Selection of Cryptographic Techniques

for

E-VoteSecure - An Online Voting System

Prepared by Group InfoSec

11.10.2024

Table of Contents

1. Introduction

This document outlines the selection of cryptographic techniques for securing various components of the proposed online voting system. It builds on the security requirements identified in conventional (manual) voting systems. The primary focus is to enforce confidentiality, integrity, authentication, non-repudiation, and voter anonymity while preserving voter verifiability.

2. System Overview and Assumptions

- **Voter Identity Management:** The NIC number assigned to each voter is used alongside a registered mobile device to perform two-factor authentication (2FA).
- **Secure Devices and Network:** The system assumes that voter devices are safe against local adversarial activities and that the election server they communicate with is protected and encrypted.
- **Adversary Model:** The system is designed to withstand network-based and server-side threats such as MITM attacks, replay attacks, and data manipulation attacks.
- **Blockchain Infrastructure:** Blockchain is for storing votes, and the record is unalterable, which makes the process checked and verifiable by other parties.

3. Cryptographic Techniques Overview

System Component	Cryptographic Technique	Purpose
Voter Authentication	AES-256 (CBC mode), HMAC-SHA256	To assure the confidentiality and integrity of OTP and biometric data
Communication Channels	TLS 1.3, RSA, AES-256	To provide secure communication and data confidentiality
Vote Encryptions	AES-256 (CBC mode)	To ensure the confidentiality of vote data
Blockchain Vote Storage	ECDSA (Elliptic Curve)	To ensure integrity and non-repudiation of votes
Hashing of Votes	SHA-3 (256-bit)	To ensure vote data integrity
Key Management	RSA-2048, ECDH	For secure key exchange
Audit Logs	SHA-3 (512-bit), AES-256	To ensure the integrity and confidentiality of logs
Random Number Generation	NIST-approved CSPRNG	To generate cryptographic keys

4. Cryptographic Techniques for Specific Components

4.1. Voter Authentication

- **Encryption Algorithm:** AES-256 (CBC mode) with a 128-bit initialization vector (IV).
- **Purpose:** OTPs used in 2-factor authentication are encrypted to ensure confidentiality during transmission.
- **Integrity Algorithm:** HMAC-SHA256
- **Justification:** AES-256 provides strong confidentiality for sensitive data, while HMAC-SHA256 ensures integrity during transmission, preventing tampering.

4.2. Secure Communication Channels

- **Protocol:** TLS 1.3, RSA 2048 bit for key exchange AES 256 for session encryption.
- **Justification:** The new encryption method in TLS 1.3 guarantees the security of the data by providing perfect forward secrecy and encryption of the data exchanged between the voter's device and the election server.

4.3. Vote Encryption

- **Encryption Algorithm:** AES-256 (CBC mode) with a 128-bit initialization vector (IV)
- **Purpose:** so that no vote is shared with any other person from the time of its placement to the time it gets connected to the election server.
- **Justification:** Based on the AES-256 in CBC mode, vote content cannot be accessed by unauthorized persons.

4.4. Blockchain Vote Storage

- **Digital Signature Algorithm:** ECDSA (Elliptic Curve Digital Signature Algorithm) with P-256 curve.
- **Purpose:** Each vote stored in the blockchain is digitally signed to prove its authenticity and hence its integrity.
- **Justification:** ECDSA is an efficient algorithm that provides comparable security guarantees with smaller keys, therefore useful in balancing large data amounts in blockchains.

4.5. Hashing Votes

- **Hashing Algorithm:** SHA-3 (256-bit).
- **Purpose:** It adds hashes votes before storage to ensure that they cannot be changed by a third party.
- **Justification:** SHA-3 offers a safeguard against collision attacks hence vote integrity is achieved.

4.6. Key Management

- **Key Exchange Algorithm**: RSA-2048 for initial key exchange, followed by ECDH for generating session keys.
- **Justification:** RSA is used for entity authentication for the first time while ECDH guarantees the confidentiality of future conversations, and makes session keys existing safe in case of threat to long-term keys.

4.7. Audit Logs

- **Integrity Algorithm:** SHA-3 (512-bit).
- **Purpose:** Authorizes that no one can modify the audit logs after they have been created.
- **Encryption Algorithm:** AES-256 (CBC mode).
- **Justification:** SHA-3 is a secure hash algorithm to ensure the hashing while AES-256 is the source of security from unauthorized access to logs.

4.8. Random Number Generation

- **Algorithm:** NIST endorsed Cryptographically Secure Pseudorandom Number Generator (CSPRNG).
- **Purpose:** Produces the symmetric and asymmetric encryption keys, other initialization vectors and random data needed for safe encrypting and signatures.
- **Justification:** CSPRNGs guarantee that all cryptographic values produced are both random and also secure.

5. Incorporation of Blind Signature Scheme

A blind signature technique is implemented into the system under which a voter identity shall be concealed, but at the same time, a vote shall be genuine.

- **Voter Preparation:** This vote is then blinded and the blinding key held by the voter and a blinded vote is then cast and transmitted to the election authority.

- **Blind Signature:** Because of blinding, the election authority signs the vote without having a clue of its content.

- **Vote Submission:** A voter unblinds the signed vote and then submits it with assurance that the vote tally is genuine but the identity of the voter cannot be established.

- **Algorithm**: RSA-based Blind Signature using a blinded key which is 2048 bit long.

- **Blinding Factor:** Generated using a CSPRNG.

- **Justification:** Blind signatures confirm the election authority is unable to establish a link that relates to the signed vote of the voter.

6. Security Parameters and Guidelines

- RSA Key Length: 2048 bits
- AES Key Length: 256 bits
- ECDSA Curve: secp256k1
- SHA-3 Block Size: 256/512 bits
- Session Key Exchange: ECDHE (Elliptic Curve Diffie-Hellman Ephemeral).
- Initialization Vectors: Must be random and generated using CSPRNGs and can not be used more than once.
- Key Rotation: This should be done for every 10000 votes to enhance security since exposure to a certain key is dangerous.

7. Conclusion

As described in this document the cryptographic techniques outlined shall help in the securing of an online election system. The maturity of the system lies in the use of symmetric and asymmetric encryption, secure hashing and blind signature whereby voter anonymity, vote confidentiality and election integrity are guaranteed. The responsibility of keeping the systems fortified against new threats should be done after a certain time or sometimes it can be done periodically.