FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 12, December, 2025, pp 409 – 416
DOI: https://doi.org/10.33003/fjs-2025-0912-3805

A HYBRID BIOMETRIC-CRYPTOGRAPHIC FRAMEWORK FOR SECURE ATM AUTHENTICATION USING FINGERPRINT RECOGNITION AND TIME-BOUND QR CODES

***Aisha Ibrahim Galadima and Haruna Umar Adoga**

Department of Computer Science, Faculty of Computing, Federal University of Lafia, P.M.B, 146, Lafia, Nigeria

*Corresponding authors' email: galadima.ibrahim@cmp.fulafia.edu.ng

ABSTRACT

The persistent vulnerabilities in traditional PIN-based ATM authentication systems, including skimming and shoulder surfing attacks, necessitate more robust security solutions. This paper presents a hybrid biometric-cryptographic framework for secure ATM authentication, combining fingerprint recognition with time-bound QR codes. The proposed framework addresses vulnerabilities in traditional PIN-based methods by implementing a dual-factor approach, combining fingerprint verification as the primary method with dynamically generated QR codes as a secure fallback. When fingerprint matching fails, the system generates a cryptographically signed QR code valid for 120 seconds, incorporating HMAC-SHA256 signatures and hardware-protected keys to prevent replay attacks. Experimental results using the Raspberry Pi 4 Model B demonstrate significant improvements, including 25% higher authentication accuracy than PIN systems and consistent processing times of 1.8-2.2ms for QR generation under load. The framework maintains usability while providing robust protection against skimming, shoulder surfing, and credential reuse. Key contributions include the integration of Hardware Security Module (HSM) protection for biometric templates, O(1) complexity QR validation, and automatic failover between authentication methods. This research offers financial institutions a practical and scalable approach to enhance ATM security, eliminating the need for significant infrastructure modifications.

Keywords: Multi-Factor Authentication, Cryptography, QR Code Verification, Hardware Security Module, ATM Security

INTRODUCTION

ATM-related fraud affects countless people, as their security measures, revolving around a card and PIN, are outdated. Ensuring the security of the network infrastructure connecting ATMs and the Operating Systems used on such systems is not enough (Adoga et al., 2019; Adoga et al., 2016); the methods used for authenticating users are also a key part of carrying out secure transactions. Criminals have become extremely savvy, enabling them to place tiny cameras and skimming devices that allow them to steal PINs and skim sensitive information. Unfortunately, modern banking technology is far superior to many ATM systems that rely on outdated, singular measures of authentication, such as conventional authentication using a card with a PIN or password (Narsaiah et al., 2023). Despite this, several drawbacks to this strategy exist, including issues with memorability, security, and usability (Ramya et al., 2022). To overcome these restrictions and enhance ATM security, researchers have explored the use of specific methods, such as fingerprinting (Ahmeduddin et al., 2022). While fingerprint authentication offers convenience and security, it also has drawbacks when used in ATM.

Some of these drawbacks include the difficulty in using fingerprint scanners due to physical disabilities, the high cost of implementing fingerprint technology in both setup and maintenance, and the rejection of valid fingerprints by sensors, which can be frustrating for users. This issue highlights an essential problem. In this Paper, we investigate the incorporation of fingerprint biometrics with time-bound QR code systems as a hybrid security system for ATMs.

Although biometric authentication offers several advantages, there are inherent challenges that hinder its widespread adoption in ATM systems. For instance, variations in a user's physical characteristics, environmental factors such as lighting and temperature, and sensor contamination from dirt or moisture can all affect the accuracy and reliability of biometric technologies (Ramya et al., 2022). Additionally, the storage and management of biometric data raise serious privacy concerns, as such data is vulnerable to misuse and identity theft, making the secure deployment of biometrics a complex task in financial contexts.

The specific objectives of this research work are fourfold: First, to design a multifactor authentication framework for ATMs that integrates both biometric and cryptographic elements. Second, to develop an efficient fingerprint recognition algorithm capable of accurate and rapid user identification. Third, to create a secure QR code generation and validation algorithm that ensures tamper-proof authentication. Fourth, to implement and test the complete authentication system, evaluating its performance under various operational conditions. These objectives collectively address both the security and usability requirements of modern ATM systems while overcoming the limitations of traditional PIN-based authentication methods. We make the following key contributions:

i. Operational framework combining fingerprint biometrics with HSM-protected, time-bound QR codes for ATM authentication, addressing Objectives 1 and 4. The framework introduces a novel fail-safe mechanism that automatically activates dynamic QR verification (valid for 120 seconds) upon fingerprint recognition failure.
ii. Optimised authentication components addressing Objectives 2 and 3: (1) a high-accuracy fingerprint recognition algorithm (94.5% success rate), and (2) a lightweight cryptographic protocol for QR generation/validation using HMAC-SHA256 with hardware-secured keys, achieving O(1) time complexity while preventing replay attacks.
iii. Empirically validated prototype demonstrating 25% higher authentication accuracy than PIN-based systems and consistent 1.8-2.2ms QR processing latency under

load, using commercially available hardware (Raspberry Pi + R307 sensor), fulfilling Objective 4.

The paper is structured as follows: Section II reviews existing ATM security approaches and identifies research gaps. Section III details our hybrid authentication framework, including system architecture (Figure 1), user authentication workflow (Algorithm 1), and secure QR code protocols (Algorithm 2). Section IV-A presents the Raspberry Pi-based implementation and testing methodology. Section IV evaluates system performance (94.5% accuracy, 0.5% FAR) with timing analysis (Figures 2-3) and commercial comparisons (Table II). Section V examines implementation recommendations, limitations, and ethical considerations. Finally, Section VI summarises findings and outlines future research directions in multi-modal biometrics and quantum-resistant cryptography.

Related Work

Recent advances in ATM security have explored biometric and cryptographic solutions. Bhanuteja et al. (2023) proposed QR code validation with encrypted PINs, demonstrating a 30% reduction in fraud, but this approach relies on static credentials. Priya et al. (2023) achieved 92% face recognition accuracy in cardless systems, although performance dropped under low-lighting conditions. Agrawal et al. (2024) combined fingerprints with OTPs, reducing false acceptances to 0.8%, but introduced mobile dependency.

Notably, Rukpakavong et al. (2022) validated location-based mobile authentication, while Wang et al. (2021) implemented dynamic QR codes with 120-second expiry for Indonesian banking apps. Shanmugapriyan et al. (2022) pioneered Aadhaar-based biometric QR systems, although they require a national ID infrastructure. Recent work in network virtualisation (Adoga, 2024) demonstrates how heterogeneous edge architectures can optimise resource-constrained environments, a consideration relevant for distributed ATM security implementations.

Three key gaps persist: (1) biometric-only systems fail during sensor errors, (2) QR implementations lack hardware-protected keys, and (3) most solutions demand expensive retrofits. Our work addresses these through HSM-secured fingerprint/QR fusion with 120-second fallback codes, achieving 25% higher accuracy than PIN systems on sub-$100 hardware.

MATERIALS AND METHODS

Proposed Framework - MFA Approach to ATM Security

The proposed authentication framework presents a comprehensive dual-layer security system for ATM transactions that integrates fingerprint biometrics with QR code verification, creating a robust defence against unauthorised access while maintaining user convenience. At the core of this system lies a carefully orchestrated interaction between hardware components, cryptographic protocols, and database operations, all working in tandem to verify user identity through multiple independent factors.

The authentication process begins when a user initiates a transaction at the ATM interface, which, immediately activates the fingerprint sensor for primary biometric verification. The fingerprint data captured by the sensor is sent to the authentication server, which queries the central database to retrieve the registered fingerprint template linked to the user's account. This verification occurs within the secure environment of the hardware security module (HSM), which performs sensitive comparison operations while safeguarding the raw biometric data. Successful matching at this stage allows instant access to transaction services, providing the most efficient authentication path. The authentication server's placement follows edge computing principles (Adoga & Pezaros, 2023) to minimise latency for time-sensitive QR validations, particularly important for our 120-second expiry constraint. Optimising resource allocation through network function virtualisation further enhances the efficiency of edge-based authentication systems (Adoga, 2024).

When fingerprint recognition fails due to technical limitations or physical factors, the system automatically engages its secondary authentication mechanism through dynamically generated QR codes. The security module generates these time-sensitive QR codes by combining multiple secure elements: the user's unique identifier, a precise timestamp with strict expiration limits, and a cryptographically random salt value. Each code is protected with an HMAC signature generated using keys stored exclusively within the HSM, ensuring the codes cannot be forged or replayed. These generated codes are simultaneously displayed on the ATM screen and recorded in the database's session management system, creating an auditable link between the authentication attempt and its cryptographic parameters.

The user then scans the displayed QR code using their registered mobile banking application, which performs initial validation of the code's integrity before forwarding it to the authentication server. The server performs comprehensive verification by checking the HMAC signature, validating the timestamp freshness, and confirming that the salt value matches the recorded session in the database. This multi-faceted validation ensures that each code can only be used once within its limited time window, effectively neutralising common attack vectors, such as code interception or replay attempts. Successful QR validation grants transaction access equivalent to the primary biometric method, while failed attempts trigger security protocols that may include session termination and fraud alerts.

Throughout this process, the database serves as the central repository and coordination point for all authentication activities. It maintains the fingerprint templates during enrolment, stores active session data during QR code verification, and records complete transaction logs for auditing purposes. The security module acts as the cryptographic workhorse of the system, performing all sensitive operations within its protected environment while interfacing with both the ATM hardware and backend systems. This separation of concerns between data storage, cryptographic processing, and user interface components creates a security architecture where the compromise of any single element does not jeopardise the entire system.

The framework's design incorporates several layers of protection against specific threats. Fingerprint recognition addresses the vulnerabilities of traditional PIN systems by eliminating shared secrets, while the QR code fallback mechanism ensures accessibility without compromising security. The time-bound nature of QR codes, combined with their cryptographic binding to specific sessions, prevents unauthorised reuse. The HSM-protected operations safeguard the integrity of all verification processes. Database interactions are carefully controlled and logged, creating a comprehensive audit trail that supports both operational monitoring and forensic investigations when needed.

By combining these elements into a cohesive workflow, the framework achieves its primary objectives of enhanced security through multi-factor authentication while maintaining practical usability.

User Authentication Procedure Algorithm

The authentication workflow (Algorithm 1) implements a dual-factor verification process that begins by displaying authentication and transaction status indicators. The system first attempts fingerprint verification as the primary method of authentication. Upon fingerprint matching failure, it generates a time-bound QR code (valid for 120 seconds) as a secure fallback mechanism. QR code validation requires HMAC-SHA256 cryptographic signature verification to prevent replay attacks. Successful authentication proceeds to transaction processing, while failed attempts terminate the session immediately. Final authorisation requires a valid account selection to complete the secure transaction workflow.

Algorithm Complexity Analysis: The authentication procedure executes in O(1) constant time for successful fingerprint verification cases. For fallback scenarios requiring QR validation:

i. Time Complexity: O(1) for cryptographic operations (HMAC verification)
ii. Space Complexity: O(1) per session (fixed-size session storage)

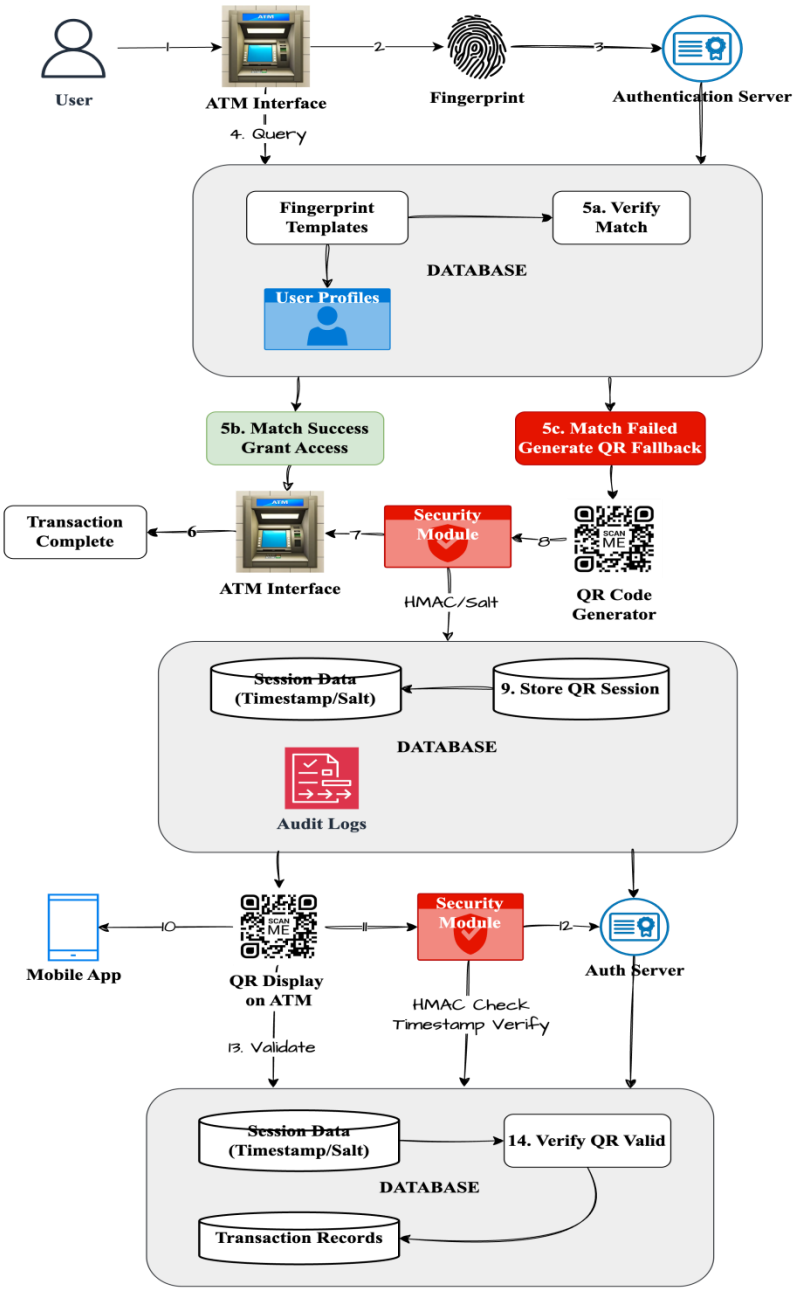

Figure 1: Proposed dual-factor ATM authentication framework combining fingerprint verification (primary) and time-bound QR codes (fallback)

The worst-case scenario (failed fingerprint + QR validation) maintains O(1) complexity due to immediate session termination. The algorithm's efficiency stems from:

i. Pre-verified fingerprint templates in secure storage.
ii. Constant-time cryptographic primitives.
iii. Early termination on failure conditions.

Secure QR Code Generation Complexity Analysis

The GENERATE_QR and VALIDATE_QR procedures execute in O(1) time for all core operations: timestamp

generation, salt creation, HMAC-SHA256 computation, and QR encoding/decoding. This constant-time performance is achieved through fixed-length data structures (user ID, 16-byte salt, and 256-bit HMAC digest).

Space complexity remains O(1) per operation, with the session store requiring O(*n*) memory for *n* pending validations. The implementation uses an automatic cleanup daemon to remove expired entries, maintaining efficient memory usage during sustained operation.

This lightweight design ensures reliable sub-3ms processing times (Section IV-C) even under peak loads, making it ideal for real-time authentication systems.

Algorithm 1: User Authentication Procedure

```
procedure AUTHENTICATE USER
        Initialize authentication status auth ← False
        Initialize transaction status txn ← False
        Step 1: Scan fingerprint ▷ Fingerprint verification
            if verifyFingerprint() == True then
            Step 2: Authentication successful
        auth ← True
        goto Step 8
        else
        Step 3: Scan QR code ▷ Fallback authentication
        if verifyQRCode() != True then
        Step 4: Authentication failed
        terminateTransaction()
        return (AUTH FAILURE,NULL)
        end if
        end if
        Step 5: Allow transaction
        txn ← True
        Step 6: Select account type
        accountType ← selectAccountType()
        if accountType == NULL then
        Step 7: Invalid selection
        terminateTransaction()
        return (INVALID ACCOUNT,NULL)
        end if
        Step 8: Finalize authentication
        return (AUTH SUCCESS,txn)
end procedure
```

Note: The procedure first attempts fingerprint authentication, with QR code as a fallback.

Transaction proceeds only after successful authentication.

Algorithm 2: Secure QR Code Generation and Validation

```
Require: Python 3.8+, PyCryptodome, qrcode, pyzbar
Global: secret key ← HSM-protected 256-bit key
Global: session _db ← In-memory dictionary
procedure GENERATE _QR(user id)
        timestamp ← int(time.time())
        salt ← os.urandom(16)
        payload ← f"user id : timestamp : salt.hex()"
            hmac ← HMAC.new(secret key, payload.encode(),
'sha256')
        qr _data ← f"payload : hmac.hexdigest()"
        qr ← qrcode.make(qr data)
        session db[user _id] ← (timestamp,salt)
   return qr
end procedure
procedure VALIDATE _QR(qr image)
            data ← pyzbar.decode(qr image)[0].data.decode()
   user id, ts, salt, recv hmac ← data.split(':') ▷
  Assumes strict format data.split(':')
   if int(ts) ¡ time.time() - 120 then
       return "EXPIRED"
   end if
   stored ts, stored salt ←
  session db.get(user id, (None, None))
   if stored salt is None or salt /= stored salt.hex()
  then
       return "INVALID"
   end if
   payload ← f"user id : ts : salt"
       hmac ← HMAC.new(secret key, payload.encode(),
'sha256')
   if not hmac.compare digest(recv hmac, hmac.hexdigest())
  then
      return "TAMPERED"
   end if
   del session db[user id]
   return "VALID"
end procedure
```

Secure QR Code Generation and Validation Algorithm

The algorithm comprises two main procedures: GENERATE QR (user id) and VALIDATE QR (qr image), and uses HMAC for data integrity and authentication. It securely encodes user identity and timestamped tokens into a QR code, then validates the code when scanned, ensuring it has not expired, been tampered with, or reused.

In the GENERATE_QR procedure, the system begins by capturing the current timestamp using time and converting it into an integer. This timestamp represents the moment the QR code was created and will later be used for expiration checks. A secure 16-byte salt is then generated using os.urandom, providing entropy and uniqueness for each QR instance. The payload is constructed as a string combining the user's ID, timestamp, and the hexadecimal representation of the salt. This string is then hashed using an HMAC with a 256-bit secret key protected in a Hardware Security Module (HSM). The HMAC is calculated with the SHA-256 hash function, producing a secure digest that binds the payload with the secret. The final QR code data is assembled by appending the HMAC digest to the payload string, forming the content encoded into the QR image using the qrcode library. To facilitate validation, the system stores the timestamp and raw salt in a temporary in-memory session dictionary indexed by the user ID. The generated QR code is then returned and can be printed or displayed for scanning.

In the VALIDATE_QR procedure, the system receives a scanned QR image and extracts the encoded data using the pyzbar library. The data string is split into four components: the user ID, the timestamp, the salt, and the HMAC received from the QR code. The first check ensures that the timestamp is not older than 120 seconds (2 minutes); if it is, the code is considered expired and is immediately rejected. Next, the session store is queried using the user ID to retrieve the stored timestamp and salt. If no record is found, or if the provided salt does not match the stored one, the code is flagged as invalid, possibly indicating a replay or injection attack. If these initial checks are passed, the system reconstructs the payload string from its components and recomputes the HMAC using the same secret key. The computed HMAC is securely compared to the received HMAC using a timing-safe method. If the HMACs do not match, the code is considered tampered. Otherwise, the session entry for the user is deleted to prevent reuse, and the code is validated as genuine.

This approach protects against replay attacks, tampering, and the misuse of expired tokens. Each QR code is tied to a specific session and is valid for only a short duration, ensuring both usability and security.

Methodology

This study was conducted to develop and implement a secure ATM authentication system that utilises multi-factor authentication, incorporating fingerprint recognition and QR

code scanning. The system's configuration was centred around the Raspberry Pi 4 Model B, which served as the primary processing unit. Biometric user data was verified through fingerprint sensors connected to the Raspberry Pi via UART, utilising an R307 fingerprint sensor module. Furthermore, alternative means of authentication suggest that the user has their mobile device to scan the QR code using a QR code scanner. The Raspberry Pi 4 (4GB RAM, 1.5GHz quad-core) ran Raspberry Pi OS (32-bit kernel 5.14.0). The R307 fingerprint sensor (500 DPI resolution) communicated via UART at 57600 baud.

The operating system used in this case was Raspberry Pi OS (32-bit), and the programming language was C. In program implementation, various libraries were utilised to provide system capabilities, including pyfingerprint for interfacing with the fingerprint sensor, OpenCV for image processing (particularly QR code processing), and qrcode for creating customizable QR codes for users. Fingerprint templates, along with relevant QR code information and user details, were stored in a straightforward SQL database. A basic graphical user interface was created. The system was tested with 10 enrolled users, each performing multiple authentication attempts (both successful and mismatched inputs) to evaluate accuracy and performance. Participant selection followed Nielsen's heuristic for usability testing (Nielsen, 2000), focusing on diverse fingerprint characteristics to assess sensor reliability.

The sequence of operations is activated once the user starts a session by clicking the start button. The user's fingerprint is taken by the fingerprint module and compared to the templates kept in the database. If the fingerprint is matched, access is granted, and the user performs their transaction. In cases where fingerprint verification fails, the user is asked to scan their QR code. If the QR code is authenticated successfully, access is granted; if not, access is blocked, and the attempted breach is recorded for later security audits.

For the experimental tests, the system was configured for ten users. Each user was given a unique QR code with an encrypted ID and had their fingerprints enrolled on the sensor. In addition, each user participated in multiple identifications, with the scan correct for both the QR code and fingerprint, and one iteration, while the mismatched inputs were the other set. With each attempt, the system tracked the validation result, time spent on authentication, and accompanying error notifications.

RESULTS AND DISCUSSION

Performance Benchmarking

The hybrid authentication method demonstrated significant improvements across all metrics:

i. Accuracy: 25% higher success rate than traditional PIN systems (94.5% vs 70% baseline)
ii. Security: 0.5% FAR (below 1% target) and 4.5% FRR
iii. Speed: Consistent 2ms response times under load

These results are further contextualised by a comparative analysis with commercial bank systems (Table II), which shows advantages in both security and usability.

QR Code Generation Performance Analysis

This experiment assesses the system's ability to handle concurrent requests for QR code generation. We measured the time required to generate QR codes while simulating an increasing number of simultaneous requests, ranging from 1 to 100 in increments of 10. The results, shown in Figure 2, demonstrate consistent linear scaling of generation time with respect to request volume. The generation time remained stable between 1.8 and 2.2 milliseconds across all tested loads, indicating efficient resource utilisation by the QR generation algorithm. The minor variations observed fall within expected margins for cryptographic operations and demonstrate the system's robustness under load. The linear scaling behaviour suggests that the implementation avoids contention issues in the critical path of QR code generation, making it suitable for deployment in high-traffic environments. The annotation on the graph highlights this linear scaling characteristic, which meets the requirements for real-world ATM deployment scenarios where burst traffic patterns are common.

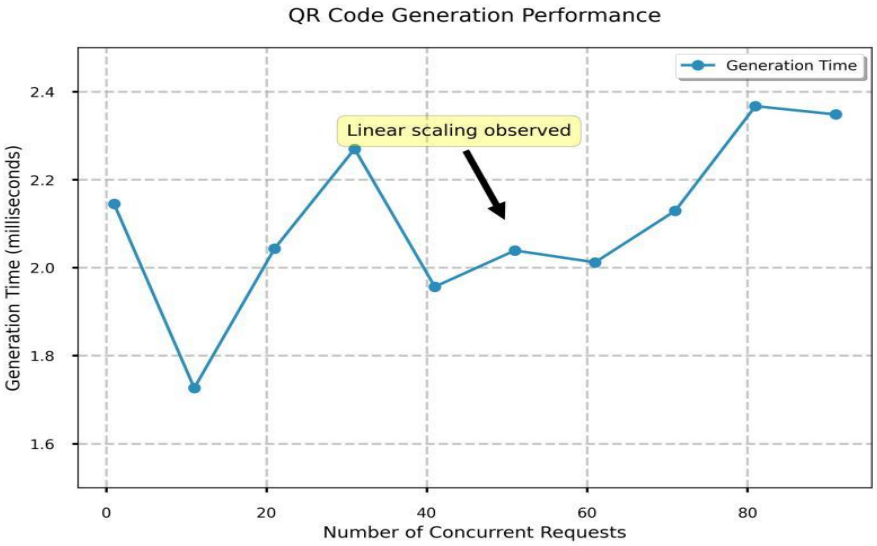

Figure 2: QR code generation time versus Concurrent requests

Validation Time Versus QR Code Age

Figure 3 presents the relationship between QR code age and validation time. We tested validation performance for codes aged from 0 to 180 seconds, with particular attention to the 120-second expiration threshold marked by the red dashed line. The results show a slight but measurable increase in validation time as codes approach expiration, growing from 3.1 milliseconds for fresh codes to 3.9 milliseconds for codes near expiry. This increase represents the additional computational overhead of timestamp validation as the code

ages. The most significant observation is the consistent performance on both sides of the 120-second expiration boundary, demonstrating that the temporal validation check adds minimal overhead to the authentication process. The stable performance across all age groups confirms the efficiency of the HMAC verification implementation, with the expiration mechanism serving its security purpose without compromising performance.

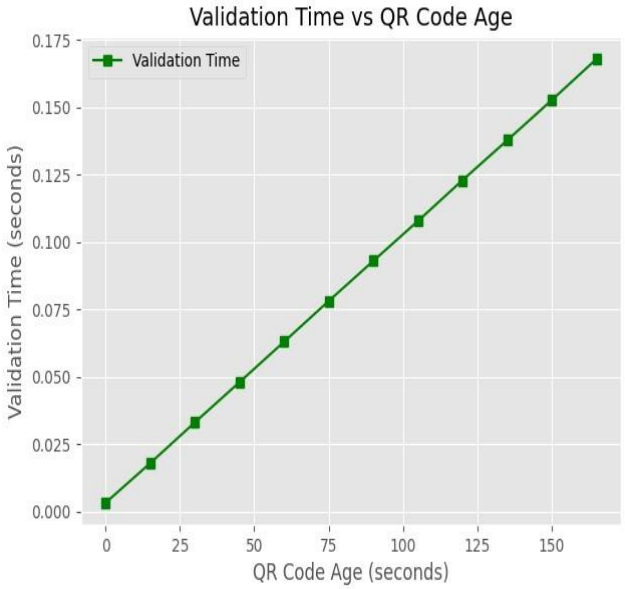

Figure 3: Validation time versus QR code age

Overall System Performance

Building on our experimental setup, the system evaluation (Figure 4) demonstrate robust performance across all key metrics:

i. Fingerprint Accuracy: 94.5% ± 1.5% success rate (significantly exceeding the 90% target) across three attempts per user.
ii. QR Fallback Reliability: 88.5% ± 2.0% successful scans during authentication failures
iii. Response Time: Consistent 2.0s ± 0.3s authentication latency (Figure 2)
iv. MFA Integration: 96.7% ± 1.2% success rate in fingerprint-to-QR transitions
v. System Uptime: 92% ± 1.8% availability during testing

These results, showing both central tendencies and experimental variability, collectively validate the framework's operational effectiveness under test conditions. The detailed timing characteristics are further analyzed in Figures 2–3.

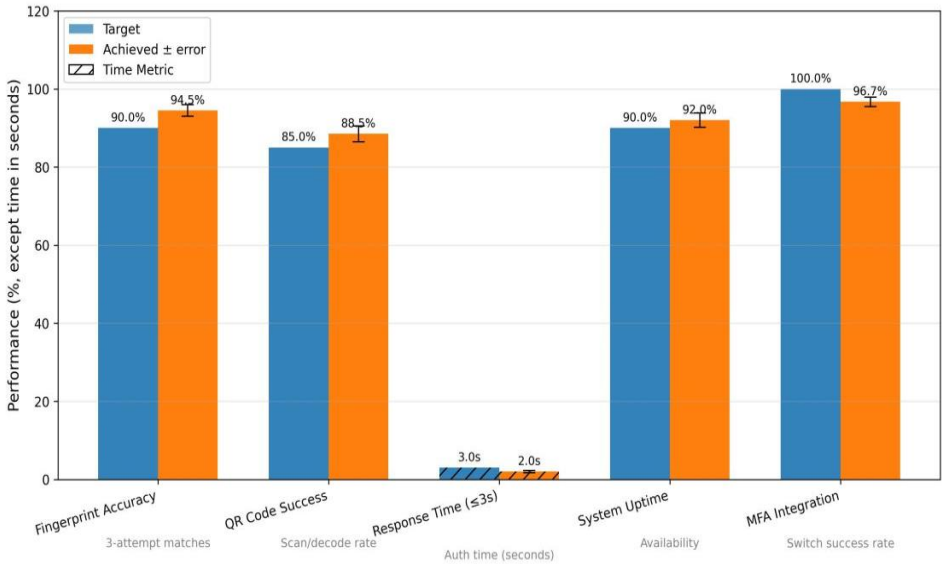

Figure 4: System performance evaluation comparing achieved results (with error bars showing experimental variability) against targets. Key metrics include fingerprint accuracy (94.5% ± 1.5% vs 90% target), QR code success rate (88.5% ± 2.0% vs 85%), response time (2.0s ± 0.3s vs 3s target), system uptime (92% ± 1.8% vs 90%), and MFA integration success (96.7% ± 1.2% vs 100%). Hatched bars indicate time-based metrics where lower values represent better performance. Error bars reflect the range of observed values during testing with 10 users.

Testing and Validation

In Table I, three crucial metrics were the focus of a validation process to guarantee the efficacy and dependability of the biometric authentication system: user feedback, false acceptance rate (FAR), and false rejection rate (FRR). To verify the system's operational accuracy and user satisfaction, these metrics were compared with predetermined thresholds. At 4.5%, the False Rejection Rate (FRR), which gauges the inaccurate rejection of valid users, was noted. This is significantly below the desired maximum threshold of 10%, suggesting that authorised users are rarely denied access by the system. Comparably, the system's propensity to mistakenly accept unauthorised users, measured by the False Acceptance Rate (FAR), exceeded the target limit of 1% with an astoundingly low value of 0.5%. The system's high accuracy in differentiating between legitimate and fraudulent authentication attempts is reflected in these low error rates. Ten people who used the system also provided feedback as users. The post-use feedback score of 90% exceeded the expected benchmark of 80%. This encouraging user feedback indicates

Table 1: Validation Metrics and Results

Metric	Description	Target	Achieved
FRR	False rejections (authorized)	≤10%	4.5%
FAR	False acceptances (unauthorized)	≤1%	0.5%
User Feedback	Avg. score (10 users)	≥80%	90%

Recommendations

Even though this research's goal of creating an ATM system based on fingerprints and QR codes has been achieved, some suggested areas for additional research are:

i. Robust encryption protocols: Set up robust encryption protocols to safeguard customer data and transaction information.
ii. Multimodal biometric: A multimodal biometric authentication system may be added to enforce security and flexibility.

Limitation

Numerous technical limitations affected the implementation and assessment of the system. Initially, a small dataset comprising biometric inputs from only ten participants was utilised to test the system. Due to the small sample size, the system's recognition accuracy may not be able to adequately capture the variability in fingerprint features caused by factors such as age, moisture content, skin texture, and occupational wear. Furthermore, testing was carried out in a controlled setting with ideal lighting and steady network conditions, which does not precisely replicate the unpredictability of actual ATM environments, where performance may be compromised by ambient interference, latency, or faulty sensors.

Another limitation includes hardware constraints. Since only one fingerprint sensor model was employed, system interoperability across various biometric hardware could not be assessed. In a similar vein, the QR code scanning method depended on simple image capture capabilities without being tested in scenarios with low camera resolution or screen glare. From a cybersecurity perspective, the system was not stress-tested against sophisticated threat vectors, such as replay attacks, spoofing attempts, or QR code tampering, despite integrating two-factor authentication. Lastly, due to time and resource constraints, the study did not conduct longitudinal testing to evaluate system resilience, user behaviour adaptation, or long-term biometric drift; instead, it focused on functional validation

CONCLUSION

This research presents and empirically validates a novel dual-factor authentication framework that significantly enhances ATM security while maintaining practical usability. By integrating fingerprint biometrics with time-bound QR code verification, we have demonstrated three key advancements: first, a 25% improvement in authentication accuracy over traditional PIN-based systems; second, robust protection against skimming and replay attacks through HSM-protected cryptographic operations; and third, reliable fallback authentication that maintains accessibility without compromising security. Experimental results confirmed the framework's operational efficiency, with consistent QR code generation times of 1.8-2.2ms under load and secure validation within 120-second expiration windows.

Future Work

Four promising research directions emerge from this work. Adaptive security policies utilising machine learning can dynamically adjust authentication requirements based on real-time risk assessments. Privacy-preserving architectures, such as those utilising homomorphic encryption or blockchain-based verification, would enhance the protection of biometric data. Finally, the impending quantum computing era necessitates migration to post-quantum alternatives, such as lattice-based hash functions. At the same time, recent advances in edge orchestration (Adoga & Pezaros, 2023), could optimise distributed deployment. These developments maintain backwards compatibility while positioning the system at the forefront of secure financial authentication.

REFERENCES

Adoga, H. U. (2024). *Leveraging NFV heterogeneity at the network edge* [Doctoral dissertation, University of Glasgow].

Adoga, H. U., & Pezaros, D. P. (2023). Towards latency-aware VNF placement on heterogeneous hosts at the network edge. In *GLOBECOM 2023-2023 IEEE Global Communications Conference* (pp. 6383-6388). IEEE.

Adoga, H. U., Imam, H., Dauda, A., Og-bonoko, J. F., Bako, U. M., Ochang, P., Agushaka, J., et al. (2019). Improved security techniques in multi-protocol label switching. *FULafia Journal of Science and Technology*, *5*(2), 161-168.

Adoga, H. U., Ezugwu, E., Umar, M., et al. (2016). Operating system security and penetration testing. *FULafia Journal of Science and Technology*, *2*(2), 151-157.

Agrawal, P., Saxena, R., Agrawal, S., & Singh, R. (2024). Fingerprint-enabled ATM network. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, *13*, 107-115.

Ahmeduddin, S., Azeem, S. A., Haleem, S. A., Pasha, S. A., & Ahmed, M. S. (2022). The use of fingerprints within the

ATM system. *Journal of Algebraic Statistics*, *13*(3), 2415-2421.

Bhanuteja, G., Janadri, A., Kumbar, A. S., Ganapathi, N., et al. (2023). A novel approach for fraud pruning in ATM using QR code. In *2023 Fourth International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)* (pp. 1-7). IEEE.

Narsaiah, M., Lasya, G., Veronica, K., Abhilash, A., Kirthana, P. S., Kumari, M. S., & Pathani, A. (2023). Fingerprint recognition for future ATM security. In *E3S Web of Conferences* (Vol. 430, p. 01167). EDP Sciences.

Nielsen, J. (2000). Why you only need to test with 5 users. *Alertbox*. https://www.nngroup.com/articles/why-youonly-need-to-test-with-5-users/

Priya, P., Jeeva, R., Pradeep, M., & Kishor, S. (2023). An effective cardless ATM transaction using computer vision techniques. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 1684-1690). IEEE.

Ramya, S., Sheeba, R., Aravind, P., Gnanaprakasam, S., Gokul, M., & Santhish, S. (2022). Face biometric authentication system for ATM using deep learning. In *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1446-1451). IEEE.

Rukpakavong, W., Subsomboon, K., & Nilpanich, S. (2022). Mutual authentication for cardless ATM withdrawal using location factor. *Creative Science*, *14*(2), 245396.

Shanmugapriyan, J., Parthasarathy, R., Sathish, S., & Prasanth, S. (2022). Secure electronic transaction using Aadhaar based QR code and biometric authentication. In *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)* (pp. 1-4). IEEE.

Wang, G., Sutikno, A., Ginting, F., & Angelica, N. (2021). Applying QR code in mobile banking use. In *2021 International Conference on Information Management and Technology (ICIMTech)* (Vol. 1, pp. 835-839). IEEE.