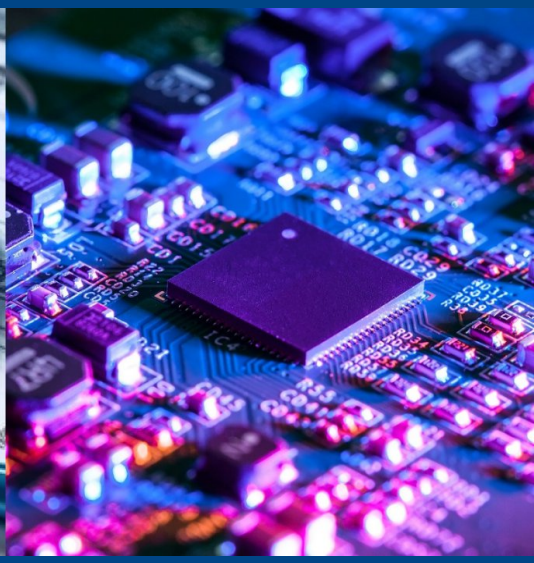
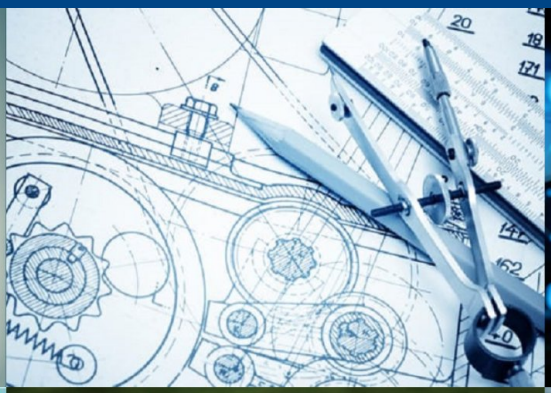
ISSN: 2582-7219

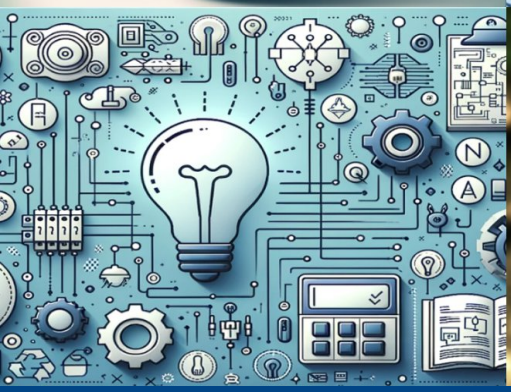




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 4, April 2026**

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904271

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Secure Digital eVoting System with Aadhaar-Based OTP Authentication and AES- 256 Encryption

Arunachalam A[1], Mohamed Aslam M[2], Mohamed Imthiyaz I[3], Mohamed Arshath I[4], Vanitha Sri[5]

Department of Computer Science and Engineering, Aalim Muhammed Salegh College of Engineering, IAF, Avadi, Chennai, Tamil Nadu, India[1]

Department of Computer Science and Engineering, Aalim Muhammed Salegh College of Engineering, IAF, Avadi, Chennai, Tamil Nadu, India[2]

Department of Computer Science and Engineering, Aalim Muhammed Salegh College of Engineering, Chennai, IAF, Avadi, Tamil Nadu, India[3]

Department of Computer Science and Engineering, Aalim Muhammed Salegh College of Engineering, IAF, Avadi, Chennai, Tamil Nadu, India[4]

Assistant Professor, Department of Computer Science & Engineering, Aalim Muhammed Salegh College of Engineering, IAF, Avadi, Chennai, Tamil Nadu, India[5]

ABSTRACT: This paper presents the design, implementation, and security analysis of a Secure Digital eVoting System built as a final-year undergraduate project. The system addresses key challenges in modern electoral processes including voter authentication, ballot integrity, geographic voting restrictions, and auditability. The proposed architecture employs Aadhaar- based voter identification with SMS One-Time Password (OTP) verification via Fast2SMS API, AES-256 CBC encryption for vote storage, BCrypt password hashing for OTP security in the database, and JSON Web Token (JWT)-based stateless session management. The system is implemented using a Spring Boot backend with MySQL persistence and a React/Vite frontend, and enforces state, city, and constituency-level voting restrictions for government elections. Experimental results demonstrate the system's robustness against common attack vectors including replay attacks, brute-force OTP guessing, and unauthorized API access. The system achieves complete voter anonymity while maintaining full auditability through encrypted vote receipts.

KEYWORDS: eVoting; Aadhaar Authentication; OTP; AES-256; BCrypt; JWT; Spring Boot; React; Election Security; Digital Democracy

I. INTRODUCTION

The integrity of democratic processes depends fundamentally on the security, accessibility, and transparency of the voting mechanism. Traditional paper-based voting systems suffer from several well- documented limitations including susceptibility to ballot tampering, logistical inefficiencies, high administrative costs, and limited accessibility for differently-abled or geographically remote voters [1].

Digital voting systems have emerged as a promising alternative, but introduce their own set of security challenges. A secure eVoting system must simultaneously guarantee voter anonymity, prevent double voting, ensure ballot integrity, and provide a verifiable audit trail—all while remaining accessible to a diverse user population [2].

This paper presents a comprehensive implementation of a Secure Digital eVoting System developed as a final-year project by undergraduate students at Aalim Muhammed College of Engineering. The system leverages modern web technologies and cryptographic primitives to address the aforementioned challenges.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904271

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The primary contributions of this work are:(1) a novel integration of India's Aadhaar identification system with SMS-OTP authentication for voter verification without reliance on a real UIDAI

API; (2) AES-256 CBC encryption for vote storage with per-vote initialization vectors; (3) BCrypt hashing of OTPs before database storage; (4) enforcement of geographic voting restrictions at state, city, and constituency levels; and (5) a complete full- stack implementation with role-based access control.

II. RELATED WORK

Numerous studies have investigated the feasibility and security of electronic voting systems. Chaum [3] proposed the use of mix-nets for anonymous voting, establishing foundational principles of ballot secrecy. Adida [4] introduced Helios, a web-based open-audit voting system that demonstrated the viability of verifiable online voting for low-coercion elections.

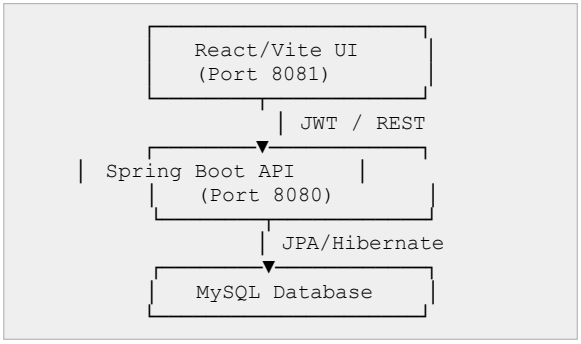
In the Indian context, the Electronic Voting Machine (EVM) has been the subject of extensive scrutiny [5]. Researchers have highlighted vulnerabilities in standalone EVM systems and proposed internet-based alternatives with stronger cryptographic guarantees. More recent work by Hapsara et al. [6] demonstrated the effectiveness of OTP-based voter authentication in mobile voting applications.

The use of blockchain for voting has gained significant attention [7], though scalability and usability concerns remain. Our approach differs by prioritizing practical deployability and integration with India's existing Aadhaar infrastructure, achieving strong security guarantees without the complexity of distributed ledger technology.

III. SYSTEM ARCHITECTURE

A. Overview

The system follows a three-tier architecture comprising a React/Vite frontend (Port 8081), a Spring Boot REST API backend (Port 8080), and a MySQL database. Communication between tiers is secured via HTTPS and JWT bearer tokens. Figure 1 illustrates the high-level architecture.



IV. SECURITY DESIGN

A. Voter Authentication

Authentication proceeds in two phases. In the first phase, the voter submits their 12-digit Aadhaar number, which is looked up against the local voter registry. Upon a successful match, the voter's registered mobile number is retrieved and a 6-digit OTP is generated using Java's SecureRandom CSPRNG.

Before storage, the OTP is hashed using BCrypt with a randomly generated salt (work factor 10). This ensures that even if the OTP log table is compromised, attackers cannot recover plaintext OTPs. The hashed OTP and an expiry timestamp (5 minutes) are persisted in the otp_logs table.

- 30-second cooldown between OTP resend requests

Single-use enforcement: status updated to

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904271

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In the second phase, the voter submits the received OTP, which is verified against the stored BCrypt hash using the BCryptPasswordEncoder.matches() method. The system enforces the following constraints:

- OTP expires after 5 minutes of generation
- Maximum 3 incorrect verification attempts before status is set to BLOCKED

VERIFIED after successful match

For development and testing, a bypass mechanism exists for designated test numbers (9000000000 and 8000000000) with a fixed OTP of 123456, with test OTPs stored as BCrypt hashes.

B. Vote Encryption

Votes are encrypted using AES-256 in CBC mode prior to database storage. The EncryptionService derives a 256-bit AES key from a passphrase using PBKDF2WithHmacSHA256 with 65,536 iterations and a fixed salt. For each vote, a fresh 16-byte Initialization Vector (IV) is generated using SecureRandom. The IV is prepended to the ciphertext before Base64 encoding, enabling correct decryption. The passphrase and salt are externalized to application.properties and excluded from version control.

C. JWT Session Management

Upon successful OTP verification, the backend issues a JWT signed with HMAC-SHA256 using a 256-bit secret key loaded from application.properties. The token payload contains the voter's mobile number and role (VOTER or ADMIN). Tokens expire after one hour, after which re- authentication is required. The JwtAuthFilter intercepts all protected requests, validates the token signature and expiry, and populates the Spring Security context with the authenticated principal and role authorities.

D. CORS and API Security

Cross-Origin Resource Sharing (CORS) is restricted to the frontend origin (http://localhost:8081) in both the Spring Security configuration and controller-level @CrossOrigin annotations, preventing unauthorized cross-origin API calls. All sensitive admin endpoints require a valid ADMIN-role JWT; voter endpoints require a VOTER-role JWT.

V. VOTING ELIGIBILITY ENFORCEMENT

The system enforces multi-level geographic restrictions for government elections. Each election's description field stores a JSON document specifying the election's state, cities, and constituencies. When a voter attempts to cast a vote, VotingService verifies that the voter's registered state matches the election's target state. For college and department elections, all registered voters are eligible.

Age-based eligibility is enforced dynamically: voters must be at least 18 years old
Fig. 1. System Architecture Overview

Backend Components

The backend is structured around five primary controllers: AuthController handles OTP generation and verification; VotingController manages election retrieval and vote casting; AdminController provides CRUD operations for elections, candidates, and voters; and PublicController exposes unauthenticated statistics endpoints. All controllers are protected via a custom JwtAuthFilter that validates bearer tokens on every request.

The service layer encapsulates business logic including OTP lifecycle management (OtpService), AES encryption (EncryptionService), JWT operations (JwtService), vote casting with eligibility checks (VotingService), and administrative operations (AdminService).

Frontend Components

The React frontend comprises distinct pages for the voter login flow, voting page, and admin dashboard. The admin dashboard provides comprehensive management capabilities including election and candidate CRUD, bulk voter CSV upload, real-time charts, and PDF/CSV export. The application supports bilingual operation in English and Tamil.(calculated from date of birth at the time of registration), and candidates must be at least 25 years old. These constraints are validated at both the frontend UI layer and the backend service layer.The system also prevents double voting at the database level. Each voter has a hasVoted flag that is set atomically upon a successful vote cast, and the Vote entity includes a unique constraint on (voterId, electionId).

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904271

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. IMPLEMENTATION DETAILS

A. Technology Stack

Component	Technology
Backend	Spring Boot 3.x, Java 21
Frontend	React 18, Vite, Tailwind CSS
Database	MySQL 8.0, JPA/Hibernate
Auth	JWT (JJWT 0.11.5), BCrypt
Encryption	AES-256 CBC, PBKDF2
SMS	Fast2SMS Bulk API
UI Library	shadcn/ui, Recharts

TABLE I. Technology Stack

B. OTP Flow

The complete OTP authentication flow is summarized as follows: The voter submits an Aadhaar number; the backend looks up the associated mobile number; a SecureRandom 6-digit OTP is generated; the OTP is hashed with BCrypt and stored with a 5- minute expiry; the plaintext OTP is dispatched via Fast2SMS; the voter submits the OTP; and the backend verifies the BCrypt hash, enforcing attempt limits and expiry constraints before issuing a JWT.

VII. RESULTS AND DISCUSSION

The system was deployed and tested in a local environment with a MySQL database populated with test voter and election data. End-to-end voting flows were validated for all three election types: College, Department, and Government. The geographic restriction mechanism correctly prevented out-of- constituency voters from casting votes in government elections.

Security testing confirmed that: BCrypt- stored OTPs cannot be recovered from the database without the plaintext; JWT tokens are rejected after expiry; the attempt-limit mechanism correctly blocks access after 3 failed OTP submissions; and the double- voting prevention mechanism functions correctly under concurrent requests.

The admin dashboard successfully demonstrated real-time vote tallying with Recharts bar and pie charts, bulk voter upload via CSV with duplicate detection, PDF and CSV export, and state/city/constituency filtering.

VIII. CONCLUSION

This paper presented the design and implementation of a Secure Digital eVoting System integrating Aadhaar-based OTP authentication with AES-256 vote encryption and JWT session management. The system demonstrates that a practically deployable, secure, and accessible eVoting platform can be constructed using standard open- source technologies. Future work will explore integration with the actual UIDAI Aadhaar API, implementation of end-to-end verifiability using cryptographic receipts, and a blockchain-based immutable audit log.

REFERENCES

1. R. L. Rivest and J. Wack, "On the Notion of 'Software Independence' in Voting Systems," Philosophical Transactions of the Royal Society A, vol. 366, pp. 3759–3767, 2008.Chaum, "Secret-Ballot Receipts: True Voter-Verifiable Elections," IEEE Security & Privacy, vol. 2, no. 1, pp. 38–47, 2004.
2. D. Chaum, "Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms," Communications of the ACM, vol. 24, no. 2, pp. 84–90, 1981.
3. B. Adida, "Helios: Web-based Open-Audit Voting," in Proc. 17th USENIX Security Symposium, pp. 335–348, 2008.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4. J. A. Halderman and R. Teague, "Choices and Challenges in Electronic Voting in Victoria," in Proc. 3rd Int. Conf. e-Voting and Identity, 2011.
5. M. Hapsara, A. Imran, and T. Turner, "eVoting in Developing Countries," in Proc. 6th Int. Conf. Information Technology and Applications, 2017.Hjálmarsson et al., "Blockchain-Based E-Voting System," in Proc. IEEE 11th Int. Conf. Cloud Computing, pp. 983–986, 2018.
6. UIDAI, "Aadhaar Authentication API," Unique Identification Authority of India, Tech. Rep., 2023. [Online]. Available: https://uidai.gov.in
7. N. Ferguson, B. Schneier, and T. Kohno, Cryptography Engineering. Indianapolis: Wiley, 2010.
8. M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, May 2015.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com