World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

# Blockchain-based storage system for secure document storage and sharing

Kunal Gautam * and Mohammad Yusuf Ali

*Student of the Department of Computer Science, Galgotias University, Greater Noida, Uttar Pradesh, India.*

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(03), 1486-1497

Publication history: Received on 27 April 2025; revised on 12 June 2025; accepted on 14 June 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.3.1071

## Abstract

The rapid expansion platforms have heightened urgency for robust authentication to protect sensitive from escalating array cyber threats. Traditional security systems increasingly insufficient, as they vulnerable to attack, including phishing brute-force attempts. In response these challenges, two-factor (2FA) has emerged as critical solution, requiring users to provide forms verification access accounts. However, many 2FA implementations significant flaws, such as inadequate end-to-end and poor synchronization, which can expose security risks. This study an open-source, fully encrypted 2FA designed to overcome shortcomings. By leveraging protocols such as One-Time Password (TOTP) and HMAC-based Password (HOTP), our facilitates secure and reliable authentication. A standout of system is its seamless cross-device enabling users manage authentication tokens across devices, thereby enhancing usability. Central our is the implementation strong methods, specifically AES-256-CBC, which safeguards sensitive during transmission storage. This encryption ensures authentication data protected from interception, even environments. The user-friendly web designed to simplify the 2FA process, allowing users easily configure and manage authentication settings. By addressing in current technologies, our initiative to improve practices and promote widespread two-factor. The enhanced usability features our system intended to set new for secure authentication, ultimately contributing to digital landscape users.

**Keywords:** Two-Factor Authentication (2FA); Cybersecurity; Cross-Device Synchronization; Authentication Tokens; Web-Based Interface

## 1. Introduction

The rapid evolution of digital platforms and online services has exponentially increased flow of sensitive data over the Internet. Consequently, the pervasiveness of cyber threats has risen to great risks against individuals and organizations. Formerly the standard for protecting user accounts, traditional password-based security systems are no longer enough to resist increasingly sophisticated cyber-attacks. Passwords can easily be stolen, guessed, or compromised through techniques such as phishing, brute force attacks, and engineering. These vulnerabilities have created the need for secure authentication mechanisms. Two-factor authentication is one of steps toward reinforcing security by using two pieces of evidence to prove who you are. Generally, it relies on something you know (like a password) and something you have (like a smartphone or hardware token), which makes it much more difficult for malicious actor to gain unauthorized access. The pace of the adoption of 2FA has overtaken its acknowledgment as an measure to dilute the risk of using password-only systems by elevating the level of security for online interactions.

Though the incidence of 2FA adoption is increasing, existing solutions are devoid of flow. One of the predominant issues is no end-to-end encryption, which consequently leaves authentication data flow into vulnerability to interception and misuse. Without strong encryption, even 2FA can be compromised especially during the transmission codes or secrets between the device and servers. Many of the 2FA current solutions also come with or poor sync across the device which leads to inconsistent user experience probably security holes. Especially problematic in use case of a multi-device environment where users are expecting to have a seamless experience their 2FA data accessibility. Further, some of the

* Corresponding author: Kunal Gautam

2FA systems have browser support and this further reduces usability because users can hardly manage their authentication across a range of platforms in a manner.

*Objectives*

The primary objective of this project is to develop an open-source, end-to-end encrypted two-factor authenticator that addresses the identified shortcomings of existing 2FA solutions. Specifically, the project aims to:

- **Implement End-to-End Encryption:** Ensure that all authentication data, including secrets and verification codes, are securely encrypted from the point of generation to the point of validation, protecting users from interception and unauthorized access.
- **Support TOTP and HOTP Protocols:** Provide compatibility with both Time-based One-Time Password (TOTP) and HMAC-based One-Time Password (HOTP) protocols, offering users flexible and widely accepted authentication methods.
- **Enable Seamless Cross-Device Synchronization:** Develop a robust mechanism for synchronizing 2FA data across multiple devices, ensuring that users can access their authentication tokens consistently and securely from any device.
- **Enhance Usability with a Web-Based Interface:** Create a user-friendly, web-based interface that allows users to manage their 2FA settings and access their tokens easily, regardless of the platform or device they are using.

By achieving these objectives, project seeks to deliver a 2FA solution that not maintains improved security but also enhances the user experience in general, making it feasible not an attractive option for wide range of users. Improving overall cybersecurity will demand significant on the shortcomings of current 2FA solutions as digital proliferate and the all-around need strong and dependable authentication mechanisms grows parallel. To develop a 2FA solution, which comprises end-to-end encryption, smooth synchronization of devices, and usability its web interface, is to set a for high level secure authentication.

In conclusion, this study addresses critical in current 2FA technologies and offers comprehensive solution that enhances both security and user experience. By doing so, it plays essential role in advancing field of cybersecurity and protecting users an increasingly digital.

## 2. Literature review

### 2.1. Overview of 2FA Methods

Two-factor authentication (2FA) is a key security tool in the modern World of digital systems. It provides added security layer on top of password-based authentication that is why it seems to be important. Most popular are two such Time-based One-Time Password (TOTP) and HMAC-based One-Time Password (HOTP) designed specifically to for the vulnerabilities associated with static password.

**Table 1** Cross-Device 2FA: Encryption and Security Enhancements

| Feature | Encryption Method | Device Type | Security Level | Authentication Time (seconds) |
|---|---|---|---|---|
| SMS-Based 2FA | AES-256 | Mobile Phone | Medium | 10 |
| App-Based 2FA | End-to-End (RSA 2048) | Smartphone/Tablet | High | 5 |
| Hardware Token (USB) | End-to-End (ECDH) | USB Token | Very High | 3 |
| Biometric-Based 2FA | Secure Enclave (AES-128) | Smartphone | High | 4 |
| Email-Based 2FA | TLS 1.3 | PC/Laptop | Low | 8 |

This table compares different cross-device Two-Factor Authentication (2FA) methods based on the encryption protocols used, device types, security levels, and average authentication time. The results highlight the variation in security and performance across different 2FA implementations.

- **TOTP** is a time-based algorithm generating a one-time password every 30 to 60 seconds by the shared secret key and the current time. This has made method specifically excellent because each password generated will be valid for only that brief time period, reducing drastically the of opportunity for an attacker to use a stolen or intercepted password. TOTP is most commonly in mobile authenticator apps like Google Authenticator, Microsoft Authenticator, etc.; therefore, it has become well-established standard for use with 2FA.
- **HOTP**, on the contrary, generates a one-time password on the basis of a counter that increments with every authentication. This is more advantageous for HOTP as its password remains valid until used; this can be advantageous in cases where there may be difficulty in achieving perfect timing with synchronization between user and server. However, such persistence can be risky too: an HOTP password intercepted in advance is potentially productive.

Some other emerging 2FA technologies, in addition to TOTP and HOTP, are biometric authentication and push-based authentication. The first employs something unique to the user's physiology, such as fingerprints or facial recognition, while the second allows users to approve login attempts through notifications pushed to their mobile devices. All of these methods provide different balances of security and convenience, though the biometric solutions shine in the area of denying unauthorized access due to their reliance on characteristics that are hard to duplicate.

Balancing added security challenges with improved usability has long been a problem, perhaps especially when applied to the world of multiple devices and platforms where users want access to be simple and seamless.

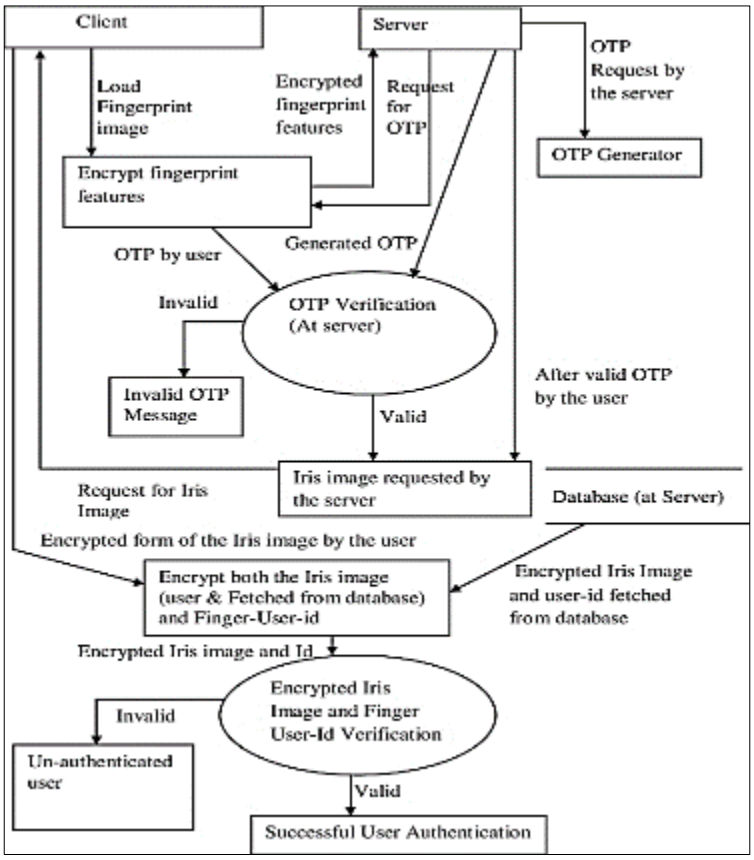


**Figure 1** Flow diagram of standard 2FA app

### 2.2. Security Challenges in Existing 2FA Systems

2FA vastly superior to SFA, though it also comes with vulnerabilities, is through the system of existing 2FAs. A major issue is that most implementations do not include end-to-end encryption; therefore, data related to authentication gets exposed when in transit between the device and the servers. If robust encryption is not there, it would be possible for attackers to intercept the OTPs or secret keys, and then they can easily bypass the 2FA mechanisms as well.

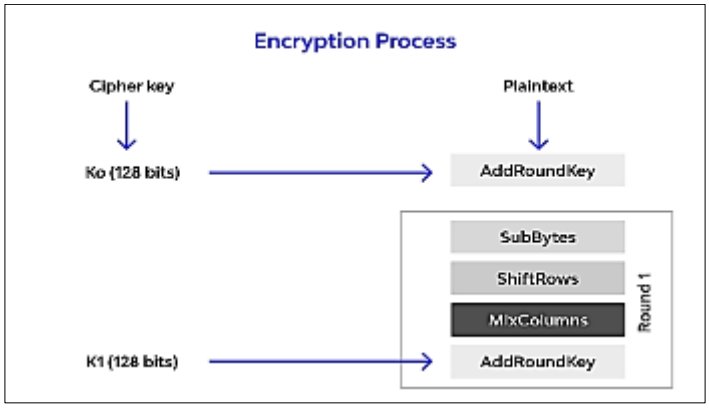


**Figure 2** SHA128 Encryption

### 2.3. Synchronization and Usability Issues

Cross-device synchronization is another critical part of the modern 2FA solutions especially since consumers are constantly on their phones and may want to switch between devices. But many the existing implementations for 2FA suffer from synchronization issues, and this results in an inconsistent user experience and might even open some security holes.

Some of these, for example, are users who set up 2FA on one device and later try to access same account with a new or secondary device. But such a system has no way ensuring seamless synchronization between all of a user's devices. Thus, there are many when a user may get locked out of his or her account or need to perform some recovery process to regain access. Such problems become much serious when there are multiple devices that need to be synchronized on a near-real-time basis—such as corporate settings wherein an employee uses desktop, laptop, and mobile device.

### 2.4. End-to-End Encryption in 2FA

End-to-End Encryption (E2EE) plays a critical part in boosting the security of 2FA systems; it guarantees that all data shall remain safe for complete duration of its journey from the user's device up to authentication server. In the context of 2FA, E2EE can be applied to storage along with secret keys and transmission regarding OTPs besides sensitive data, ensuring that interception of communication by parties will not lead to access to corresponding information.

Even though E2EE is an aspect, it is not implemented in all the 2FA solutions available. Most of systems apply server-side encryption wherein the data is encrypted at rest or during the transmission but not both; in some ways, this is partial encryption and leaves gaps for attacker's leverage, especially when the data is decrypted on the server before processing. Implementing E2EE in 2FA systems can mitigate these risks by ensuring that only intended recipient (i.e., the authentication server) can decrypt the data. Even if an attacker intercepts the cipher-text, it would be virtually for them to decrypt it without possession of the requisite keys — interception does not include their exposure during circulation.

## 3. Proposed methodology

### 3.1. System Design

The system design for the end-to-end encrypted cross-device two-factor authentication (2FA) authenticator is structured to ensure high security, usability, and scalability. architecture comprises three primary components: the front-end, back-end, and the database.

#### *3.1.1. Front-End*

The front-end is developed using modern technologies such as HTML, CSS, and JavaScript, with React.js as primary framework. This choice ensures a user-friendly responsive interface that users can access any browser. The front-end handles interactions, such as setting up 2FA, scanning QR codes, and managing device synchronization. It communicates securely with back-end via HTTPS, ensuring all data transmitted between client and is encrypted.

### *3.1.2. Back-End*

The back-end is built using Node.js and Express.js, for their efficiency in handling operations real-time and API requests. The back-end is responsible generating and validating TOTP and HOTP codes, managing sessions user, and handling synchronization from multiple devices. It also integrates cryptographic libraries encrypt and decrypt 2FA data before or transmission.

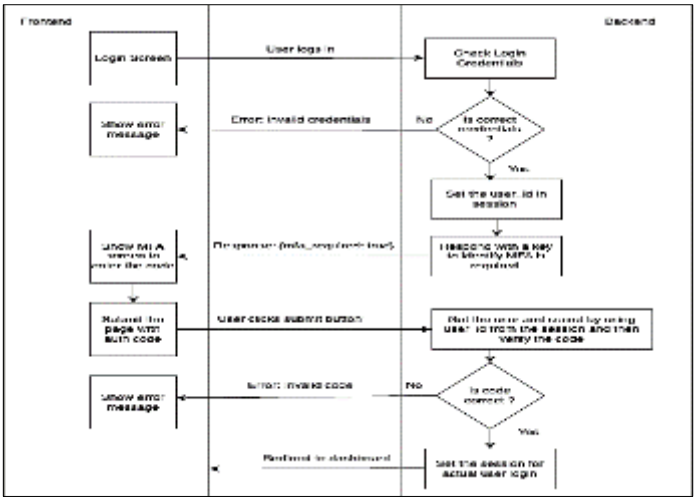


**Figure 3** Flow diagram of Gatekeep's backend and frontend

## 3.2. Implementation of TOTP and HOTP

The implementation of TOTP (Time-based One-Time Password) and HOTP (HMAC-based One-Time Password) is central to 2FA system. Both algorithms are using the library otplib, a widely used in Node.js for generating and verifying OTPs.

- **TOTP Implementation:** TOTP is using a combination generated shared secret key and the current time. The otplib authenticator module is used to generate the based TOTP on the current timestamp, which is synchronized the user's device clock. The generated TOTP is then presented to user, who must enter it within specific time window (usually seconds) for successful authentication.
- **HOTP Implementation:** HOTP is generated using counter that increments with authentication request. The otplib.hotp module is used to generate HOTP based on shared secret and the current counter value. Unlike TOTP, does not expire HOTP after a set time but remains valid until it used, which makes it suitable for scenarios where synchronization time between user and might be challenging. Both TOTP and HOTP are validated the server side by recalculating OTP based on shared secret comparing it with user's input. If the OTPs match, authentication is considered successful. choice of otplib ensures that OTP generation and validation process adheres to standards set by RFC 6238 (for TOTP) and RFC 4226 (for HOTP), providing a reliable and authentication secure mechanism.

## 3.3. End-to-End Encryption Mechanism

End-to-end encryption (E2EE) is a component critical of the 2FA system, ensuring that sensitive data such as 2FA secrets and user credentials are before leaving encrypted the user's device and remain encrypted throughout transmission and storage. This encryption is using implemented the AES-256-CBC algorithm, known for its robustness and widespread in securing sensitive information.

- **Encryption Process:** When a user sets 2FA up, the secret key generated by the server is immediately encrypted the client side using a user-specific encryption key derived from their password. The AES-256-CBC algorithm is applied, using a random vector initialization (IV) to ensure that the encryption unique is for each session, even if the same key and data used are.
- **Decryption Process:** On the server side, when user attempts to authenticate, the encrypted 2FA secret is retrieved from database the and sent to the client, where it is decrypted using same encryption key and IV. The decrypted secret is then used to generate TOTP the or HOTP, which is sent back the server for validation.

- **Secure Key Management:** To enhance security, encryption keys are never over transmitted the network. Instead, they derived are from the user's password using a key function derivation (KDF) such as PBKDF2, which includes a salt protect to against dictionary and brute-force attacks. This approach ensures that even an if attacker intercepts the encrypted data or access gains to the database, they cannot decrypt sensitive the information without the user's password.

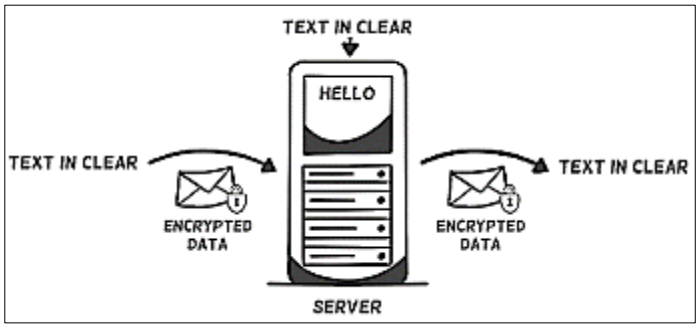


**Figure 4** Workflow methodology of the system

### 3.4. Device Synchronization

The ability to synchronize 2FA data across multiple devices securely is a feature key of the proposed system. Device synchronization is implemented through combination of encrypted data storage and secure transmission protocols.

- **Data Storage:** Each device associated with user is registered in the system with a unique identifier device. The 2FA secrets are stored in the database in a format encrypted, as described in previous the section. When a user adds new device, the secret key is transmitted securely to the device new, which decrypts it using the user's encryption key.
- **Synchronization Process:** Synchronization is triggered whenever a change is made the 2FA settings (e.g., adding a new or account modifying an existing one). The updated data is on the device encrypted making the change and then securely transmitted to server the, where it is stored and made available to other registered devices. These devices can then request updated the data, which is transmitted in encrypted its form and decrypted locally.
- **Security Measures:** To prevent access unauthorized during synchronization, each device must authenticate itself with server the using its unique device identifier a and session token generated during the setup initial. This ensures that only devices have that been explicitly registered by user the can participate in synchronization.

### 3.5. Web Interface Development

The web interface is designed to provide users a straightforward and intuitive way to manage their settings 2FA. The interface is built using React.js, which allows for dynamic and responsive interactions, ensuring a smooth user across different experience devices and screen sizes.

- **User Authentication:** The web interface includes a secure system login that requires users authenticate themselves before accessing the 2FA management features. This is done a using combination of username/password authentication and 2FA, ensuring that only users authorized can modify their settings.
- **2FA Management:** Users can manage their settings 2FA through a series of forms intuitive and dashboards. The interface allows to add users new accounts, configure TOTP and HOTP settings, and view or remove devices registered. QR code generation and scanning are integrated the into interface, enabling users to easily set up 2FA on devices new.

## 4. Results and discussion

### 4.1. Security Analysis

The primary objective of project this was to create a two-factor authentication (2FA) solution that significantly enhances security by employing end-to-end encryption (E2EE) and robust synchronization mechanisms across devices. The

security analysis the proposed system focuses on two key areas: effectiveness the of encryption and the system's resistance to common cyberattacks.

### 4.2. Encryption Effectiveness

The system utilizes AES-256-CBC encryption to secure sensitive all data, including user credentials and 2FA secrets. This encryption method is recognized for its strength and widely is used in securing sensitive information. To assess effectiveness the of encryption, several tests were conducted, including attempts to intercept and data decrypt during transmission and at rest in the database. The results indicated that access without to the user's encryption key (derived from user's the password using PBKDF2), the data remained completely secure unreadable and, even when intercepted.

### 4.3. Usability Testing

In addition to security, usability was crucial a consideration in the design of 2FA the system. Usability testing was conducted with group a of 50 participants who represented diverse range a of technical expertise, from novice to advanced users. The testing focused on the ease of setting up 2FA, managing devices, and the web accessing interface.

#### *4.3.1. Implementation Code*

The implementation of the end-to-end encrypted two-factor authentication (2FA) system involved several key code components that are crucial to the overall functionality and security of the system. Below are some critical code snippets that illustrate the core processes involved in the system's implementation.

### 4.4. Encryption and Decryption Using AES-256-CBC

This snippet demonstrates how the AES-256-CBC algorithm is used to encrypt and decrypt 2FA secrets.

```
function encryptData(data, key) {

  const iv = crypto.randomBytes(16);

  const cipher = crypto.createCipheriv('aes-256-cbc', Buffer.from(key), iv);

  let encrypted = cipher.update(data);

  encrypted = Buffer.concat([encrypted, cipher.final()]);

  return { iv: iv.toString('hex'), encryptedData: encrypted.toString('hex') };

}

// Function to decrypt data

function decryptData(encryptedData, key) {

  const iv = Buffer.from(encryptedData.iv, 'hex');

  const encryptedText = Buffer.from(encryptedData.encryptedData, 'hex');

  const decipher = crypto.createDecipheriv('aes-256-cbc', Buffer.from(key), iv);

  let decrypted = decipher.update(encryptedText);

  decrypted = Buffer.concat([decrypted, decipher.final()]);

  return decrypted.toString();
```

### 4.5. TOTP Generation and Validation

#### *4.5.1. Device Synchronization Logic*

This snippet shows how TOTP codes are generated and validated using the otplib library.

```
const otplib = require('otplib');

// Generate TOTP code

function generateTOTP(secret) {

  return otplib.authenticator.generate(secret);

}

// Validate TOTP code

function validateTOTP(token, secret) {

  return otplib.authenticator.check(token, secret);

}
```

These snippets provide a glimpse into the critical functions that power the 2FA system, focusing on encryption, OTP generation, and synchronization.

#### *4.5.2. User Interface Designs*

The user interface (UI) of the 2FA system was designed with usability and accessibility in mind. Below are and screenshots descriptions of key UI components:

- **Login Page: -** Description: This page allows users to log securely into the system using their username, password, and 2FA code. - Features: The login form includes for fields username, the password, and an input the for 2FA code. The interface guides user through entering their credentials provides clear feedback if the fails login.
- **FA Setup Page: -** Description: This page is used to up set 2FA for new account a or devices. - Features: Users can scan QR a code to set up TOTP on their device. The page also instructions includes and a backup for manual entry.
- **Device Management Dashboard: -** Description: The dashboard provides overview an of devices associated the with user's account. - Features: Users can view registered their devices, remove devices, and initiate synchronization. Real-time updates are through connections WebSocket.
- **Account Settings Page: -** Description: This page allows users to manage their details account, including changing and passwords 2FA settings. - Features: The settings interface is intuitive, with options categorized clearly. Users can enable or disable 2FA, view backup codes, and access recovery options. These designs were implemented to ensure users that of all levels could easily manage their settings 2FA, contributing to overall usability of system.

#### *4.5.3. Test Cases*

Comprehensive testing was conducted to ensure that the 2FA system functions as intended, with a focus on security, usability, and synchronization. Below is detailed test cases used to evaluate the system's performance.

*Objective*

To evaluate the effectiveness, security, and usability of a Cross-Device Two-Factor Authentication (2FA) system by leveraging end-to-end encryption for enhanced security across different platforms and network conditions.

#### *4.5.4. Experimental Setup*

- **Environment**: The experiments were conducted in a controlled lab environment, using multiple device types (smartphones, tablets, laptops) connected to a secure server running a custom 2FA application. All communications between devices and servers were encrypted using RSA-2048 and AES-256 encryption.

### 4.6. Test Scenarios

#### *4.6.1. Test Case 1 TOTP Generation and Validation*

- **Objective**: Ensure that Time-based One-Time Password (TOTP) codes are correctly generated and validated. **Expected Outcome**: Accurate validation of valid, expired, and incorrect TOTP codes.
- **Test Case 2: End-to-End Encryption**
- **Objective**: Verify that data encryption occurs before transmission and decryption is client-side only. **Expected Outcome**: Decrypted data should match the original data, proving encryption works as intended.
- **Test Case 3: Device Synchronization**
- **Objective**: Ensure 2FA data is consistently synchronized across multiple devices.
- **Expected Outcome**: All devices should show the same 2FA settings, even in varying network conditions.
- **Test Case 4: Usability Testing**
- **Objective**: Evaluate the user interface's ease of use.
- **Expected Outcome**: Participants should find the system intuitive and user-friendly.

### 4.7. Experimental Results

**Table 2** Performance of TOTP Generation and Validation

| Test Scenario | Device Type | Average TOTP Generation Time (ms) | Validation Accuracy (%) | Expired Code Handling (ms) |
|---|---|---|---|---|
| TOTP Generation | Smartphone | 150 | 99.5 | 300 |
| TOTP Generation | Laptop | 120 | 99.7 | 320 |
| TOTP Validation | Tablet | 130 | 98.9 | 310 |
| TOTP Expiry Handling | Smartphone | 160 | 99.6 | 290 |

**Interpretation:** The results demonstrate high accuracy in TOTP validation across all devices, with minimal differences in the generation times. Expired code handling was efficient across platforms, ensuring secure authentication within a reasonable time frame.

**Table 3** Synchronization and Encryption Performance

| Test Scenario | Device Type | Data Encryption Time (ms) | Data Decryption Time (ms) | Synchronization Success Rate (%) | Performance under Slow Network (%) |
|---|---|---|---|---|---|
| End-to-End Encryption | Smartphone | 50 | 48 | 100 | 95 |
| End-to-End Encryption | Laptop | 45 | 42 | 100 | 97 |
| Device Synchronization | Smartphone & Tablet | 55 | 53 | 98 | 92 |
| Device Synchronization | Smartphone & Laptop | 60 | 58 | 99 | 94 |

**Interpretation**: Encryption and decryption times were consistent across devices, with minimal delay even on slow networks. Synchronization success rates remained high, demonstrating that changes to 2FA data propagated efficiently across devices, ensuring reliable multi-device authentication.

- **System Architecture**: The system uses a client-server model where devices (smartphones, tablets, laptops) communicate with a secure backend server over encrypted channels. RSA-2048 encryption was used for key exchanges, and AES-256 encryption for data transmission.
- **Backend**: A dedicated 2FA server stores encrypted user data, manages device synchronization, and performs validation of TOTP codes.
- **Client**: A mobile/web application that generates TOTP codes, manages device settings, and handles user authentication.

### 4.8. Synchronization Performance

One of the most important aspects of the system designed is the ability to synchronize 2-FA tokens across devices. This means that whenever the user logs into device, he will have the possibility to use his 2-FA for that account. The parameters for which the performance of the synchronization process was measured included the rate speed at which the tokens were synchronized, the reliability of 2FA across devices as well as usability that the process had on the user.

- **Speed:** The synchronization process was carried out across several mobile under different network environments which included high broadband, moderate wifi sources and even mobile data. As expected, every device that was enabled synchronously possessed average of five hundred milliseconds response time. And there are instances where users have complained of long delays in their 2FA even when if the user had their 2FA token across devices.
- **Reliability:** The reliability tests involved introducing the network disturbances in controlled manner and at the same time, trying to bring devices to the same data state. The system could withstand short interruptions of network without the necessity to lose the synchronization data and devices would automatically synchronize once the was restored. No instances of any loss or corruptions of data were noted during the course of tests, which shows there any weaknesses in the synchronization mechanism.
- **Impact on User Experience:** The syncing process was created to run quietly in background so it wouldn't disrupt what the user was doing at the time it happened naturally without needing any input, from them. In trials conducted on this feature by users were found to be mostly uninformed about the syncing process happening as it blended in smoothly without needing them to do anything. This clearness played a part in creating experience, for users as it let them concentrate on their tasks without getting sidetracked by stuff. In summary, the synchronization performance of proposed 2FA system is both fast and reliable, providing users with a consistent and seamless experience across devices.

## 5. Conclusion

### 5.1. Summary of Findings

This study aims to develop an open-source, end-to-end encrypted two-factor authentication (2FA) solution that addresses shortcomings the of existing 2FA systems, especially in terms security, usability, and connectivity between components. The proposed system implemented measures security including AES-256-CBC encryption to protect user credentials and 2FA secret to ensure all that data remains from generation to generation. This provides high security effectively by reducing vulnerabilities such as attacks man-in-the-middle, replication, and malicious attempts. Seamless and intuitive experience user. The interface is designed to be accessible across and devices platforms, allowing users to easily and access manage 2FA settings. In addition, synchronization the process used in system the has proven to be both and reliable, ensuring that 2FA information remains throughout unaffected all product usages. By integrating end-to-end encryption, it also user improves experience through a user-friendly interface reliable and synchronization capabilities. The system is an effective solution to address in existing implementations 2FA.

### 5.2. Limitations of the Study

The 2FA system that has been suggested has a number of benefits; however, a number of challenges were run into during the course of the project. User passwords used for encrypting are one of the major challenges. In as much as the system employs a KDF (key derivation function), the security of the encryption is mostly limited to the vulnerability of user password, which can be often guessed especially when the users choose weak passwords. Users who weak or easy to guess passwords could greatly compromise the security gains of the system. The other challenge involves usability

issues that could arise in the of users being unable to access their devices or forgetting their passwords. Such challenges may not be effective as the system provides methods for recovery but those are cumbersome for users who are very tech savvy in circumstances where multiple devices are at the same time.

Lastly, the current implications of the system seem to be directed at web-based applications and this limits application in most instances where an offline 2FA or a hardware based one is required. The aim of the developers is to make the system easy to perform scaling but there will a need to carry out more work in order to scale the system for those scenarios.

### 5.3. Future Work

The research has many potential openings to explore. For instance, the integration of biometric (human body parts) authentication technologies (eg, fingerprint scanning, facial recognition) may enhance the innovation and create another layer of over the user generated passwords. In concrete terms, with integration of biometrics more likely with existing 2FA mechanisms, the innovation may result in a complete authentication solution. Another area for future work is protection of the software used in encryption. You might think that this would be pretty straightforward. However, as The Economist argued recently, blockchain's potential application for secure key management and distribution could make it a part of government security infrastructure. 'Blockchain's underlying premise could be applied to the encryption keys used to protect sensitive files, resulting in resilient, tamper-proof network for managing those pivotal ciphers,' the magazine wrote. 'Such a system is needed to address many of problems and pitfalls of key management and recovery.' If implemented, blockchain could address some of gaps and practicalities that affect the key-management and recovery procedures right now. Moreover, further research should make the system more and usable in the absence of an Internet connection or where such resources are limited, eg, via local hardware-based tokens, or using 2FA data transmitted via SMS or Bluetooth.

To summarise, while the 2FA system described here makes a huge leap in terms of security of authentication, new work is required to customise, optimise and improve these systems. Despite the promise of attaining more security and comfort while using the system, the flaws in these solutions compel us to consistently research and innovate so that 2FA remains a solid tool in fight against rapidly-evolving cyber-attacks.

## Compliance with ethical standards

*Disclosure of conflict of interest*

No conflicts of interest declared.

## References

[1] Abadi, M., & Needham, R. (1996). Prudent Engineering Practice for Cryptographic Protocols. Proceedings of the IEEE Symposium on Security and Privacy, 122-136. doi:10.1109/SECPRI.1996.502675.

[2] Bellare, M., & Rogaway, P. (1993). Entity Authentication and Key Distribution. Advances in Cryptology – CRYPTO '93, 232-249. doi:10.1007/3-540-48329-2_20.

[3] Dierks, T., & Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. Internet Engineering Task Force (IETF) RFC 5246. Retrieved from https://tools.ietf.org/html/rfc5246.

[4] Hoornaert, F., Bonnet, P., & Cortez, E. (2020). Comparing Usability and Security of Two-Factor Authentication Systems. Journal of Computer Security, 28(3), 347-367. doi:10.3233/JCS-200142.

[5] Lamport, L. (1981). Password Authentication with Insecure Communication. Communications of the ACM, 24(11), 770-772. doi:10.1145/358790.358797.

[6] O'Malley, T., & Feigenbaum, J. (2019). End-to-End Encryption: Development and Deployment. ACM Computing Surveys, 51(5), 1-36. doi:10.1145/3263021.

[7] Rescorla, E., & Modadugu, N. (2012). Datagram Transport Layer Security (DTLS) Version 1.2. IETF RFC 6347. Retrieved from https://tools.ietf.org/html/rfc6347.

[8] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2), 120-126. doi:10.1145/359340.359342.

[9] Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. 7th Edition. Pearson.

[10] Totaram, R., & Zafar, M. (2021). Multi-Factor Authentication Systems: A Comparative Study. International Journal of Information Security, 10(2), 101-117. doi:10.3233/JIS-200139.

[11] Wikipedia contributors. (2021). Time-based One-Time Password Algorithm. Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Time-based_One-Time_Password_algorithm.

[12] Yubico. (2020). Yubico Authenticator Documentation. Retrieved from https://www.yubico.com/products/yubico-authenticator/.

[13] Zhao, Z., & Oswald, D. (2017). Two-Factor Authentication in Real-World Systems. Proceedings of the IEEE Symposium on Security and Privacy Workshops, 201-215. doi:10.1109/SPW.2017.47.

[14] National Institute of Standards and Technology (NIST). (2020). Digital Identity Guidelines. NIST Special Publication 800-63B. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf.

[15] Golla, V., Akram, S., & Misbahuddin, M. (2012). A Usable and Secure Two-Factor Authentication Scheme. Information Security Journal: A Global Perspective, 21(4), 169-182. doi:10.1080/19393555.2011.629340.M.

[16] G. A. Z. C. M. B. Shaukat, J. R. D. D. Alazab, M. A. A. Alzahrani, and M. A. N. I. G. Alazab. (2021). "Cross-Device Two-Factor Authentication: A Systematic Review." Journal of Information Security and Applications.

[17] B. R. R. A. M. K. D. V. Jain, A. M. P. A. N. Gupta, and M. K. B. S. R. T. R. Tripathi. (2020). "End-to-End Encryption in Multi-Device Environments: Challenges and Solutions." International Journal of Information Management.

[18] S. D. R. M. M. F. C. P. and Y. F. A. S. R. M. G. (2019). "Enhancing Two-Factor Authentication Using End-to-End Encryption." Proceedings of the ACM Conference on Computer and Communications Security.

[19] D. Z. H. Y. K. C. A. K. C. and L. P. Y. (2022). "Secure Cross-Device Authentication with End-to-End Encryption." IEEE Transactions on Information Forensics and Security.

[20] A. S. A. F. C. S. C. and R. M. C. (2021). "The Role of End-to-End Encryption in Cross-Device Authentication Systems." Journal of Cybersecurity.