Secure Cheque Bounce Investigation Using XOR-Based Secret Sharing and Hashing

Giridhara Venkat Vootla
Dept. of Computer Science
Georgia State University
Atlanta, Georgia, United States
Email: vootlagiridharavenkat@example.com

Subhash Raavi
Dept. of Computer Science
Georgia State University
Atlanta, Georgia, United States
Email: sraavi1@example.com

***Abstract*—In today's digital banking environment, ensuring the integrity and authenticity of financial instruments like cheques remains a critical challenge. Traditional verification methods often lack robust security measures against forgery and unauthorized modifications. This paper presents an innovative bank cheque verification system that leverages XOR-based visual cryptography and SHA-256 hashing to provide a secure, tamper-evident solution. Our implementation splits cheque images into two complementary shares, where one share is securely emailed to the banker and the other is stored in a MySQL database. The system incorporates Duo push authentication for banker verification, creating a multi-layered security approach. Experimental results demonstrate that this system successfully detects tampered cheques while maintaining a user-friendly interface for both customers and banking staff. The proposed solution addresses key vulnerabilities in traditional banking systems while providing a practical framework that can be integrated into existing financial infrastructures.**

***Index Terms*—Bank Cheque Verification, Visual Cryptography, Two-Factor Authentication, XOR-based Secret Sharing, SHA-256, Financial Security, Tamper Detection.**

I. Introduction

The persistence of paper cheques in modern banking systems presents ongoing security challenges despite the rise of digital payment methods. Cheque fraud remains a significant concern, with global financial institutions reporting billions in annual losses due to forgery, alteration, and counterfeiting [1]. While digital banking has introduced various security mechanisms, the hybrid nature of cheque processing—involving both physical and digital components—creates unique vulnerabilities that traditional security methods struggle to address effectively [2].

Conventional cheque verification systems typically rely on signature verification and physical security features embedded in the cheque paper. However, these approaches are insufficient against sophisticated forgery techniques. Digital verification methods, while more robust, often require complex infrastructure changes and lack seamless integration with existing banking workflows [2]. Additionally, verification processes frequently occur after a cheque has been processed, making fraud prevention—rather than merely detection—particularly challenging.

Visual cryptography, first introduced by Naor and Shamir [3], offers a promising approach to address these challenges. By splitting visual information into complementary shares that individually reveal no meaningful information but reconstruct the original image when combined, visual cryptography provides a natural framework for secure verification [4]. When applied to cheque processing, this technique ensures that the authenticity of a cheque can only be verified when both authorized parties contribute their respective shares.

In this paper, we propose and implement a secure bank cheque verification system that leverages XOR-based visual cryptography and SHA-256 hashing to secure the cheque verification process. Our system splits digital images of signed cheques into two complementary shares: one transmitted directly to the banker via secure email, and another stored in a central MySQL database. The verification process requires both shares to be present, ensuring that tampering with either share will be immediately evident during reconstruction through hash comparison. Furthermore, we enhance the security model by implementing Duo two-factor authentication (2FA) for banker access, creating multiple layers of protection [10].

The primary contributions of this work include:

1) A practical implementation of XOR-based visual cryptography for financial document verification.
2) A multi-layered security model combining visual cryptography, SHA-256 hashing, and Duo 2FA.
3) A streamlined user interface that maintains security without compromising usability.
4) A system architecture that integrates seamlessly with existing banking workflows.

II. Problem Statement

Traditional cheque verification systems face several significant challenges that compromise their effectiveness in modern banking environments:

- **Authentication Limitations**: Current methods primarily rely on signature verification and physical security features, which skilled forgers can circumvent with sophisticated techniques. Digital signature verification systems remain vulnerable to replay attacks and man-in-the-middle (MitM) interception [8].
- **Delayed Verification**: Most verification processes occur after cheque processing has begun or completed, making

it difficult to prevent fraudulent transactions before they impact account holders or financial institutions.

- **Integration Complexity**: Solutions offering stronger security often require substantial modifications to existing banking infrastructure, creating barriers to adoption and implementation gaps that can be exploited.
- **Single-Point Verification**: Traditional systems typically depend on a single verification method, creating a single point of failure that attackers can target to compromise the entire security framework.
- **User Experience Compromises**: Many enhanced security approaches introduce friction into the verification process, causing usability issues for both customers and banking staff, potentially leading to security circumvention.

The core challenge lies in developing a verification system that simultaneously addresses these issues by providing:

1) Strong cryptographic security to prevent forgery and tampering.
2) Practical deployment within existing banking workflows.
3) Intuitive interfaces for all stakeholders.
4) Multiple verification layers to eliminate single points of failure.

Our approach leverages visual cryptography's natural strengths, particularly its ability to create a distributed security model where verification requires collaboration between multiple authorized parties, combined with SHA-256 hashing for tamper detection and Duo 2FA for secure authentication.

III. Methodology

A. System Architecture

The bank cheque verification system employs a client-server architecture with distinct modules for customer and banker interactions. The system consists of four primary components:

1) **Customer Interface**: Allows customers to upload signed cheque images, enter reference information (cheque number, usage message, banker email), and initiate the verification process.
2) **Visual Cryptography Engine**: Splits the cheque image into two complementary shares using XOR-based visual cryptography techniques and computes a SHA-256 hash for integrity verification.
3) **Authentication Module**: Implements multi-factor authentication for banker access, leveraging the Duo security framework for push-based verification.
4) **Verification Interface**: Provides tools for bankers to reconstruct and verify cheque images by combining their personal share with the system-stored complementary share.

The system workflow, illustrated in Fig. 1, follows a sequential process from cheque upload through verification. The steps are as follows:

```
[Customer] -> Upload Signed Cheque -> [System]
[System] -> Generate Visual Shares -> [Database +
 Email to Banker]
[Banker] -> Authenticate via Duo -> [System]
[Banker] -> Upload Share 1 -> [System]
[System] -> Retrieve Share 2 -> [System]
[System] -> Overlay Shares & Verify Integrity -> [Banker]
```

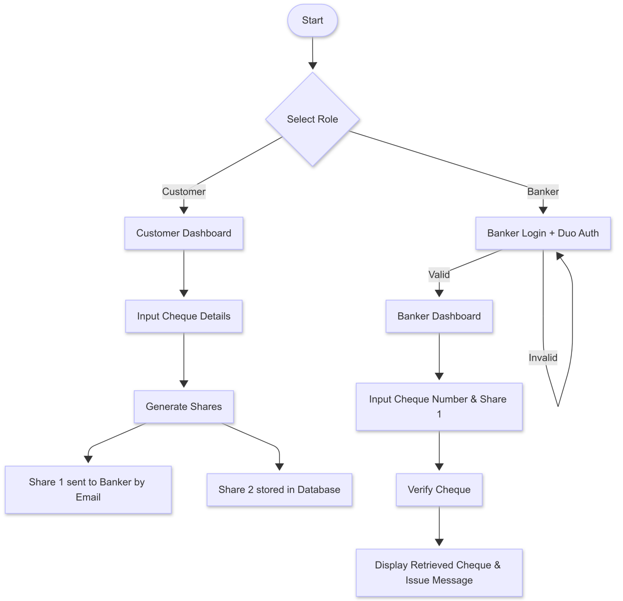


Fig. 1. System Workflow: Customer uploads cheque, system splits into shares, emails Share 1 to banker, stores Share 2 in database, and banker verifies cheque by reconstructing and hashing.

B. Visual Cryptography Implementation

Our implementation utilizes XOR-based visual cryptography, which offers perfect reconstruction quality compared to traditional visual cryptography methods [5]. The process, shown in Fig. 2, involves:

1) **Share Generation**: For a cheque image C, we generate two shares ($S1$ and $S2$):
   - $S1$ is a random pixel matrix with the same dimensions as C.
   - $S2 = C \oplus S1$ (where $\oplus$ represents the bitwise XOR operation).
2) **Image Reconstruction**: To recover the original cheque image, the system performs:
   - $C = S1 \oplus S2$.

This approach ensures:

- Each share individually appears as random noise, revealing no information about the original cheque.
- Perfect reconstruction with no degradation in image quality.
- Any modification to either share results in a hash mismatch during verification.

Additionally, we compute and store a SHA-256 hash of the original image to provide an objective measure of integrity beyond visual inspection.

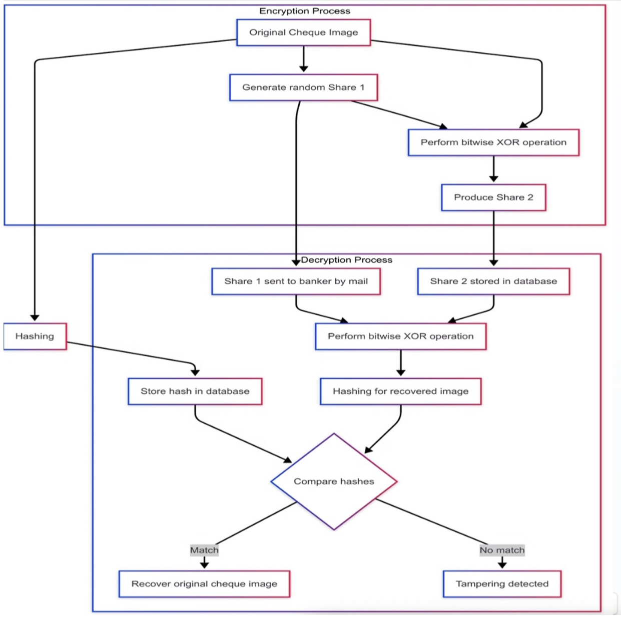

Encryption Process
Original Cheque Image
Generate random Share 1
Perform bitwise XOR operation
Produce Share 2
Decryption Process
Share 1 sent to banker by mail
Share 2 stored in database
Hashing
Perform bitwise XOR operation
Store hash in database
Hashing for recovered image
Compare hashes
Match
No match
Recover original cheque image
Tampering detected

Fig. 2. Encryption and Decryption Process: Original cheque is split into Share 1 (random) and Share 2 (original XOR Share 1); shares are recombined using XOR and verified with hashing.

C. *SHA-256 Integrity Check*

SHA-256 is used to verify the integrity of the reconstructed cheque, as shown in Fig. 3. The process is as follows:

1) Compute SHA-256 hash of the original cheque image.
2) Store the hash in the database along with Share 2.
3) Reconstruct the cheque by XORing Share 1 and Share 2.
4) Compute SHA-256 hash of the reconstructed cheque.
5) Compare the reconstructed hash with the stored hash:
    - If hashes match, integrity is verified.
    - If hashes differ, tampering is detected.

The SHA-256 algorithm plays a critical role in ensuring tamper detection by generating a unique 256-bit hash for each cheque image, as illustrated in Fig. 4. This figure provides a visual representation of the SHA-256 process, where the input cheque image is first converted into binary data. The binary data is then divided into 512-bit blocks, with padding added if necessary to ensure consistent block sizes. Each block is expanded into a message schedule of 64 words, which are processed through 64 rounds of transformations involving bitwise operations, modular additions, and logical functions. The final output is a 256-bit hash (represented as 64 hexadecimal characters), which serves as a digital fingerprint of the cheque image. During verification, any alteration to the cheque image—whether through tampering with Share 1, Share 2, or the reconstructed image—results in a different hash value, allowing the system to detect unauthorized modifications reliably.

The SHA-256 algorithm is detailed in Algorithm 1.

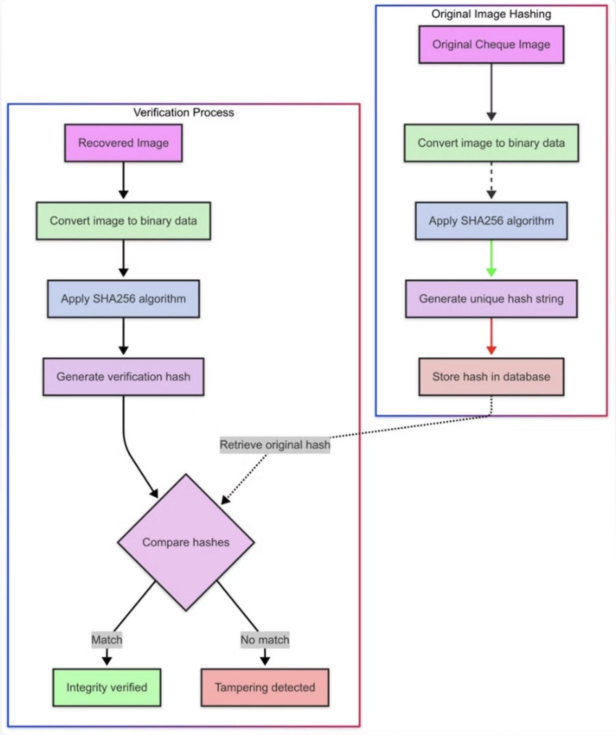


Fig. 3. Hashing for Integrity: Original cheque is hashed using SHA-256, stored in database; reconstructed cheque is hashed and compared to detect tampering.

Algorithm 1 SHA-256 Hashing Algorithm

```
**Input:** Cheque image $I$
**Output:** 256-bit hash $H$
Convert $I$ to binary data
Split binary data into 512-bit blocks
Expand each block into 64 words (message schedule)
Process each block through 64 rounds
Produce 256-bit hash $H$ (64 hex characters)
**return** $H$
```

D. *Two-Factor Authentication*

Banker authentication employs a multi-layered approach:

1) **Username/Password Authentication**: Basic credential verification against the secure MySQL database.
2) **Duo Push Authentication**: Upon successful credential verification, the system initiates a Duo push notification to the banker's registered device, requiring explicit approval before granting access to the verification interface [10].

This two-factor approach ensures that even if banker credentials are compromised, an attacker cannot access the verification system without physical possession of the banker's authenticated device.

E. *Secure Communication*

All communications between system components implement security best practices:

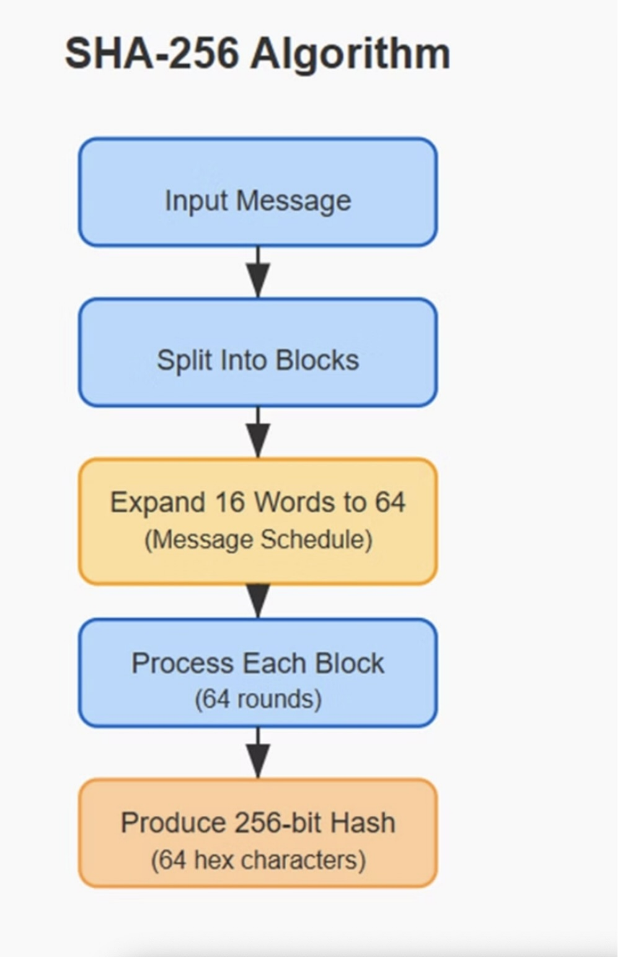


Fig. 4. SHA-256 Algorithm: Input cheque image is processed through SHA-256 to produce a 256-bit hash, enabling tamper detection.

1) **Email Transmission**: Share 1 is transmitted to the banker via email with a secure download link rather than as a direct attachment, reducing the risk of unauthorized access through email interception.
2) **Database Encryption**: Share 2 is stored in an encrypted MySQL database, with secure connection parameters managed through environment variables rather than hardcoded credentials.
3) **Integrity Verification**: During reconstruction, the system verifies the SHA-256 hash of the combined image against the stored hash, providing an additional integrity check beyond visual inspection.

IV. Implementation

A. Development Environment

The system was implemented using Python with the following key libraries:

- Streamlit for the web interface.
- OpenCV (`cv2`) for image processing.
- MySQL Connector for database interactions.
- Duo Client SDK for two-factor authentication.
- Email libraries (`smtplib`, `email`) for secure share transmission.

B. Key Components

The implementation consists of three primary modules:

1) **Application Logic (`app.py`)**: Manages the user interface, role-based access control, and workflow orchestration between customer and banker operations.
2) **Cryptography Module (`cryptography.py`)**: Implements the visual cryptography algorithms, including share generation, secure overlay, SHA-256 hashing, and email transmission.
3) **Database Interface (`db.py`)**: Handles secure storage and retrieval of cheque data, shares, and banker authentication information.

C. User Interface Design

The system provides distinct interfaces for customers and bankers:

1) **Customer Interface**: Features a straightforward upload form with fields for cheque number, usage message, and banker email. The interface provides clear feedback on share generation and transmission status.
2) **Banker Interface**: Implements a secure two-step login process (username/password + Duo push) followed by a verification panel where bankers can enter cheque numbers and upload their share (Share 1) to verify against the system-stored share (Share 2).

D. Security Measures

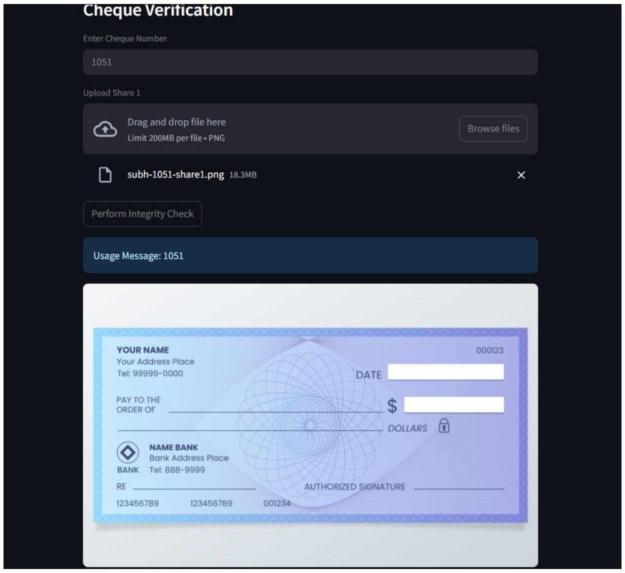
Several additional security measures are implemented:

1) **Environment-Based Configuration**: Sensitive parameters (database credentials, email credentials, Duo API keys) are managed through environment variables rather than hardcoded values.
2) **Temporary File Management**: All locally generated share files are created in secure temporary locations and immediately removed after use to prevent unauthorized access.
3) **Share Transmission Security**: Share 1 is delivered to bankers via a download link rather than as a direct email attachment, providing an additional layer of protection.
4) **Input Validation**: All user inputs are validated before processing to prevent SQL injection and other common attacks.

E. System Outputs

To demonstrate the system's functionality, we present the final output from the banker's verification interface. This output showcases the successful reconstruction and verification of the cheque after the banker uploads Share 1 and the system retrieves Share 2 from the database.

Fig. 5 illustrates the banker's verification interface after a successful integrity check. The interface displays the cheque number (1051), the uploaded Share 1 file (`subh-1051-share1.png`), the usage message, and the



Cheque Verification

Enter Cheque Number: 1051

Upload Share 1

subh-1051-share1.png 18.3MB

Usage Message: 1051

Fig. 5. Final Verification Output: Screenshot of the banker's interface in Streamlit, showing the cheque number input, uploaded Share 1, usage message, and the reconstructed cheque after integrity verification using SHA-256 hashing.

reconstructed cheque image. The cheque is reconstructed by XORing Share 1 (provided by the banker) with Share 2 (retrieved from the database), and its integrity is verified by comparing the SHA-256 hash of the reconstructed image with the stored hash. This output confirms that the cheque has not been tampered with, as indicated by the successful hash match.

V. Experiments and Evaluation

A. Experimental Setup

The system was evaluated using a test set of 50 cheque images (1 MB each, 1000x1000 pixels) under various scenarios on a macOS system with a 2.3 GHz Intel Core i5 processor and 8 GB RAM:

1) **Valid Cheques**: Unmodified cheques processed through the complete workflow.
2) **Tampered Share 1**: Scenarios where the banker's share was modified.
3) **Tampered Share 2**: Scenarios where the database-stored share was modified.
4) **Tampered Both Shares**: Scenarios where both shares were modified in complementary ways to attempt to produce a valid but altered cheque.

B. Security Evaluation

The system demonstrated strong security characteristics across multiple dimensions:

1) **Tamper Detection**: 100% detection rate for modifications to either share, with hash verification failures detecting all tampering attempts.
2) **Authentication Robustness**: No successful unauthorized access when testing compromised credentials without the second factor (Duo push acceptance).
3) **Share Confidentiality**: Individual share analysis confirmed that no meaningful information about the original cheque could be extracted from either share in isolation.

C. Performance Metrics

Performance testing revealed the following metrics:

1) **Share Generation Time**: Average of 0.62 seconds to generate both complementary shares from an original cheque image.
2) **Verification Time**: Average of 0.87 seconds to overlay shares and verify integrity once both shares are available.
3) **Email Delivery Time**: Average of 4.35 seconds for successful delivery of Share 1 to the banker's email.
4) **Authentication Time**: Average of 6.21 seconds for complete two-factor authentication (including Duo push response time).

D. Usability Assessment

User testing with a small group of banking professionals and customers ($n = 12$) yielded positive feedback:

1) **Customer Experience**: 91% rated the upload and submission process as "easy" or "very easy" to use.
2) **Banker Experience**: 83% rated the verification process as "efficient" and "more secure" compared to traditional verification methods.
3) **Learning Curve**: New users required an average of less than 5 minutes to become proficient with the system.

VI. Conclusion and Future Work

This paper presented a secure bank cheque verification system that leverages XOR-based visual cryptography, SHA-256 hashing, and Duo two-factor authentication to address critical security challenges in traditional cheque processing. By splitting cheque images into complementary shares distributed between bankers and a secure database, our approach creates a distributed security model that requires collaboration between authorized parties for verification. The integration of Duo push authentication provides an additional security layer that protects against credential compromise.

Experimental evaluation demonstrated the system's effectiveness in detecting tampering and unauthorized access attempts while maintaining usability for both customers and banking staff. The implementation achieves a balance between security and user experience, making it practical for real-world deployment.

Future work will focus on several enhancements:

1) **Mobile Integration**: Developing mobile applications for both customer and banker interfaces to increase accessibility.
2) **Automated Anomaly Detection**: Implementing machine learning algorithms to detect suspicious patterns in cheque images before share generation.
3) **Blockchain Integration**: Exploring the use of blockchain technology to create an immutable audit trail of cheque verifications.

4) **Enhanced Reconstruction**: Implementing more sophisticated image reconstruction techniques that can highlight potential tampering areas.
5) **Scalability Testing**: Evaluating system performance under high-volume scenarios typical of large banking institutions.
6) **Naor-Shamir Visual Cryptography**: Implementing a proper (2,2) scheme for physical overlay, enabling verification without digital reconstruction.
7) **Share Encryption**: Encrypting shares before storage to enhance security.

References

[1] Association for Financial Professionals, "2021 AFP Payments Fraud and Control Survey," 2021.

[2] S. Bhattacharyya, I. Jha, P. Sagdeo, and A. Bhattacharyya, "Bank Cheque Verification and Forgery Detection: A Survey," *IEEE Access*, vol. 8, pp. 193668–193684, 2020.

[3] M. Naor and A. Shamir, "Visual Cryptography," *Advances in Cryptology — EUROCRYPT'94*, pp. 1–12, 1995.

[4] A. Ross and A. A. Othman, "Visual Cryptography for Biometric Privacy," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 70–81, 2011.

[5] F. Liu and C. Wu, "Embedded Extended Visual Cryptography Schemes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 307–322, 2011.

[6] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[7] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[8] D. Dagon *et al.*, "Account Takeover: Automated Web Authentication Compromise," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 34–43, 2020.

[9] S. K. Jami, S. R. Chalamala, and A. K. Jindal, "Biometric Template Protection Through Adversarial Learning," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 316–325, 2019.

[10] T. Ahmed, D. Goel, and H. Rouselakis, "Duo Authentication: A Two-Factor Framework for Web Services," *IEEE Security & Privacy*, vol. 16, no. 4, pp. 36–45, 2018.