A Novel Steganographic Approach To Strengthen Enhanced MFA And Attack Prevention For Credential Transmission

Mrs. Banuppriya P[1], Bharathiraja S[2], Jeyachandran R[3], Pradeep Raj S[4], Rakesh R[5]
[1]Assist prof, Dept of Computer Science and Engineering
[2, 3, 4, 5] Dept of Computer Science and Engineering
[1, 2, 3, 4, 5] Mahendra Institute of Engineering and Technology, Namakkal, Tamil Nadu, India

Abstract- *Digital banking infrastructure faces escalating and sophisticated threats including phishing, Man-in-the-Middle (MITM) interceptions, session hijacking, replay attacks, credential stuffing, and denial-of-service (DoS) campaigns. Conventional single-factor authentication mechanisms based on username-password pairs offer insufficient protection, while existing Multi-Factor Authentication (MFA) implementations—such as SMS-based One-Time Passwords (OTP), hardware tokens, and basic biometric checks—continue to exhibit exploitable vulnerabilities. This paper proposes a novel five-layer secure authentication and transaction authorization framework tailored to digital banking environments. The system integrates: (i) Grassmann manifold-based facial recognition for biometric enrollment and live verification, replacing hardware USB tokens with a mathematically robust biometric factor; (ii) multi-factor login combining credential-based authentication with biometric matching; (iii) dynamic per-session cryptographic key generation using SHA-512 with user-specific salts; (iv) QR-code Least Significant Bit (LSB) steganography for covert session key transmission to the user's registered email, hiding sensitive token data within an innocuous carrier image; and (v) per-transaction session key validation with real-time unauthorized-access alerting. The proposed architecture extends and improves upon the Secure Multi-Factor Authentication (SMFA) framework by Sarower et al. [1] by eliminating physical device dependency, adding biometric security, and introducing a banking-domain-specific steganographic session key channel. Security analysis via Burrows-Abadi-Needham (BAN) logic demonstrates protocol correctness. The facial recognition module achieves a 97.3% True Acceptance Rate (TAR) with a False Acceptance Rate (FAR) below 0.8%. Steganographic embedding achieves a PSNR of 43.2 dB, well above the 40 dB imperceptibility threshold. Total authentication pipeline latency is approximately 2.3 seconds on standard hardware.*

Keywords: Biometric authentication, banking security, Grassmann algorithm, LSB steganography, multi-factor authentication, QR code steganography, session key management, SHA-512.

I. INTRODUCTION

In contemporary digital banking environments, the security of user authentication and transaction authorization has emerged as one of the most critical challenges in cybersecurity. The accelerating adoption of internet and mobile banking platforms has made financial systems prime targets for a broad spectrum of adversarial activities. Attackers routinely exploit weaknesses in authentication mechanisms to gain unauthorized access to accounts, intercept financial transactions, and exfiltrate sensitive personal and financial data [2].

Traditional authentication systems in banking predominantly rely on the single-factor paradigm: a username combined with a static password. While conceptually simple, this approach is fundamentally vulnerable. Passwords can be compromised through brute-force enumeration, dictionary attacks, credential-stuffing campaigns using leaked credential databases, phishing websites that mimic legitimate banking portals, and social engineering. Morris and Thompson's seminal analysis [8] established password vulnerability as a structural property rather than an implementation deficiency. More than four decades later, large-scale data breaches continue to expose hundreds of millions of password-based credentials annually.

Multi-Factor Authentication (MFA) was introduced as a systematic countermeasure, combining two or more independent verification factors drawn from three categories: (1) something the user knows—a password, PIN, or security answer; (2) something the user has—a hardware token, smart card, mobile device, or OTP; and (3) something the user is—a biometric characteristic such as fingerprint, iris, or facial geometry [1]. By requiring multiple independent factors, MFA substantially raises the cost and complexity of unauthorized access.

However, first-generation MFA deployments in banking have introduced their own vulnerabilities. SMS-based OTP delivery is susceptible to SIM-swapping attacks, in which an adversary socially engineers a mobile carrier into reassigning a victim's phone number to an attacker-controlled SIM card, thereby intercepting all SMS-delivered tokens. The SS7 signaling protocol, which underlies SMS routing, contains well-documented security gaps exploitable for OTP interception. Email-delivered OTPs face analogous risks if the user's email account is compromised. Hardware security keys such as FIDO U2F tokens address many of these weaknesses but introduce device management overhead, loss and theft risks, and usability challenges in mobile banking contexts [1].

Session management represents a further attack surface. Even when authentication credentials are correctly protected, the session tokens generated upon login can be hijacked through cross-site scripting (XSS), cross-site request forgery (CSRF), network-level interception, or endpoint compromise. If a valid session token is obtained by an adversary, they gain full access to all capabilities available within that session without needing to replay the authentication process.

Steganography—the ancient art of concealing the existence of a secret message within an innocuous carrier medium—offers a powerful supplementary security mechanism. Unlike cryptography, which encrypts message content while leaving the existence of a communication visible to an observer, steganography conceals the very existence of the secret data. In digital systems, steganographic techniques embed secret information within the pixel-level channels of images, audio, or video files. The Least Significant Bit (LSB) substitution technique is the most widely deployed, replacing the least significant bits of pixel intensity values with secret data bits in a manner that produces imperceptible changes in the visible appearance of the image. A cover image with PSNR greater than 40 dB relative to the corresponding stego image is considered imperceptible to human visual inspection [1].

The SMFA framework introduced by Sarower et al. [1] demonstrated the effectiveness of combining steganography with MFA by using a USB device as a physical authentication token and embedding One-Time Tokens (OTTs) within cover images before transmission to an authentication server. SMFA validated its security using BAN logic and showed resistance to replay, MITM, impersonation, offline dictionary, and DoS attacks. However, SMFA did not incorporate biometric verification, retained hardware USB dependency, and was not tailored to banking transaction workflows.

Biometric facial recognition has emerged as a convenient and increasingly accurate authentication factor, particularly in mobile banking applications. The accuracy of facial recognition is critically dependent on the mathematical framework used for feature extraction and representation. Shallow feature matching techniques that compare pixel-level or simple gradient-based facial descriptors are vulnerable to spoofing using photographs, video playback, or 3D-printed masks. The Grassmann manifold-based algorithm addresses this limitation by representing facial feature vectors as points on a Riemannian manifold of linear subspaces, where geodesic distances between manifold points provide rotation-invariant, illumination-robust similarity metrics [4].

This paper makes the following contributions to the field of secure banking authentication:

- A Grassmann algorithm-based facial biometric enrollment and verification module that replaces hardware USB tokens, providing a stronger, hardware-independent biometric factor.
- A multi-factor login pipeline combining username/password credential verification with live biometric facial matching, both required for authentication success.
- A dynamic per-session key generation scheme using SHA-512 with user-specific salts, ensuring each session token is unique, cryptographically strong, and non-replayable.
- A QR-code LSB steganography module that embeds session keys within QR code images before email delivery, concealing sensitive token data from passive network observers.
- A per-transaction session key validation mechanism with real-time unauthorized-access alerting that notifies the legitimate account holder immediately upon detection of suspicious activity.
- A formal security analysis using BAN logic and a comparative evaluation against the SMFA base protocol and conventional banking authentication schemes.

II. LITERATURE SURVEY

A. Foundations of Multi-Factor Authentication

The foundational challenge of authentication—verifying that a party claiming an identity genuinely possesses has been studied since the earliest days of networked computing. Morris and Thompson [8] established that password-based authentication is structurally vulnerable to offline dictionary attacks. Bonneau et al. conducted a systematic comparison of web authentication schemes,

evaluating each along dimensions of security, deploy ability, and usability, concluding that no single scheme optimally satisfies all dimensions and that MFA schemes represent the most promising direction for high-security applications.

Das [9] proposed a three-factor authentication mechanism for distributed wireless sensor networks combining password, smart card, and biometric factors, claiming resilience against common attacks. Babu et al. subsequently identified vulnerabilities in Das's protocol—specifically susceptibility to replay attacks and session-specific temporary information leakage—and proposed a hardened variant. These iterative improvements illustrate the adversarial nature of authentication protocol design, where each published scheme undergoes scrutiny and often exposes new attack surfaces.

Amin et al. proposed a two-factor RSA-based authentication system for multi-server environments combining password and smart card, validated through BAN logic for session key freshness and mutual authentication. Cho et al. extended this line of work with a three-factor mutual authentication protocol for e-governance systems operating in multi-server environments, verified using informal security analysis, BAN logic, the ROR model, and simulation via the AVISPA tool. These formal verification approaches establish a methodological standard that the present work also adopts.

B. Steganography in Authentication Systems

The SMFA framework by Sarower et al. [1] represents the most directly relevant prior work. SMFA combines steganography, asymmetric RSA encryption, SHA-512 hashing, and a USB device authentication token to create a multi-server MFA protocol resistant to five major attack classes. The authentication process in SMFA proceeds through three phases: the request phase, in which the USB device generates an OTT and the client embeds hashed credentials within a steganographic cover image selected based on the OTT value; the authentication phase, in which the authentication server extracts and verifies the embedded credentials; and the response phase, in which the application server creates a session and grants access. BAN logic analysis formally validates that the authentication server correctly establishes belief in the user's identity through the message exchange protocol.

The computational cost of SMFA was measured at 7.41 ms on an Intel Core i3-10105 at 3.70 GHz with 8 GB RAM, expressed as 7T_h + 12T_sto + 2T_se where T_h denotes one-way hash operations, T_sto steganographic operations, and T_se symmetric encryption operations.

Comparative analysis against eleven existing protocols demonstrated that SMFA achieves the best overall security coverage despite its slightly elevated computational cost arising from steganographic operations.

Madhuravani et al. explored dynamic hashing combined with steganography for web authentication, demonstrating that steganographic embedding substantially increases the computational complexity faced by adversaries attempting to decode intercepted credentials. Gunawardena et al. described the imgAuth framework using picture steganography for user profile management and authentication. Ihmaidi et al. proposed a method for securing online shopping portals using steganography and biometrics to conceal login credentials within images. Danuputri et al. introduced the ste-Chy prototype combining Vigenère cipher encryption with LSB steganography for private authentication on Android, validated with SHA-256 integrity checks.

C. Biometric Authentication in Banking

Machap [4] analyzed the integration of facial recognition as an additional security layer in mobile banking systems, demonstrating measurable reduction in unauthorized access incidents. The study highlighted that recognition accuracy is critically dependent on the feature extraction algorithm, with mathematically grounded representations outperforming shallow descriptors. Khan et al. developed a secure mutual authentication system for smart grid communications using biometric-based elliptic curve cryptography, demonstrating that biometric factors can be integrated into cryptographic protocols without prohibitive computational overhead.

Masud et al. devised a lightweight user authentication mechanism for IoT-based healthcare that preserves user anonymity, demonstrating that biometric factors can be incorporated into resource-constrained environments. Hossain and Raza [5] studied the effectiveness of MFA in preventing unauthorized banking access, concluding that systems combining biometrics with cryptographic session management achieve the highest resistance to social-engineering-based attacks. Karim et al. [2] conducted a systematic literature review of online banking authentication methods, identifying biometric integration combined with robust session management as the primary gap in current deployments.

D. Session Security and Alert Mechanisms

Standard session management in web banking applications employs session identifiers generated upon login and transmitted in cookies or HTTP headers. These tokens, if

intercepted, enable session hijacking attacks in which an adversary assumes the authenticated identity of a legitimate user without knowledge of their credentials. Huang and Li identified session hijacking as one of the primary vulnerabilities in the AKA protocol used in UMTS mobile networks, a finding with direct relevance to mobile banking security.

Cavus et al. [3] recommended real-time anomaly detection and immediate user notification as essential components of a robust mobile banking security architecture, noting that users who receive timely alerts can take immediate remedial action—such as invalidating sessions and changing credentials—before significant financial harm occurs.

Ahmed et al. [6] proposed blockchain-based transaction audit trails as a tamper-proof supplement to standard session management, providing post-hoc evidence for fraud investigation even when real-time prevention fails.

The gap identified across this body of literature is the absence of a unified system that combines Grassmann-based biometric verification, steganographic session key transmission, and real-time alerting within a coherent banking authentication architecture. The proposed system addresses precisely this gap.

III. EXISTING SYSTEM

A. Current Banking Authentication Architectures

Contemporary digital banking authentication systems are architecturally heterogeneous, reflecting the incremental adoption of security measures over decades. The predominant pattern combines password-based first-factor authentication with a second factor delivered through an out-of-band channel, most commonly SMS OTP.

Some institutions have deployed hardware tokens such as RSA SecurID or FIDO U2F keys as the second factor for high-net-worth or corporate clients. Mobile banking applications frequently supplement these with biometric authentication using device-native APIs—fingerprint scanners or basic facial recognition built into iOS Face ID or Android biometric frameworks.

Session management in banking web applications typically follows the OWASP-recommended pattern: a cryptographically random session token of at least 128 bits is generated upon successful authentication, stored server-side with an associated expiry time, and transmitted to the client in a secure, HttpOnly cookie over TLS. Subsequent requests present this token for server-side validation. Session tokens are invalidated upon logout or after a configurable inactivity timeout.

B. Technical Limitations of Existing Systems

Despite these measures, several structural vulnerabilities persist in mainstream banking authentication:

Password Factor Weaknesses: Static passwords remain the primary authentication factor in most banking systems. Users frequently reuse passwords across multiple services, creating credential-stuffing vulnerabilities whenever any service in their ecosystem suffers a breach. Even when banks enforce strong password policies, phishing attacks can harvest credentials directly from users who are deceived into entering their passwords on attacker-controlled imitation portals.

SMS OTP Vulnerabilities: The SS7 signaling protocol, which underlies the global telephony infrastructure, contains design-era security assumptions that are no longer valid in the current threat landscape. SS7 exploitation allows adversaries to intercept SMS messages in transit, effectively bypassing SMS-based OTP without the user's knowledge. SIM-swapping attacks, in which an adversary convinces a mobile carrier to port a victim's number to a new SIM card, are increasingly common and have been used in high-profile cryptocurrency and banking heists.

Basic Biometric Vulnerabilities: Device-native facial recognition systems, while convenient, have been demonstrated to be vulnerable to photograph-based spoofing on older devices and video-based spoofing on systems without liveness detection. The feature extraction algorithms underlying these systems are typically optimized for speed and device-level resource constraints rather than cryptographic-strength accuracy.

Session Management Weaknesses: Even a perfectly executed authentication process does not guarantee session security. Session tokens transmitted in cookies are exposed to JavaScript-accessible theft through XSS vulnerabilities. Long session durations, which banks often permit for usability, extend the window of exposure if a token is compromised. Standard session tokens do not incorporate transaction-specific authorization, meaning a hijacked session grants full access to all transaction capabilities.

Absence of Covert Channel Protection: No mainstream banking authentication system employs steganographic techniques to conceal the existence of authentication tokens. All second-factor deliveries—whether

OTP SMS, email OTP, or push notification—are transmitted in a form that makes the existence of a secret value visible to network observers, enabling targeted interception and real-time phishing relay attacks.

Absence of Real-Time Alerting: While most banks implement fraud detection systems that flag unusual transaction patterns after the fact, very few provide immediate real-time authentication-level alerts that notify users the moment an unauthorized login attempt is detected. The gap between unauthorized access and user notification allows adversaries a window to complete fraudulent transactions before the legitimate account holder becomes aware.

IV. PROPOSED SYSTEM

A. System Overview and Design Principles

The proposed system is designed around three core principles: defense in depth, credential concealment through steganography, and real-time incident response. Defense in depth ensures that the compromise of any single authentication component does not result in full system compromise. Credential concealment through steganography ensures that even passive network interception of transmitted authentication data does not reveal usable secret values. Real-time incident response ensures that legitimate users are informed immediately of suspicious activity, enabling rapid remediation.

The system architecture comprises four primary entities: the User Client (UC)—the user's browser or mobile application; the Authentication Server (AS)—responsible for credential verification, biometric matching, session key generation, and steganographic embedding; the Application/Banking Server (BS)—responsible for transaction processing and session key validation; and a secure Database (DB) storing facial templates, hashed credentials, session key records, and audit logs.

B. Architectural Components and Data Flow

The authentication and transaction authorization pipeline proceeds through five sequential phases as illustrated in the system architecture:

Phase 1 – Registration: The user accesses the registration portal and provides their full name, username, email address, and password. The system captures one or more facial images using the device camera. The Grassmann algorithm processes these images to generate a facial template—a compact geometric representation of the user's facial feature subspace on the Grassmann manifold. The password is hashed using SHA-256 before storage. All data is encrypted at rest in the secure database.

Phase 2 – Multi-Factor Login: The user submits their username and password. The AS verifies the credential pair against the stored hashed password. Upon successful credential verification, the AS activates the biometric verification stage: the user is prompted to capture a live facial image, which is processed through the Grassmann algorithm to generate a live facial template. The geodesic distance between the live template and the stored enrollment template is computed; if this distance falls below a calibrated matching threshold, biometric verification succeeds. Access is denied and an alert is triggered if either verification stage fails.

Phase 3 – Session Key Generation: Upon successful multi-factor authentication, the AS generates a unique session key K_session = SHA-512(UID || timestamp || server_salt || nonce), where UID is the user identifier, timestamp is the current Unix epoch time in milliseconds, server_salt is a per-user secret salt stored in the AS, and nonce is a cryptographically random value generated fresh for each session. This construction ensures that session keys are unique, temporally bounded, and non-deterministic even for a known UID.

Phase 4 – QR Code Steganography and Delivery: The session key is encoded into a byte array and embedded within a QR code image using the LSB substitution steganography technique. A standard QR code is first generated encoding a publicly-visible innocuous value (e.g., a session reference ID). The session key bits are then embedded into the LSBs of the QR code image's pixel channels. The resulting stego QR code is visually indistinguishable from the original, with PSNR maintained above 40 dB. The stego QR code image is transmitted to the user's registered email address via an encrypted SMTP connection using TLS 1.3.

Phase 5 – Transaction Verification and Alerting: When the user initiates a banking transaction, the mobile or web application prompts them to scan the received stego QR code. The application decodes the LSB layer to extract the embedded session key and submits it to the BS along with the transaction request. The BS validates the submitted key against the AS-stored session record, verifying both key correctness and session validity window. If validation succeeds, the transaction is authorized. If validation fails—due to an incorrect key, session expiry, or suspected replay—the transaction is blocked, the session is invalidated, and an immediate alert notification is dispatched to the user.

C. Improvements Over Base SMFA Protocol

The SMFA framework of Sarower et al. [1] constitutes the theoretical and methodological foundation of the proposed system. The present work extends SMFA in five substantive dimensions:

Biometric Factor Replacement: SMFA employs a physical USB device containing a Stego validator that generates OTTs as the primary possession factor. The proposed system replaces this hardware dependency with Grassmann algorithm-based facial biometrics. This substitution eliminates device loss, theft, and manufacturing cost concerns while introducing a biometric factor that is inherently tied to the user's person and cannot be transferred or replicated.

Steganographic Medium: SMFA embeds credentials within arbitrary cover images selected from a collection based on the OTT value. The proposed system uses QR codes as the steganographic medium, providing a structured carrier that simultaneously encodes a machine-readable session reference while concealing the actual session key in pixel-level channels. This dual-encoding approach enhances the deception of adversaries who would otherwise identify a suspicious image as a credential carrier.

Domain-Specific Transaction Authorization: SMFA operates as a general-purpose MFA protocol and does not include per-transaction authorization. The proposed system adds a transaction-level verification layer in which each banking transaction must be accompanied by the session key extracted from the stego QR code. This means that even a successfully hijacked session token does not grant the ability to execute transactions without concurrent possession of the decoded session key.

Real-Time Alert Mechanism: SMFA does not include a user notification mechanism for failed authentication attempts. The proposed system dispatches immediate email and push notifications to the legitimate user upon detection of any failed biometric verification, failed session key validation, or unusual transaction pattern, enabling immediate incident response.

Computational Architecture: SMFA employs RSA asymmetric encryption for OTT protection. The proposed system uses SHA-512 for session key generation—a one-way operation—and TLS 1.3 for channel security, reducing the computational overhead of asymmetric operations while maintaining equivalent security strength through the use of modern authenticated encryption primitives.

V. METHODOLOGY AND MODULES

Module 1: User Registration with Grassmann Biometrics

The registration module is the foundation of the biometric authentication pipeline. During registration, the user provides their personal details—full name, email address, and a chosen username and password. The password is immediately hashed using SHA-256 before any storage operation, following the principle of never persisting plaintext credentials.

The biometric enrollment process begins with the capture of three to five facial images of the user under varying slight pose and lighting conditions using the device camera, ensuring that the enrolled template encompasses natural appearance variation. Each captured image is pre-processed: the face region is detected and cropped using a Haar cascade classifier implemented in OpenCV; the cropped region is resized to a standardized 128×128 pixel frame; histogram equalization is applied to normalize illumination.

The pre-processed facial images are then processed through the Grassmann algorithm. The Grassmann manifold Gr(p, n) is the set of all p-dimensional linear subspaces of R^n. Facial feature vectors extracted using Principal Component Analysis (PCA) from each image are stacked to form a matrix whose column space defines a point on the Grassmann manifold. The enrollment template is computed as the Karcher mean of the manifold points corresponding to the multiple enrolled images, producing a single representative point that minimizes geodesic distance to all enrollment samples. This template is stored in the database alongside the user's encrypted personal details.

The Grassmann representation offers several advantages over Euclidean feature spaces for facial recognition: it is invariant to linear transformations of the feature basis, making it robust to illumination changes that alter the scale and orientation of feature vectors; it captures the geometric structure of facial appearance variation rather than treating it as noise; and geodesic distance on the manifold is a more meaningful similarity metric for subspace-represented features than Euclidean distance in the ambient space.

Module 2: Multi-Factor Login and Authentication

The login module implements a two-stage sequential verification protocol. In Stage 1, the user submits their username and password through a standard login form over HTTPS. The AS retrieves the stored SHA-256 hash for the given username and computes the hash of the submitted

password, comparing the two hashes using a constant-time equality function to prevent timing-based side-channel attacks. If the hashes match, Stage 1 succeeds. If they do not match, the login attempt is logged, a failed-attempt counter is incremented, and after three consecutive failures the account is temporarily locked and an alert is sent to the registered email address.

Stage 2 activates upon Stage 1 success. The AS sends a biometric verification challenge to the UC, prompting the user to capture a live facial image. The live image undergoes the same pre-processing pipeline as during enrollment. A Grassmann representation is computed from the live image. The AS retrieves the stored enrollment template and computes the geodesic distance d_geo between the live and enrollment manifold points. If d_geo $\leq \tau$ (where τ is a pre-calibrated threshold balancing TAR and FAR), biometric verification succeeds. The threshold τ is set to achieve TAR $\geq$ 97% and FAR $\leq$ 1% based on empirical calibration on the training dataset. If biometric verification fails, the attempt is logged, an alert notification is dispatched to the user, and access is denied.

Module 3: Session Key Generation

The session key generation module executes immediately following successful multi-factor authentication. The key generation function is: K_session = SHA-512(UID || T_ms || S_user || N_rand), where UID is the authenticated user's identifier (an internal numeric ID, not the username); T_ms is the current timestamp in milliseconds since Unix epoch; S_user is a 256-bit per-user secret salt generated during registration and stored in the AS database; and N_rand is a 128-bit cryptographically random nonce generated fresh for each session using the OS-provided cryptographically secure pseudo-random number generator (CSPRNG).

The resulting SHA-512 digest (512 bits = 64 bytes) constitutes the session key. This construction provides the following security properties: uniqueness across sessions for the same user (ensured by the timestamp and nonce combination); resistance to prediction even with knowledge of the UID and timestamp (ensured by the unknown per-user salt); non-replayability (ensured by the nonce); and one-wayness (ensured by the SHA-512 preimage resistance). The session key record stored in the AS database includes the key itself, the associated UID, the session creation timestamp, and the session expiry timestamp (configurable, default 30 minutes).

Module 4: QR Code Steganography

The QR code steganography module is responsible for covertly embedding the session key within a QR code image and delivering it to the user via email. The module operates in two sub-stages: QR code generation and LSB embedding.

In the QR code generation sub-stage, a standard QR code is generated encoding a session reference string—a short, publicly-innocuous identifier such as 'SESSION-XXXXXX'—using the qrcode Python library at error correction level H (approximately 30% of data capacity reserved for error recovery). The resulting QR code image is saved as a lossless PNG file at a resolution sufficient for reliable camera scanning.

In the LSB embedding sub-stage, the 64-byte (512-bit) session key is serialized into a bit stream. The PNG image is converted to a NumPy array of pixel intensity values. The bit stream is embedded into the LSBs of the red channel of the image pixels, proceeding in raster order (left to right, top to bottom). For a 200×200 pixel QR code image, there are 40,000 pixels available in the red channel alone, providing capacity for embedding 40,000 bits—far exceeding the 512-bit session key. The proportion of LSBs modified is therefore extremely small, and the visual impact is correspondingly negligible.

The PSNR between the original and stego QR code images is computed as: PSNR = $10 \cdot \log10(MAX_I^2 / MSE)$, where MAX_I is the maximum pixel intensity (255 for 8-bit images) and MSE is the mean squared error between corresponding pixels of the original and stego images. For the described embedding density, the PSNR consistently exceeds 43 dB, well above the 40 dB imperceptibility threshold established in [1]. The stego QR code is then attached to an email message sent to the user's registered email address via SMTPLIB over a TLS 1.3-encrypted connection.

Upon receiving the email, the user opens it in the banking application or the system-provided QR scanning interface. The application captures the stego QR code using the device camera, reconstructs the PNG from the camera image, and applies the inverse LSB extraction procedure to recover the session key bit stream. The recovered session key is held in application memory (never written to persistent storage) for submission during transaction authorization.

Module 5: Transaction Verification and Real-Time Alert

The transaction verification module provides the final layer of security, ensuring that successful authentication does not in itself authorize financial transactions. Each transaction request submitted by the user must be accompanied by the

session key extracted from the stego QR code. This requirement transforms the session key from a login credential into a per-transaction authorization token, substantially raising the bar for adversaries who may have obtained a session identifier through network-level interception.

When the user initiates a transaction (e.g., fund transfer, bill payment), the application automatically extracts the session key from the stego QR code held in memory and includes it in the transaction request sent to the BS over HTTPS. The BS forwards the session key to the AS for validation. The AS checks: (1) that the submitted key matches the stored session key for the authenticated user; (2) that the session has not expired; and (3) that the session key has not been used beyond a configurable maximum number of transactions. If all checks pass, the BS processes the transaction and returns a confirmation. If any check fails, the transaction is rejected, the session is invalidated, a detailed incident record is written to the audit log, and an immediate alert notification is sent to the user's registered email and mobile number.

The alert notification includes the timestamp of the failed attempt, the IP address and device fingerprint of the requester, and a direct link to initiate emergency account lockdown. This real-time response capability enables users to take protective action within seconds of a suspicious event, minimizing the window of potential financial harm.

VI. SECURITY ANALYSIS

A. BAN Logic Verification

Burrows-Abadi-Needham (BAN) logic is a formal framework for analyzing the security properties of authentication protocols. It operates through a set of inference rules applied to an idealized model of the protocol message sequence, enabling the derivation of formal beliefs that each principal holds about keys, timestamps, and the identities of other principals. BAN logic's three primary rules are: (i) the Message Meaning Rule, which states that if principal P shares a key K with Q and receives a message encrypted under K, P believes Q said that message; (ii) the Nonce Verification Rule, which states that if P believes a value X is fresh and believes Q said X, then P believes Q believes X; and (iii) the Jurisdiction Rule, which states that if P believes Q has jurisdiction over X and believes Q believes X, then P believes X.

For the proposed protocol, the following initial assumptions are established: (A1) AS believes AS↔K_u U; the AS and user share an authenticated channel. (A2) AS believes fresh(T_ms); the AS considers the current timestamp fresh. (A3) BS believes BS↔K_au AS; the BS and AS share an authenticated channel. (A4) BS believes fresh(T_au); the BS considers the authentication server's timestamp fresh. (A5) U believes fresh(K_session); the user considers the session key fresh upon receipt.

Message 1: U → AS: {T_ms, UID, H(password), FT_live, K_session_request} K_u. The user sends their timestamp, UID, hashed password, live facial template, and a session key request, encrypted under the shared user-AS key. Applying the Message Meaning Rule, AS believes U said (UID, H(password), FT_live). Applying Nonce Verification with A2, AS believes U believes (UID, H(password), FT_live). Since H(password) matches the stored hash and $d\_geo(FT\_live, FT\_enrolled) \leq \tau$, AS verifies the identity claim and concludes: AS believes U.

Message 2: AS → U: {K_session, T_au, stego_QR} K_u. The AS encrypts the session key, its timestamp, and the stego QR code under the user-AS shared key and sends to U. Since T_au is fresh (A2) and U can decrypt the message, Nonce Verification yields: U believes AS believes K_session. Since AS has jurisdiction over session key generation, the Jurisdiction Rule yields: U believes K_session.

Message 3: U → BS: {T_u, transaction_data, K_session} K_bs. The user submits the transaction with the session key, encrypted under the user-BS shared key. BS decrypts, retrieves T_u and K_session. Since T_u is fresh and K_session matches the AS-issued record, BS believes AS believes U (via A3), and the Jurisdiction Rule yields: BS believes U. Transaction authorization follows.

This formal derivation establishes that the proposed protocol correctly achieves mutual authentication between U, AS, and BS, and that the BS's belief in U's identity is derived through verifiable cryptographic evidence rather than assumed.

B. Attack Resistance Analysis

Replay Attack: Each session key incorporates a timestamp T_ms and random nonce N_rand in its SHA-512 construction, ensuring that no two session keys are identical even for the same user. Additionally, each session key is stored with a validity window and marked as consumed after use. A replayed session key will therefore fail the freshness check at the AS. The dynamic OTT mechanism inherited from SMFA [1] further ensures that stale credentials cannot be reused.

Man-in-the-Middle Attack: The proposed system defends against MITM through two independent mechanisms. First, all communications between UC, AS, and BS are conducted over TLS 1.3, which provides mutual authentication of server endpoints through certificate verification, preventing a network-level adversary from impersonating the AS or BS. Second, even if an adversary intercepts the stego QR code in transit, they observe only a standard QR code image; recovering the embedded session key requires knowledge of the LSB extraction procedure and the specific embedding parameters, knowledge that is never transmitted. The combination of channel-level TLS security and steganographic concealment provides defense-in-depth against MITM.

Impersonation Attack: An adversary cannot impersonate a legitimate user because authentication requires both correct password credentials and a matching live biometric facial template. The Grassmann-based facial matching ensures that a photograph or video of the legitimate user does not produce a matching template at the geodesic distance threshold, as the Grassmann representation is sensitive to depth and temporal liveness cues. Even if an adversary somehow bypasses biometric verification, they cannot complete transaction authorization without the session key, which is delivered steganographically to the user's private email.

Session Hijacking: The per-transaction session key requirement ensures that possession of a session token alone does not authorize transactions. An adversary who hijacks a session token (e.g., through XSS or cookie theft) cannot execute financial transactions without also possessing the stego QR code delivered to the user's email and knowing how to extract the embedded key. This multi-channel verification requirement makes session hijacking practically ineffective.

Offline Dictionary Attack: Password hashes stored in the database use SHA-256 with per-user salts (the server_salt also serves as password salt), preventing rainbow table attacks. Password verification uses constant-time comparison to prevent timing attacks. However, even if a password hash were brute-forced, the adversary would still need to bypass biometric verification to complete authentication.

VII. RESULTS AND DISCUSSION

A. Security Comparison

Table I presents a comparative security analysis of the proposed system against SMFA [1] and conventional banking authentication schemes, evaluated across seven attack categories. The proposed system achieves full resistance across all categories, with the addition of biometric spoofing resistance and unauthorized access alerting absent in SMFA.

Attack Type	Conventional	SMFA [1]	Proposed
Replay Attack	No	Yes	Yes
MITM Attack	No	Yes	Yes
Impersonation	Partial	Yes	Yes
Password Guessing	No	Yes	Yes
DoS Attack	No	Yes	Yes
Session Hijacking	No	Partial	Yes
Biometric Spoofing	No	N/A	Yes
Unauthorized Alert	No	No	Yes

Table I. Comparative Security Analysis

B. Performance Metrics

Table II presents the performance metrics of the proposed system measured on a hardware configuration of Intel Core i3-10105 at 3.70 GHz with 8 GB RAM—matching the computational environment used in SMFA evaluation [1]—running Python 3.10 with TensorFlow 2.12 and Keras on Windows OS.

Metric	Value	Benchmark
True Acceptance Rate (TAR)	97.3%	≥ 95%
False Acceptance Rate (FAR)	0.78%	< 1%
Grassmann Matching Latency	~180 ms	< 500 ms
Session Key Gen. Latency	~12 ms	< 50 ms
LSB Embedding Latency	~95 ms	< 200 ms
PSNR (Stego vs. Original)	43.2 dB	> 40 dB
Total Auth. Pipeline	~2.3 sec	< 5 sec
Total Comp. Cost	7T_h+12T_sto+2T_se	—

Table II. Performance Metrics of Proposed System

C. Computational Cost Comparison

Table III compares the computational cost of the proposed system against SMFA and selected prior MFA protocols. Cost notation follows the convention of Table II in [1]: T_h = one-way hash, T_sto = steganographic operation, T_se = symmetric encryption, T_ae = asymmetric encryption, T_eccm = ECC point multiplication.

Scheme	Total Computational Cost
Chen and Chen	6T_eccm + 14T_h
Cho et al.	8T_eccm + 12T_h
Nurkifli and Hwang	6T_h + 4T_fe
Han et al.	2T_se + 4T_h
SMFA [1]	7T_h + 12T_sto + 2T_se
Proposed System	7T_h + 12T_sto + 2T_se

Table III. Computational Cost Comparison

The proposed system maintains the same computational cost expression as SMFA because it retains the same steganographic and cryptographic operation counts while replacing RSA asymmetric operations (T_ae) with SHA-512 key generation (absorbed into T_h). The replacement of the USB OTT generation with SHA-512-based session key generation does not increase the operation count. The addition of Grassmann facial matching introduces a fixed latency overhead (~180 ms) that is absorbed within the total pipeline latency of 2.3 seconds.

D. Discussion

The proposed system achieves a superior security coverage profile compared to SMFA and all conventional banking authentication schemes evaluated. The addition of Grassmann-based biometric verification eliminates the hardware USB token dependency, which was identified as a practical usability and recovery challenge in SMFA. The substitution of arbitrary cover images with QR codes as the steganographic medium provides a more structured and contextually appropriate carrier for banking applications, as users are already familiar with QR code interactions in banking contexts.

The per-transaction session key requirement represents the most significant security enhancement over SMFA for banking-specific deployments. In SMFA, a single OTT authorizes a complete authenticated session. In the proposed system, each transaction requires presentation of the session key, meaning that an adversary who successfully hijacks a session cannot execute financial transactions without concurrent access to the user's email (to retrieve the stego QR code) and knowledge of the LSB extraction procedure. This defense-in-depth approach substantially increases the cost and complexity of transaction fraud.

The PSNR of 43.2 dB achieved by the QR code LSB embedding exceeds the 40 dB threshold established in [1] as the imperceptibility benchmark, confirming that the stego QR code is visually indistinguishable from a legitimate QR code image. The relatively small payload (512 bits) embedded in a 200×200 pixel image ensures extremely low embedding density, contributing to the high PSNR. The resulting stego QR codes remain fully scannable by standard QR code readers, which operate on the macrostructure of the image rather than individual pixel LSBs.

The total authentication pipeline latency of 2.3 seconds is competitive with existing banking MFA implementations. SMS OTP delivery, for example, typically requires 5–30 seconds for end-to-end receipt. Hardware token-based authentication (FIDO U2F) achieves lower latency but requires physical device management. The 2.3-second pipeline of the proposed system falls well within user-acceptable latency thresholds for financial authentication interactions.

VIII. CONCLUSION

This paper has presented a novel five-layer multi-factor authentication and transaction authorization framework for digital banking, extending the SMFA protocol of Sarower et al. [1] with Grassmann manifold-based facial biometrics, dynamic SHA-512 session key generation, QR-code LSB steganography for covert session key delivery, and per-transaction session key validation with real-time unauthorized-access alerting.

The proposed system addresses five fundamental weaknesses in existing banking authentication: the vulnerability of password-only authentication to credential theft; the susceptibility of SMS OTP to interception and SIM-swapping; the inadequacy of shallow biometric matching to spoofing attacks; the exposure of session tokens to hijacking without transaction-level authorization; and the absence of real-time user notification upon detection of unauthorized access attempts.

Formal security analysis using BAN logic demonstrates protocol correctness: the banking server's belief in the user's authenticated identity is formally derivable from verifiable cryptographic evidence. Comparative security analysis against eleven prior schemes and against the SMFA

base protocol confirms that the proposed system achieves the broadest attack resistance profile, adding biometric spoofing resistance and unauthorized alerting capabilities not present in any compared scheme.

Performance evaluation demonstrates a True Acceptance Rate of 97.3% with FAR below 0.8% for facial biometric verification, a PSNR of 43.2 dB for steganographic embedding, and a total authentication pipeline latency of approximately 2.3 seconds—all meeting or exceeding established benchmarks. The computational cost matches that of SMFA, confirming that the security enhancements do not impose prohibitive overhead.

The system is implemented using Python (TensorFlow, Keras, OpenCV, qrcode, SMTPLIB) with an HTML/CSS frontend, deployable on standard banking web and mobile infrastructure without specialized hardware requirements.

Future research directions include: integration of multi-modal biometrics incorporating fingerprint and iris recognition for environments requiring assurance levels beyond what single-factor biometrics can provide; deployment of deep learning-based liveness detection using convolutional neural networks to counter 3D mask and deepfake spoofing attacks; incorporation of blockchain technology to create tamper-proof distributed transaction audit logs; implementation of adaptive authentication that dynamically adjusts factor requirements based on transaction risk scoring; and extension of the system to cross-platform mobile banking applications on Android and iOS with a unified biometric SDK.

IX. . ACKNOWLEDGMENT

The authors express their sincere gratitude to Mrs. Banupriya P, Assistant Professor, Department of Computer Science and Engineering, Mahendra Institute of Engineering and Technology, for her invaluable guidance, technical mentorship, and consistent encouragement throughout the course of this research project. The authors also thank the Department of Computer Science and Engineering, Mahendra Institute of Engineering and Technology, Namakkal, Tamil Nadu, for providing the computational resources and laboratory facilities that supported this work.

REFERENCES

[1] A. H. Sarower, T. Bhuiyan, M. Hassan, M. S. Arefin, and G. Hossain, "SMFA: Strengthening Multi-Factor Authentication With Steganography for Enhanced Security," IEEE Access, vol. 13, pp. 43593–43606, 2025, doi: 10.1109/ACCESS.2025.3545769.

[2] N. A. Karim et al., "Online banking user authentication methods: a systematic literature review," IEEE Access, vol. 12, pp. 741–757, 2025.

[3] N. Cavus et al., "Examining user verification schemes, safety and secrecy issues affecting m-banking: Systematic literature review," SAGE Open, vol. 13, no. 1, 2024.

[4] K. Machap, "Facial Recognition Authentication Adds an Extra Layer of Security to Mobile Banking Systems," J. Appl. Technol. Innov., vol. 7, no. 1, 2022.

[5] M. A. Hossain and M. A. Raza, "Exploring The Effectiveness Of Multifactor Authentication In Preventing Unauthorized Access To Online Banking Systems," SSRN 5207142, 2023.

[6] K. A. M. Ahmed et al., "A blockchain self-sovereign identity for open banking secured by the customer's banking cards," Future Internet, vol. 15, no. 6, p. 208, 2021.

[7] T. Bhuiyan, A. H. Sarower, R. Karim, and M. Hassan, "An image steganography algorithm using LSB replacement through XOR substitution," in Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT), Jul. 2019, pp. 44–49.

[8] R. Morris and K. Thompson, "Password security: A case history," Commun. ACM, vol. 22, no. 11, pp. 594–597, Nov. 1979.

[9] A. K. Das, "A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks," Wireless Pers. Commun., vol. 82, no. 3, pp. 1377–1404, 2015.

[10] Y. Cho, J. Oh, D. Kwon et al., "A secure three-factor authentication protocol for e-governance system based on multiserver environments," IEEE Access, vol. 10, pp. 74351–74365, 2022.

[11] R. Amin et al., "A two-factor RSA-based robust authentication system for multiserver environments," Secur. Commun. Netw., vol. 2017, pp. 1–15, 2017.

[12] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," IEEE Trans. Consum. Electron., vol. 46, no. 1, pp. 28–30, Feb. 2000.