BallotGuard: Face-Verified Blockchain-Inspired Secure Voting System

Project Synopsis Submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

Of

Bachelor of Technology

in

Computer and Communication Engineering

by

Shambhavi Sinha, K Liya, Sruthi D V

Reg. No. 230953060, Reg. No. 230953118, Reg. No. 230953142

Under the guidance of

Dr. Akshay KC (Lab Faculty 1)
Assistant Professor - Senior Scale
School of Computer Engineering
Manipal Institute of Technology
MAHE, Manipal, Karnataka, India

Dr. Snehal Samanth (Lab faculty 2)
Assistant Professor
School of Computer Engineering
Manipal Institute of Technology
MAHE, Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

INSPIRED BY LIFE

August 2025

Objective: To create a face-verified, privacy-preserving electronic voting system that guarantees voter authenticity, vote confidentiality, and tamper-proof auditability by incorporating homomorphic encryption, RSA digital signatures, blockchain-inspired integrity tracking, and machine learning-based facial recognition within a hybrid Tkinter desktop and Flask server architecture.

Scope:

- Voter Authentication: Face recognition to ensure only registered voters can vote.
- Vote Confidentiality: Use of Paillier homomorphic encryption to allow tallying without decrypting individual votes.
- Data Integrity: SHA-256 hashing with blockchain ledger to detect and prevent vote tampering.
- Auditability: Blockchain audit ensures verifiable and immutable voting records.
- Scalability: Capable of running multiple voting booths connected to a central server.

Need for the Application:

- Existing e-voting systems often face trust issues due to concerns about tampering, hacking, or unauthorized access.
- Many online voting platforms lack strong biometric verification, making them vulnerable to impersonation and identity fraud.
- Manual or paper-based voting is time-consuming, error-prone, and resource-intensive.
- Ensuring vote confidentiality while maintaining verifiable integrity is challenging in traditional systems.
- A face-verified, blockchain-audited, and encryption-protected electronic voting system can greatly enhance security, transparency, and public trust in elections.

Project Description

Problem Statement

Conventional e-voting systems face three key challenges:

- Authenticity – Ensuring the voter is genuine and authorized.
- Confidentiality – Keeping the vote private even from administrators.
- Integrity – Guaranteeing that no vote can be modified after being cast.

Proposed Solution

The proposed system addresses these challenges by:

- Using face recognition to authenticate voters before allowing them to vote.
- Applying RSA digital signatures to confirm that each vote was cast by a legitimate voter.
- Employing Paillier homomorphic encryption to store and tally votes without decrypting them.
- Storing SHA-256 hashes of each vote in a lightweight blockchain ledger for immutability and tamper detection.
- Implementing a Tkinter desktop application for voting booths with a Flask-based backend server for centralized, secure tallying.

Functionalities

- Voter Registration – Capture and store voter face data securely.
- Face Authentication – Verify voter before unlocking voting screen.
- Vote Casting – Candidate selection via Tkinter GUI.
- Cryptographic Security – Digital signature, homomorphic encryption, and hashing of votes.
- Blockchain Audit Trail – Append vote hash to blockchain ledger for tamper-proofing.
- Admin Panel – Verify signatures, tally votes using homomorphic addition, decrypt only the final total.

Hardware Requirements:

Client (Voting Booth)

- Processor: Intel i3 or above
- RAM: 4 GB minimum
- Storage: 10 GB
- Webcam: HD quality
- OS: Windows 10 or above

Server

- Processor: Intel i5 or above
- RAM: 8 GB minimum
- Storage: 50 GB
- OS: Linux/Windows Server

Software Requirements:

Client (Voting Booth Application)

- Python 3.x
- Tkinter (GUI)
- Face Recognition, OpenCV, dlib
- PyCryptodome (RSA), python-paillier (encryption)
- hashlib (hashing)
- sqlite3 (offline storage)
- PyInstaller (for packaging)

Server (Backend)

- Python 3.x
- Flask (REST API)
- PostgreSQL / MySQL
- PyCryptodome, python-paillier
- hashlib
- Custom Python blockchain implementation
- Gunicorn + Nginx (deployment)

Submitted by

Name	Registration number	Roll Number	Semester & Branch	Section
Shambhavi Sinha	230953060	6	V (CCE)	A
K Liya	230953118	12	V (CCE)	A
Sruthi D V	230953142	14	V (CCE)	A