 22I-1711

Date: 13th September 2025

Table of Contents

Title: Secure E-Voting System

1. Team Information

Name	Roll No	Role
Javeriah Faheem	22i-7421	Research & Documentation
Sabreena Azhar	22i-1751	Development & Integration
Ayesha Areej	22i-1711	Testing & Evaluation

Collaboration approach:

We will follow an agile-inspired approach, holding weekly check-ins to review progress. Each member is primarily responsible for their assigned role but will peer-review each other's work to ensure code quality, adherence to secure coding principles, and consistency across the project.

2. Problem Statement

Modern voting systems face serious security challenges. Paper-based systems are slow, prone to human error, and susceptible to manipulation. Existing online voting systems often fail to provide confidentiality, integrity, and auditability simultaneously.

- From a secure software design perspective, the challenges include:
- Preserving the confidentiality of voter choices.
- Ensuring integrity so that votes cannot be altered or duplicated.
- Maintaining availability throughout the election period.
- Providing audit trails for transparency and accountability.

This project addresses these challenges by developing a Secure E-Voting System prototype, integrating end-to-end encryption, voter authentication, and audit logging into the Secure Software Development Lifecycle (S-SDLC).

3. Objectives

- Apply secure software design principles to build a privacy-preserving voting system.
- Implement end-to-end encryption for voting secrecy.
- Enforce voter authentication to prevent unauthorized or duplicate votes.
- Maintain integrity using digital signatures and secure logging.
- Introduce audit trails to ensure election results can be independently verified.
- Deliver a working prototype, documentation, and security analysis report.

4. Proposed Solution

System Overview:
The proposed system is a lightweight GUI-based e-voting application. Registered voters receive unique credentials. Votes are encrypted before submission, authenticated using tokens, stored securely, and logged in auditable trails.

Secure Software Principles Integrated:

- Confidentiality: End-to-end encryption of votes.
- Integrity: Digital signatures and tamper detection.
- Availability: Lightweight architecture with redundancy and error handling.
- Accountability: Secure audit trails and logging for verification.

Threats & Countermeasures:

Threat	Risk Level	Mitigation
Double voting / replay attack	High	Unique voter tokens, one-time authentication
Vote tampering	High	Digital signatures, hashing, audit logs
Unauthorized access	Medium	Strong voter authentication, input validation
Denial of Service (DoS)	Medium	Lightweight design, redundancy
Privacy leaks	High	Separation of identity from vote, blind signatures

5. Methodology

The project follows the Secure Software Development Lifecycle (S-SDLC):

- Requirements Analysis: Identify functional needs (voter registration, casting, tallying, audit). Define security goals using CIA triad and threat modeling (STRIDE).
- Design: Use security frameworks (NIST SSDF, principle of least privilege). Design modules for registration, encryption, authentication, and auditing.
- Implementation: Apply secure coding practices – input validation, no hard-coded secrets, exception handling, cryptographic best practices.
- Testing: Conduct unit tests, integration tests, and security testing (penetration tests, boundary testing, replay attack simulations).
- Deployment: Secure configuration, controlled environment, minimal privileges for execution.
- Maintenance: Patch vulnerabilities, update cryptographic libraries, review logs for anomalies.

6. Tools and Technologies

- Languages: Python
- Frameworks & Libraries: Tkinter (GUI), Cryptography libraries (RSA, blind signatures), UUID
- Development Tools: VS Code, GitHub for version control
- Security Tools: Digital signatures, encryption protocols, secure logging mechanisms

7. Expected Deliverables

- Prototype: Secure e-voting application with authentication, encryption, and audit logging.
- Documentation: Design diagrams, SDLC-based security mapping.
- Security Report: Threat model, risk analysis, test results.
- Presentation: Demonstration of prototype functionality and security principles applied.

8. Timeline

Phase	Duration	Deliverables
Requirement Gathering	Week 1	Requirement Specification
System Design	Week 2	Architecture Document
Prototype Development	Weeks 3–5	GUI + Core E-Voting Functions
Security Implementation	Week 6	Encryption, Authentication, Audit Trails
Testing & Validation	Week 7	Test Report
Documentation & Report	Week 8	Proposal Draft + Final Report
Presentation Preparation	Week 9	Slides & Prototype Demo

9. References

1. D. Chaum, “Blind signatures for untraceable payments,” *Advances in Cryptology*, 1983.
2. Wikipedia, “Blind Signature,” [Online]. Available: https://en.wikipedia.org/wiki/Blind_signature
3. Springer, “Blind Signatures,” in *Encyclopedia of Cryptography and Security*, 2021.
4. ACE Project, “Electronic Voting and Transparency,” [Online]. Available: https://aceproject.org/ace-en/topics/et/eth/eth02/eth02b
5. National Institute of Standards and Technology (NIST), *Secure Software Development Framework (SSDF)*, 2022.