ISSN: 2663-5798 || Arab Journal for Scientific Publishing

الإصدار السابع – العدد تسعة وستون
تاريخ الإصدار: 2 – تموز – 2024م

www.ajsp.net

"Implementing a Secure Mobile Application for Cardless Transactions using QR Code and Hybrid AES-ECC Encryption"

Researchers:

Noor Jaber Hamad[1, *], Abbas Abdulazeez Abdulhameed[2], Mudhafar Hussein Ali[3]

[1] Computer Engineering Department, College of Engineering, Al–Iraqia University, Baghdad, Iraq

[2] Department of Computer Science, University of Mustansiriyah, Baghdad, Iraq

[3] Network Engineering Department, College of Engineering, Al–Iraqia University, Baghdad, Iraq

Corresponding Author:

Noor Jaber

ISSN: 2663-5798 || Arab Journal for Scientific Publishing

الإصدار السابع – العدد تسعة وستون
تاريخ الإصدار: 2 – تموز – 2024م

www.ajsp.net

Abstract:
The use of mobile banking has gained wide acceptance, due to the convenience and ease of access via mobile phone. However, the increasing reliance on it by users has been accompanied by security challenges such as phishing and data breaches. Ensuring the security and integrity of data transmission is crucial to building user trust. Data encryption during transactions is the ideal solution for data security and integrity. To achieve this, we propose a system that uses mobile applications to transmit and secure cardless transactions, using QR code fusion with a hybrid AES-ECC algorithm. This algorithm encrypts the data and authenticates it via a QR code. The method involves hybrid encryption that combines Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC). Instead of using the AES key directly for encryption, a key is generated through the ECC algorithm. Decryption is done using the ECC private key. When the user receives the QR code, they can scan it to access the original text. The proposed system has the advantage of storing QR codes on users' phones instead of servers, while keeping the encryption keys embedded in the hybrid algorithm for greater efficiency and ease. The efficiency of the proposed system was tested using different data sizes, to measure the encryption and QR generation time, and the time required to scan the QR code and decrypt. In addition, the QR code's ability to store data. The results showed the effectiveness of the system, its ease of use, and its ability to transfer data securely.

Keywords: Encryption, Decryption, Hybrid (AES&ECC), QR Code, Security.

1. Introduction

The emergence of mobile banking has transformed financial transactions. Mobile banking offers convenience, accessibility, security and ease of use, making it the preferred choice for users. However, the proliferation of online transactions has created security challenges and threats such as phishing and data breaches, which undermine user confidence in using online financial services. To address these challenges, there is an urgent need for solutions that prioritize the security and integrity of transactions while maintaining user convenience. Dr. Shihab El-Din (Shahabuddin, 2018) emphasizes the importance of strong data security, stressing that financial institutions must reassure users by ensuring secure transaction transmission and an impenetrable system. The rise of cardless transactions via mobile apps is a way to enhance security and thwart malicious activity. However, these systems are not without vulnerabilities, which require strong authentication and encryption mechanisms to protect sensitive data and mitigate risks (Isobe & Ito, 2021) (Boraiah, 2019). Several cryptographic techniques have been introduced to address concerns about data security and integrity, including symmetric key encryption, which uses a single key for both encryption and decryption (Alenezi & et al 2020), and asymmetric key encryption methods, which use different keys for encryption and decryption (Lalem & et al, 2023). Later, hybrid encryption techniques were applied, which combines asymmetric and symmetric encryption algorithms (Kuppuswamy & Al-Khalidi, 2014). Motivated by these concerns, this research seeks to develop a mobile application designed to secure cardless transactions by integrating QR code authentication with a hybrid AES-ECC algorithm, addressing the critical need for secure and user-friendly solutions in mobile banking.

2.Literature review

The following papers present work related to the proposed approach, which can be classified into four main sections:

2.1 Mobile Banking Features

In their study, (Yu & Nuangjamnong, 2022) identified the key features of mobile banking, highlighting its convenience and adaptability. Their study emphasized functions such as bill payment, balance inquiries, and check scanning, and demonstrated the ease with which users can engage in mobile banking activities, including text messaging and locating nearby ATMs or banks. Researchers noted the efficiency of mobile banking, which is represented by password-protected withdrawals and seamless money transfers between accounts.

(Lee & et al, 2013) focus on mobile banking strategies used by financial institutions and assess their readiness to provide such services. Their survey-based approach reveals that larger institutions are more likely to adopt mobile banking, leveraging their size and financial strength to provide advanced security services and procedures. However, credit unions, although small in size, show a greater propensity to offer mobile banking, due to their collaborative nature and customer focus. The study emphasizes the importance of security and technological progress in shaping the future course of mobile banking services.

2.2 The Security Features of Banking App

In their study, (Rahman & et al, 2020) highlight the importance of security concerns in hindering the adoption of mobile banking. They found that enhanced performance, stronger security measures, and increased trust positively influence the uptake of mobile banking for financial transactions. They stressed that the development of the banking industry and the expansion of its markets depends on developing effective security procedures and enhancing customer confidence in banking procedures.

Similarly, (Nimmi & Janet, 2018) conducted a comprehensive functional analysis of various mobile banking applications, including Tez /Google Pay, Paytm, Paypal and Bhim. Their study focused primarily on evaluating the security features of these applications, taking into account the critical nature of transactions conducted through mobile banking services installed on smartphones. The mobile payment applications analyzed included a range of security measures, such as authentication protocols, machine learning-based fraud detection mechanisms, OTP validation of transactions, TLS connection mechanisms, and user identification and password requirements to access the application. Table 1 provides a detailed overview of security features across different banking applications.

Table 1 Security Feature Comparison of Banking Applications (Nimmi & Janet, 2018)

Basis for Comparison	Paytm	BHIM	Google Tez
Auto logout feature	No	Yes/Timeout	Yes
Authentication	Username and password, biometric Authentication	Password (4- digit-UPI pin)	Google PIN or screen lock
Confidentiality	OTP	3-Factor Authentication	Audio QR (QAR) and UPI Pin
Transaction time	Medium	High	Low
Cash Mode	No	No	Yes
Access without Internet	Phone call and secured Paytm PIN	Unstructured Supplementary Service Data (USSD) based	USSD based

2.3 Secure Data Transfer

Authors (Nie & Hu, 2008) describe the security challenges facing mobile banking, including denial of service attacks via SMS and virus attacks via wireless networks or Bluetooth. Proposed security measures include encryption techniques, identity verification, and digital signatures. Encryption, especially using AES and ECC algorithms, plays a vital role in ensuring data confidentiality and integrity. AES-128 is used to encrypt data, with ECC generating the encryption key, which provided strong security and faster encryption and decryption. ECC is highlighted for its effectiveness in mobile environments, which presents challenges to attackers due to the complexities involved in cracking an ECC key.

The study conducted by (Mallouli & et al, 2019) Use of symmetric and asymmetric encryption algorithms in cloud computing for data security. Symmetric algorithms are faster and require less computational power than asymmetric algorithms, but the 256-bit symmetric AES algorithm is preferred for banking because of its strong security properties. ECC is defined as a secure and efficient encryption algorithm, especially suitable for mobile phones and Android applications, compared to RSA. The results of which are presented in Table 2.

Table 2 Comparison Between RSA and ECC (Mallouli & et al, 2019)

Security Bit Level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

(Sudha & Ganesan, 2013**)** created a mobile application for the medical sector that requires authentication, control and access to healthcare systems. The application focuses on the security of patient data and ensures secure access to medical records by authorized users only. The system uses ECC for authentication and access control. Implementation the application on Wi-Fi-enabled Android phones facilitates secure data transfer and enhances security in electronic medical records (EMR) delivery to patients.

The study conducted by (Nguyen & Lam, 2021) on the implementation of ECC as a powerful alternative to RSA in smart parking system. ECC shows efficiency in terms of key length and processing speed, enhancing security with smaller key sizes. Performance evaluations confirm the superiority of ECC, especially on resource-constrained devices, providing efficient digital signatures in mobile applications.

(Boraiah, 2019) proposed a system designed to counter cybercriminals who try to steal financial transaction information. The use of ECC technology and QR code in the secure Android app for cardless transactions ensures the confidentiality of encrypted financial details. Public and private keys facilitate secure communication between the bank and the customer, providing effective protection against cybercrime activities.

Presented by (Hodowu & et al, 2020) presents an improved data security model for secure data transfer in cloud environments. The use of two-level encryption technology including AES and ECC ensures secure communication between client devices and the cloud. The use of ECC in cooperation with ECDSA ensures data integrity throughout the transfer process, providing effective protection against unauthorized access.

Rehman, & et al (2021)**.** Propose a secure and optimized approach to sharing data across cloud environments. The combination of ECC and AES algorithms ensures data security and integrity while reducing storage requirements. ECC generates keys for AES, which ensures the security of the ciphertext and provides an effective solution for secure data transfer in cloud storage technologies.

2.4 Authentication

In their study, (Al Imran & et al, 2019) presented a cardless transaction system that uses one-time password (OTP) authentication, focusing on its role in enhancing security and mitigating vulnerabilities in biometric authentication. Meanwhile, (Wahjuni & Pristian, 2016**)** created an Android-based online transaction system that integrates token authentication using the One Time Pad (OTP) algorithm, famous for its strong encryption using short keys. Addressing SMS-related transaction challenges, (Kumar & et al, 2015) proposed a QR-based solution, ensuring exclusive decryption by the bank to maintain data integrity and transaction reliability. (Adukkathayar & et al, 2015) presented an NFC-based payment system that includes multi-factor authentication, leveraging facial recognition and a 4-digit PIN, while (Sharma & Bohra, 2017**)** presented a comprehensive five-stage authentication framework that links QR codes with user IDs. Unique IDs (UIDs) to enhance security.

3.Research Methodology

Previous studies have shown that implementing a secure and efficient system for exchanging transactions requires strong mechanisms for authentication and data encryption. In response to these requirements, we proposed a new approach that improves data security and integrity, by encrypting data using the hybrid AES-ECC algorithm and QR code authentication. QR code is used to hide data encrypted and retained, which is then stored in the mobile.

3.1 Hybrid AES-ECC Algorithm

Recent research results revealed that integrating the ECC and AES algorithms into a hybrid encryption approach is a powerful strategy to enhance the security and efficiency of encryption systems. This combination exploits the strengths of both algorithms, yielding many useful results tailored to diverse cryptographic requirements. These advantages include tight security measures, enhanced operational performance, and the use of shorter encryption keys (Rehman, & et al, 2021) (Kader & et al, 2014) (Deepa & Parvathi, 2015).

3.1.1 Hybrid ECC-AES encryption/decryption methodology

The proposed system presents a hybrid approach that combines (AES -128 bits) and (ECC -160 bits). The methodology is summarized as follows:

1. A text file is chosen as input for encryption and decryption operations.
2. The text file is encrypted using Advanced Encryption Standard (AES), a symmetric encryption algorithm. As a second stage, the AES key itself is encrypted using a Public key generated by the ECC algorithm.
3. The encrypted file is uploaded to the server.
4. When recovering the encrypted file, the received file first undergoes the AES key decryption process using the private key generated through the ECC algorithm. The recovered AES key is then used to decrypt the data, which has been encrypted using the AES algorithm.

This approach seamlessly combines the efficiency of AES in handling data encryption with the security advantages provided by ECC for secure key transmission (Sudha & Ganesan, 2013) (Vahdati & et al, 2019) (Singh & Singh, 2015) (Bommala & et al, 2019) (Hosam & Ahmad, 2019).

Worth noting that the encryption and decryption keys used remain within the hybrid algorithm. This makes it inaccessible to individuals, including users. Figure 1 shows the methodology for encrypting and decrypting data using the hybrid AES-ECC algorithm.

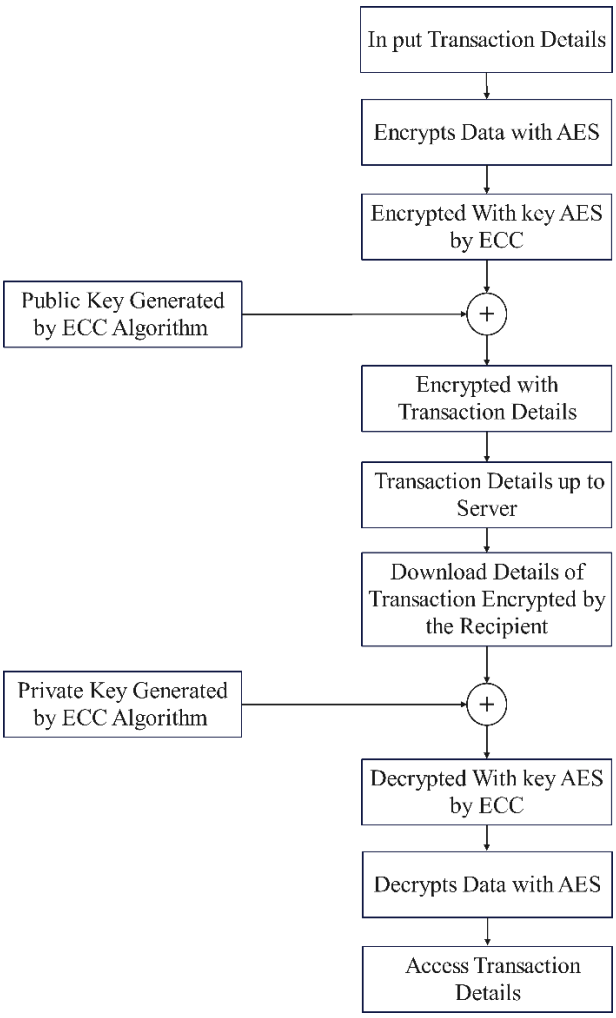


Figure 1. AES-ECC Encryption/Decryption

3.1.2 Application operational procedures

The hybrid system consists of AES 128-bit algorithm and ECC 160-bit algorithm. In the AES algorithm, the system generates a 128-bit random number (K) to serve as the key for the AES-128 algorithm. In return, the system generates a 160-bit random number as the private key for the ECC-160 algorithm. This private key is used internally to generate a public key for AES key encryption. When you enter information, the system encrypts it using the AES algorithm using the key (K) and then encrypts the AES key using the public key generated by the ECC algorithm. Upon receiving the encrypted data, the system decrypts the AES key using the private key of the ECC algorithm. Finally, the recovered AES key is used to decrypt the content of the AES-encrypted text file.

By integrating the key generation process into the hybrid algorithm and automating the encryption and decryption procedures, the system simplifies the encryption and decryption processes for the user, ensuring security and ease of use.

The operational procedures include two algorithms. Algorithm 1 describes the process of encrypting the text using the public key generated by ECC, while Algorithm 2 defines the decryption of the text using the private key generated by ECC. These algorithms involve basic mathematical operations, including multiplication, addition, and subtraction, primarily concerned with manipulating points on a given elliptic curve within the application (Hosam & Ahmad, 2019).

Algorithm 1: Encryption using the ECC-generated public key

Output: Ciphertext ©
C = [(K * G), (M + K * Q)]
Where: K is an integer chosen randomly from the range (1, p-1).
M: represents the ciphertext (AES Key) to be transmitted and is a point on the curve.
C = (C_1, C_2)

Algorithm 2: Decryption using the ECC-generated private key
M = C_2 – [d * C_1]
Extract the key K from M.
Utilize K to decrypt the ciphertext.

3.2. Proposed Approach

The proposed approach consists of three distinct phases, as shown in Figure 2:

The first stage: registration and installation
Users first need to install the secure mobile application provided by the banking institution. Registration within the system requires providing basic details, Including the password and email address specified by the bank. Upon successful completion of the process, users will be able to access the application.

The second stage: data encryption and exchange
Users initiate the process of exchanging data with the intended recipients. Users provide the recipient's email address and account number, enabling subsequent actions. Data intended for transmission is encrypted using the hybrid EAS-ECC algorithm. Users generate a QR code containing the encrypted data, which is stored on their mobile device. Email acts as a medium to transmit the QR code image to the intended recipient. The transfer process uses a security protocol an additional layer of authentication is introduced, in the form of an additional password.

The third stage: Data reception and decoding
Upon successful receipt of the sent encrypted data, the designated recipient immediately receives a notification containing a QR code. The recipient stores this QR code on their mobile device. Then, using the secure app pre-installed on the device, the recipient follows established authentication protocols during the login procedure. The user presents the QR code to scan and decryption able to access and read the original text. Figure 2 depicts a flowchart of the data transfer process and provides a visual representation of the three stages.

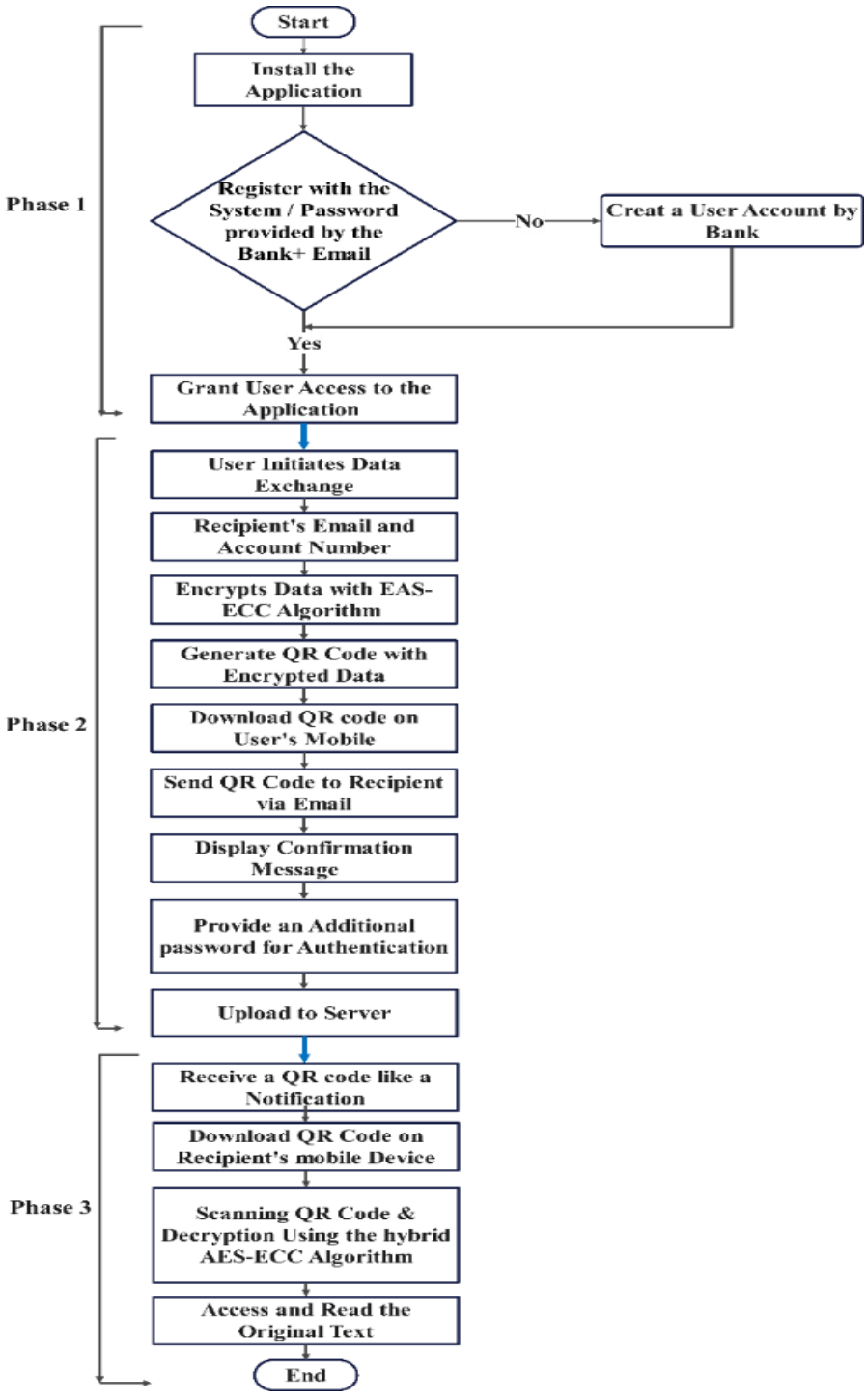


Figure 2. Flowchart Depicting the Basic Stages of the Proposed Approach

3.3. Design Specification

The proposed application integrates QR code authentication with a hybrid AES-ECC algorithm. This paper presents the application design specifications systematically, dividing the processes into clear and defined steps. For ease of understanding, the design is illustrated using system architectural diagrams and flowcharts.

3.3.1 System Architecture

The system's architectural framework includes three main components as shown in Figure 3: users, financial institutions, and Firebase servers. These elements are connected to each other through a central Firebase server. The “Users” segment represents users who use mobile applications to transact with banking institutions. Conversely, the “Banking Institutions” component includes financial institutions responsible for user services, including transaction processing and data encryption/decryption. The central Firebase Servers element acts as a cloud-based platform, providing a secure and scalable infrastructure that supports critical services such as data storage, messaging, and authentication. It is necessary to emphasize that this central server plays a pivotal role as the primary communication center, ensuring data exchange and secure messaging between users and banking institutions.

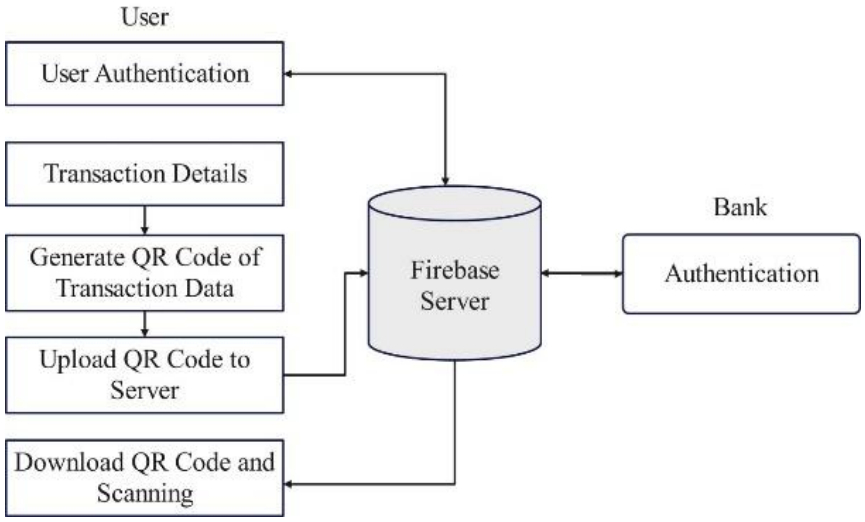


Figure 3. Architecture Diagram System

3.3.2 Proposed System Structure

To utilize the system, users must create an account within the app, and provide basic details such as username, email address, and bank-specified password for authentication. Once the account is created, Firebase creates a unique user ID on the server to track the user's actions. When logged in, users can initiate secure transactions. Data intended for sharing is subject to AES encryption, with the public key generated by ECC incorporated into the hybrid algorithm. This encrypted data is converted into a QR code and sent to the recipient via email. After logging into the app, the recipient saves the QR code on their mobile device, and the recipient scans the QR code to decrypt the data. Figure 4 Flowchart depicts a detailed visual representation of the system's operations.

ISSN: 2663-5798 || Arab Journal for Scientific Publishing

الإصدار السابع – العدد تسعة وستون
تاريخ الإصدار: 2 – تموز – 2024م
www.ajsp.net

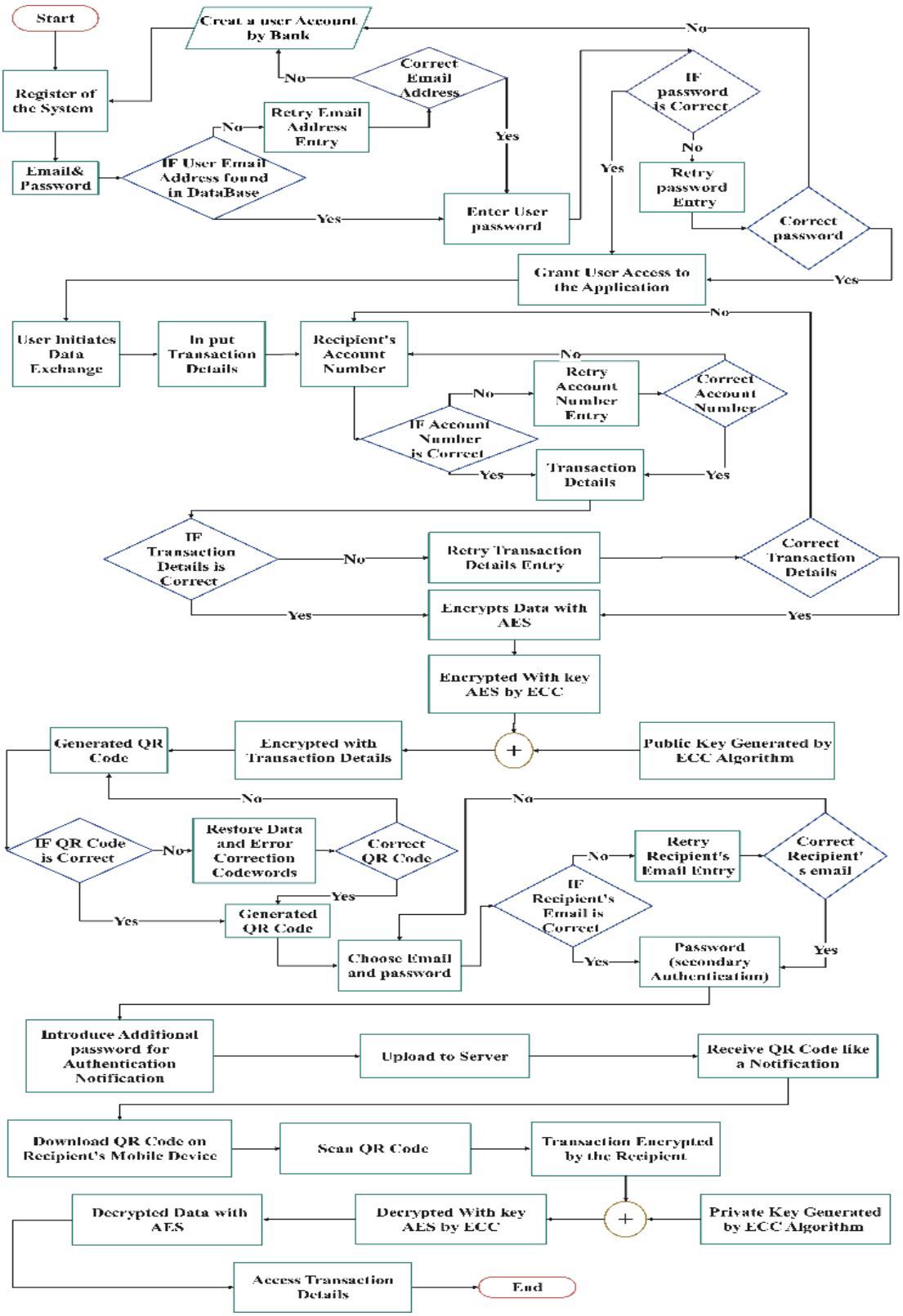


Figure 4. Flow Diagram of the Application

4.Components of the Proposed System

The proposed system consists of three main components, which collectively contribute to the operation of the system:

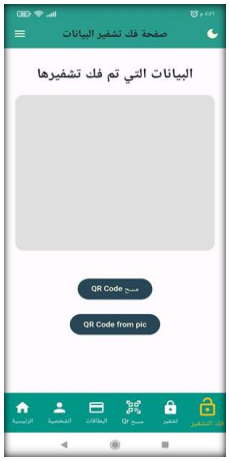
4.1 Mobile Application Interface

This part of the system allows users to engage in secure cardless transactions. It includes interfaces for registration, login, encryption, QR code generation, payment confirmation, transaction history viewing, QR code scanning and decryption. Figure 5 contains visual representations of the interface design.

(a) Interface Registration and Login

(b) Interface for Transferring Financial Transactions

(c) Hybrid AES-ECC Encryption and QR Code Generation



(d) QR Code Scanning and Hybrid AES-ECC Decryption

Figure 5. Mobile Application Interface (a) Interface Registration and Login, (b) Interface for Transferring Financial Transactions, (c) Hybrid AES-ECC Encryption and QR Code Generation, (d) QR Code Scanning and Hybrid AES-ECC Decryption.

4.2 QR Code Generation and Scanning

The mobile application can generate and scan a QR code. Users can easily enter encrypted transaction details and generate a QR code within the application. Then sent to the recipient. Upon receipt, the QR code is scanned and the data is decrypted. Figure 6 shows a diagram of QR code generation and secure data transfer processes.

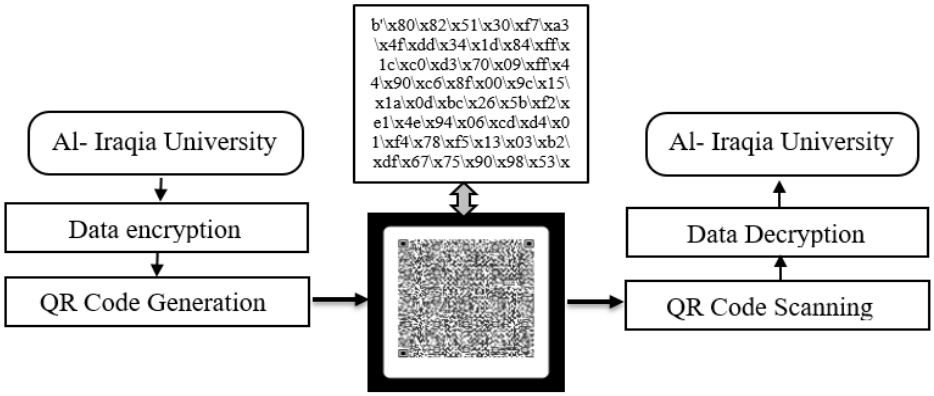


Figure 6. QR Code Creation and Secure Data Transfers

4.3 Hybrid AES-ECC Encryption/Decryption

The motivation behind choosing the hybrid AES-ECC algorithm in proposed system is its advantages, which include enhanced security, efficient performance, shorter encryption keys, and resilience against potential security breaches.

5.Implementation of the Proposed System

The proposed system is designed to run on the Android operating system. To edit the code Visual Studio Code is used, and Flutter facilitates GUI development in the Dart language. The Pointy Castle library (Pointy Castle Library, 2020) is used to execute the code, including implementing a hybrid encryption algorithm by combining the AES-128-bit and ECC-160-bit algorithms. To generate the QR code, it relies on the QR code generator provided by Dart's qr Flutter package. The implementation of the proposed system is based on Firebase services, which includes several aspects including login authentication, cloud storage, and real-time database. Embedding Firebase code within the system is a crucial step, as it creates a seamless connection between the Firebase platform and the mobile application installed on the user's device. Within the app, the authentication mechanism is based on email credentials and password specified by the bank. Upon successful authentication, a unique client ID is generated, which plays a pivotal role in facilitating various functions within the application.

6.System Testing and Evaluation

To ensure that the system sections work as expected, testing and evaluation of the function and performance of each section was conducted.

6.1 Evaluation of the Mobile Application Interface

Tests included compatibility with different Android devices through the application's ability to adapt to different screen sizes. A questionnaire was given to ten users to collect their opinions about the application. In addition, the automatic logout feature was tested when executing other applications, such as receiving an SMS or phone call, or leaving the device inactive for 90 seconds. The results of these tests showed that each mobile application interface is effective and easy to use.

6.2 Measure Encryption Time and Generate QR Code

The encryption time was measured for seven files with different data sizes using the hybrid AES – ECC algorithm, as shown in Table 3

Table 3 Time for Encrypting Data and Generating QR Code

File	Size (Byte)	Execution1 (ms)	Execution2 (ms)	Execution3 (ms)	Average (ms)
1	50	290	270	250	270
2	100	311	300	297	302
3	200	310	330	294	311
4	300	330	325	313	322
5	400	419	320	305	348
6	500	549	321	330	400
7	550	550	450	368	456

6.3 QR Code Scanning and Decryption

The time required to scan and decrypt the data presented in Section 6.2 was calculated as shown in Table 4

Table 4 The Time for Scanning the QR Code and Decryption

File	Size (Byte)	Execution 1 (ms)	Execution 2 (ms)	Execution 3 (ms)	Average (ms)
1	50	173	108	97	126
2	100	140	120	136	132
3	200	200	150	130	160
4	300	220	182	108	170
5	400	152	242	158	184
6	500	150	301	180	204
7	550	Error	Error	Error	Error

The results shown in Table 3 and Table 4 indicate that the data encryption time is greater than the decryption time. The time it takes to complete these tasks expands as the file size increases, as shown in Figure 7. It should be noted that the encryption time may change based on factors including the complexity of the AES-ECC algorithm, the length of the text, and the bandwidth of the Internet.

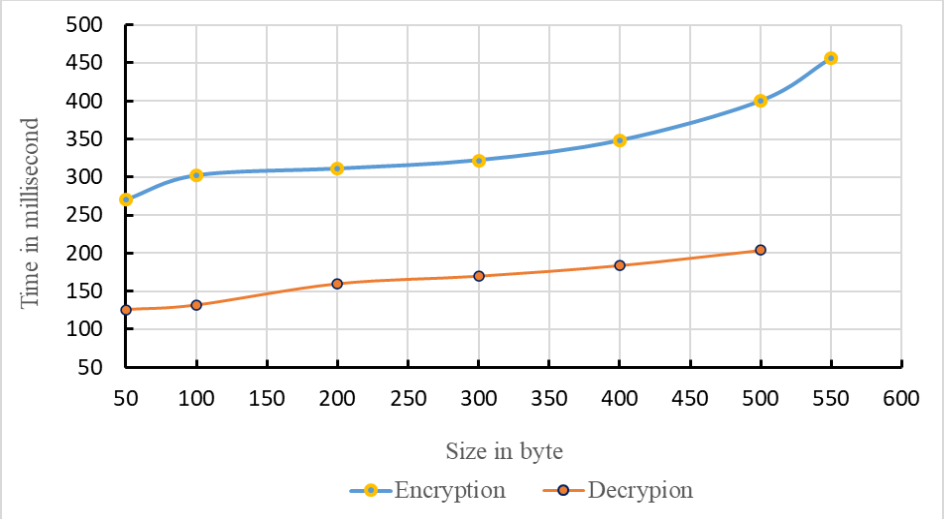
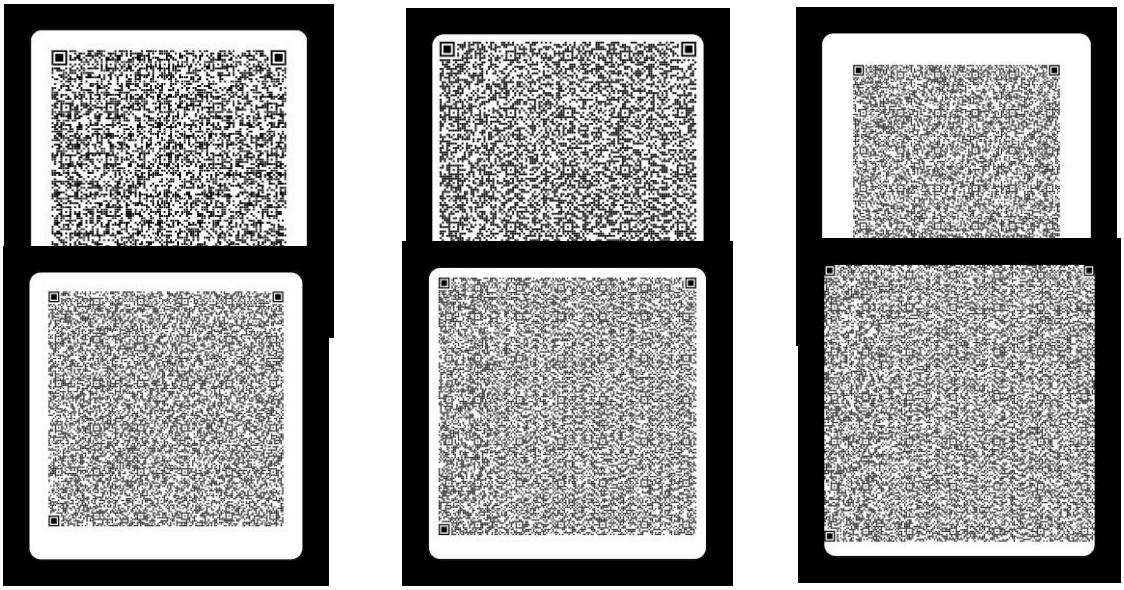


Figure 7. Encryption and Decryption Time in the Hybrid AES-ECC Algorithm

6.4 Testing the QR Code's Ability to Hold Files

This test aims to evaluate the ability of the QR code to accommodate different file sizes. The test results shown in Table 4 showed that the maximum size of data that the QR code can scan is 500 bytes. Figure 8 shows the different formats of QR codes containing encrypted data of different sizes.



300 Byte **400 Byte** **500 Byte**

ISSN: 2663-5798 || Arab Journal for Scientific Publishing

الإصدار السابع – العدد تسعة وستون
تاريخ الإصدار: 2 – تموز – 2024م
www.ajsp.net

550 Byte (Without generating)

Figure 8. Various Encrypted QR Code Formats

6.5 Testing the Efficiency of Hybrid AES-ECC Encryption Against AES/ECC

The testing focused on comparing hybrid AES-ECC versus ECC-384-bit and AES-192-bit based on data encryption and decryption times. Encryption and decryption times were calculated independently for each algorithm. Table 5 shows the test results.

Table 5 Comparing Encryption and Decryption Times

Size (byte)	Time (ms) AES	Time (ms) ECC	Time (ms) ECC-AES
50	52	489	396
100	58	504	434
200	68	533	471
300	76	552	492
400	83	633	532
500	94	698	604

The results in Table 5 show that when ECC is used to encrypt and decrypt data, it takes longer compared to the hybrid AES-ECC algorithm. While the AES algorithm, when implemented, takes less time, the algorithm alone is not as secure as the proposed hybrid system. The reason for this security difference is that a hybrid system adds an extra layer of security. If a hacker can decrypt one level of encryption, it will be more difficult for him to decrypt the second level using the same algorithm, enhancing overall security. Figure 9 shows a comparison of total encryption and decryption times between AES, ECC, and the AES-ECC hybrid model.

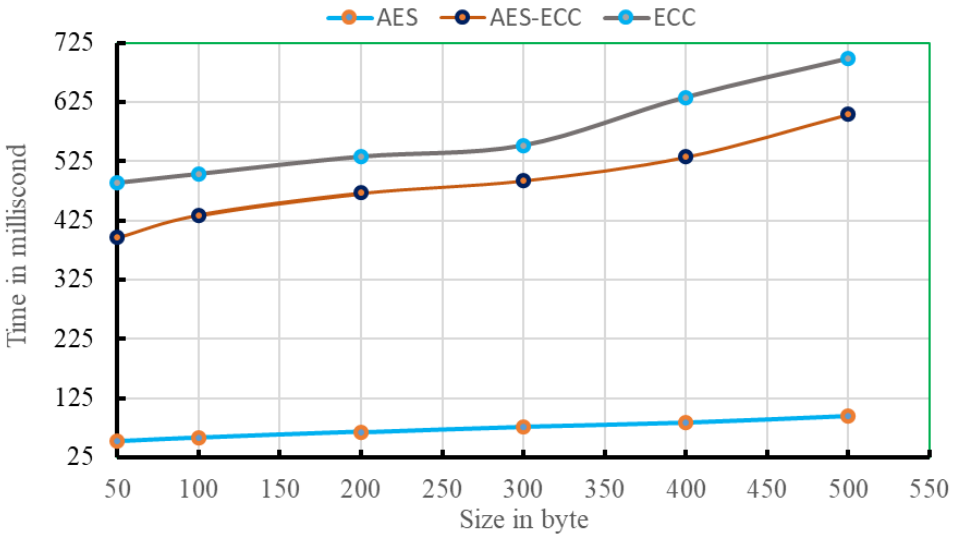


Figure 9. Comparison of Encryption and Decryption Times

7. Encryption Throughput

Throughput refers to how fast the encryption process occurs, and it can be calculated using the relationship (Orobosade & et al, 2020)

$$Encryption\ throughput = \frac{plaintext\ size\ in\ bytes}{encryption\ time\ in\ milliseconds}$$

Measuring encryption throughput aims to quantitatively evaluate the speed of data encryption, which helps in choosing the most appropriate encryption method. There is a direct relationship between encryption speed and power consumption, as faster algorithms usually require less power, which is beneficial for battery-powered devices. The results in Table 6 shown that the time rate for encrypting and decrypting data using AES is the fastest, followed by the hybrid AES-ECC algorithm, while the time rate for encrypting and decrypting data using the ECC algorithm is the slowest. In terms of encryption throughput, AES has the highest throughput, followed by hybrid AES-ECC, and finally ECC has the lowest throughput. This means that AES consumes the least power, followed by the hybrid AES-ECC algorithm, while the ECC algorithm consumes the most power. The bar chart is shown in Figure10 the throughput of each algorithm.

Table 6 Comparison of the Throughput of AES, ECC and AES-ECC Algorithm

Algorithm	Average time (ms)	Average Size (byte)	Throughput
AES	71.8	258.3	3.60
AES-ECC	488	258.3	0.53
ECC	568	258.3	0.45

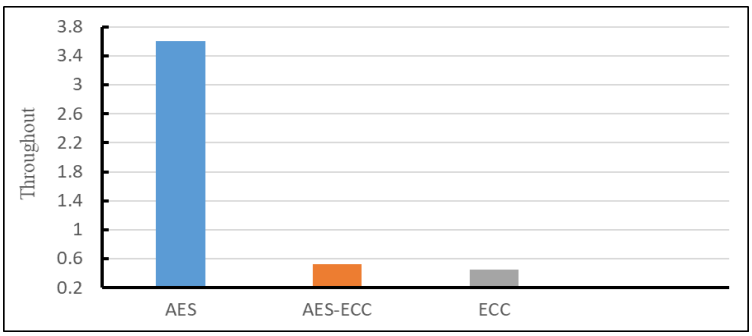


Figure 10. Comparison of the Throughput of AES-ECC, AES and ECC Algorithm

8. Application Security Features Status

Table 7 The Current Status of the Provided Security Features

Security feature	Status	Deception
Encryption of Transit Data	Yes	Data Encryption before Transmission
Data Hiding	Yes	Inclusion of Encrypted data in QR Codes
End-to-End Encryption	Yes	Achieved through Asymmetric keys in a Hybrid Algorithm.
Data Integration	Yes	Data Remains unread on the Server and is Protected with QR Codes
Multipl Authentication	Yes	Additional password
Automatic logout Mechanism	Yes	Activation of Automatic Logout
System Update	Yes	Update the System Periodically by Changing the Keys and Modifying the Algorithm

8.1 Comparison with Previous Work

A comparison was made between the system approach proposed in this study and the method used in (Boraiah, 2019), as described in the research paper titled 'Securing Cardless Transactions on Android App with ECC Algorithm and QR Code'. As shown in Table 8.

Table 8 Comparative Analysis of Our Study and Comparative Study (Boraiah, 2019) in Secure Cardless Transactions

Feature	Our Study	Comparative Study (Boraiah, 2019)
Algorithm Used	Hybrid Algorithm ECC-AES	ECC Algorithm Only
Encryption/Decryption Method	AES + ECC	ECC for Encryption, Private Key for Decryption
Key Storage	Embedded in Hybrid Algorithm	Public Key Shared, Private Key Held by Users
QR Code Storage Location	Mobile	Server
Authentication Methods	Multiple Authentications	Single Authentication
Usability	Easy to Use	Less User-Friendly

8.2 Limitation of the Proposed System

1. The proposed system is limited to securing data transfer and authentication. It may not adequately address important security concerns, such as maintaining data privacy.
2. The maximum data size that a QR code can hold is 500 bytes.

9. Discussion

For the purpose of verifying the implementation of the proposed system in accordance with the objectives, several practical tests were conducted. In Section 6.1, the mobile application interfaces were subjected to performance and usability evaluations. The evaluation of the mobile application interface highlighted its effectiveness and ease of use. Compatibility with various Android devices and positive comments from users confirmed that the design is easy to use. This indicates that our approach not only prioritizes security, but also emphasizes a seamless user experience. To evaluate the performance associated with hybrid encryption, Section 6.2 ran tests to measure encryption and QR code generation time, and QR code scanning and decryption time. The results showed that although the encryption time tends to be greater than the decryption time, it increases as the file size increases. Encryption and decryption times may vary based on factors including the complexity of the AES-ECC algorithm, text length, and Internet bandwidth. This emphasizes the importance of taking data volume into consideration when evaluating system performance and efficiency. Furthermore, embedding the keys within the hybrid algorithm and storing the QR code in the mobile phone simplifies the encryption and decryption procedures, enhancing user convenience without compromising security. To reduce security concerns and prevent possible violations, the system is updated periodically by changing the keys and modifying the algorithm. In the same context, test results showed that the capacity of the QR code is affected by the size of the data. This limitation may limit the amount of data that can be sent. In Section 6.5, a comparison between the efficiency of the AES-ECC algorithm and the AES,ECC algorithms is presented. The results indicated that although AES alone may provide faster encryption and decryption times, the hybrid approach outperforms both AES and ECC in terms of security. The added layer of encryption provided by the hybrid algorithm enhances data protection, making it the preferred choice for secure cardless transactions.

10. Conclusion and Future Work

The proposed system focuses on the security of cardless data transactions in mobile applications by implementing “Secure Cardless Transaction Mobile application using QR code and hybrid AES – ECC algorithm.” The proposed system aims to improve mobile phone services, especially in the field of secure data transfer and authentication. In addition to addressing growing concerns related to security breaches .Upon review of the literature, it was found that many technologies have come up with solutions for secure data transfer and authentication. However, our proposed approach can provide a viable solution. This approach takes advantage of hybrid encryption technology, specifically using the hybrid AES-ECC algorithm and QR code authentication. What distinguishes our approach is that encryption and decryption keys are combined within the hybrid algorithm, making it extremely difficult for potential attackers to decrypt, especially when supported by multiple authentication layers. Furthermore, the QR code is securely stored on the mobile device and scanned, thus enhancing security and reducing exposure to external threats and unauthorized access. This work could be expanded in the future, to enhance mobile banking, improve authentication methods and ease of use, along with incorporating additional features. This may include, generating a dynamic QR code instead of a static QR code, implementing a remote app lock feature, and adding digital signatures to ensure the authenticity of incoming messages.

References:

Adukkathayar, A., Krishnan, G. S., & Chinchole, R. (2015, July). Secure multifactor authentication payment system using NFC. In 2015 10th International Conference on Computer Science & Education (ICCSE) (pp. 349-354). IEEE.

Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. International Journal of Communication Networks and Information Security, 12(2), 256-272.

Al Imran, M., Mridha, M. F., & Nur, M. K. (2019, January). OTP based cardless transction using ATM. In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 511-516). IEEE.

Bommala, H., Kiran, S., Pujitha, M., & Reddy, R. P. K. (2019). Performance of Evaluation for AES with ECC in Cloud Environment. International Journal of Advanced Networking and Applications, 10(5), 4019-4025.

Boraiah, S. P. (2019). Secure Cardless Transaction Android Application using ECC algorithm and QR code (Doctoral dissertation, Dublin, National College of Ireland).

Deepa, M., & Parvathi, M. (2015). Adoption of Hybrid Cryptography in an Acknowledgement Based Intrusion Detection System for Manets. International Journal, 4(4), 79-82.

Hodowu, D. K. M., Korda, D. R. & Ansong, E. D. (2020). Enhancing Data Security in Cloud Computing with the Implementation of a Two-Level Cryptographic Technique Using AES and ECC Algorithms, Int. J. Eng. Res. Technol., 9(09):639-650.

Hosam, O., & Ahmad, M. H. (2019). Hybrid design for cloud data security using combination of AES, ECC and LSB steganography. International Journal of Computational Science and Engineering, 19(2), 153-161.

Isobe, T., & Ito, R. (2021). Security analysis of end-to-end encryption for zoom meetings. IEEE access, 9, 90677-90689.

Kader, H. M. A., Hadhoud, M. M., El-Sayed, S. M., & AbdElminaam, D. S. (2014). Performance evaluation of new hybrid encryption algorithms to be used for mobile cloud computing. International Journal of Technology Enhancements and Emerging Engineering Research, 2(4), 63.

Kumar, D., Agrawal, A., & Goyal, P. (2015, March). Efficiently improving the security of OTP. In 2015 International Conference on Advances in Computer Engineering and Applications (pp. 912-915). IEEE.

Kuppuswamy, P., & Al-Khalidi, S. Q. (2014). Hybrid encryption/decryption technique using new public key and symmetric key algorithm. International Journal of Information and Computer Security, 6(4), 372-382.

Lalem, F., Laouid, A., Kara, M., Al-Khalidi, M., & Eleyan, A. (2023). A novel digital signature scheme for advanced asymmetric encryption techniques. Applied Sciences, 13(8), 5172.

Lam, N. T. T. & Tra, L. T. (2021, September). Elliptic Curve Cryptography (ECC) Algorithm and Its Application in Building a Smart Auto Parking System, Master's Thesis, Advisor: Dr. Luu Thanh Tra.

Lee, H., Zhang, Y., & Chen, K. L. (2013). Journal of International Technology and Information Managemen t. Journal of International Technology and Information Management Volume, 22(4).

Mallouli, F., Hellal, A., Saeed, N. S., & Alzahrani, F. A. (2019, June). A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. In 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom) (pp. 173-176). IEEE.

Nie, J., & Hu, X. (2008, December). Mobile banking information security and protection methods. In 2008 International Conference on Computer Science and Software Engineering (Vol. 3, pp. 587-590). IEEE.

Nimmi, K., & Janet, B. (2018, December). An analysis of the balance between security and utility of mobile applications. In 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET) (pp. 1-4). IEEE.

Orobosade, A., Favour-Bethy, T. A., Kayode, A. B., & Gabriel, A. J. (2020). Cloud application security using hybrid encryption. Communications on Applied Electronics, 7(33), 25-31.

PointyCastle Library. "PointyCastle: Elliptic Curve Cryptography (ECC) Library," [Online]. Available: https://github.com/PointyCastle/pointycastle [Dec 23, 2020].

Rahman, M. H., Al-Amin, M., & Lipy, N. S. (2020). An investigation on the intention to adopt mobile banking on security perspective in 20angladesh. Risk and Financial Management, 2(2), p47-p47.

Rehman, S., Talat Bajwa, N., Shah, M. A., Aseeri, A. O., & Anjum, A. (2021). Hybrid AES-ECC model for the security of data over cloud storage. Electronics, 10(21), 2673.

Shahabuddin, M. D., Reasons why cyber security is important for banks, Cyber Secur. Solut. Serv. – IT Secur., 2018. [Online]. Available: http://www.infoguardsecurity.com/reasons-why-cyberse.

Sharma, N., & Bohra, B. (2017, February). Enhancing online banking authentication using hybrid cryptographic method. In 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT) (pp. 1-8). IEEE.

Singh, L. D., & Singh, K. M. (2015). Image encryption using elliptic curve cryptography. Procedia Computer Science, 54, 472-481.

Sudha, G., & Ganesan, R. (2013, April). Secure transmission medical data for pervasive healthcare system using android. In 2013 international conference on communication and signal processing (pp. 433-436). IEEE.

Vahdati, Z., Yasin, S., Ghasempour, A., & Salehi, M. (2019). Comparison of ECC and RSA algorithms in IoT devices. Journal of Theoretical and Applied Information Technology, 97(16), 4293.

Wahjuni, S., & Pristian, R. (2016, October). Android-based token authentication for securing the online transaction system. In 2016 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 174-177). IEEE.

Yu, J., & Nuangjamnong, C. (2022). The Impact of Mobile Banking Service on Customer Satisfaction: A Case Study of Commercial Banks in China. United International Journal for Research & Technology, 3(10), 43-64.

ISSN: 2663-5798 || Arab Journal for Scientific Publishing

الإصدار السابع – العدد تسعة وستون
تاريخ الإصدار: 2 – تموز – 2024م
www.ajsp.net

" تنفيذ تطبيق جوال آمن للمعاملات بدون بطاقة باستخدام رمز QR وخوارزمية AES-ECC الهجينة"

إعداد الباحثين:

نور جابر حمد[1]، عباس عبد العزيز عبد الحميد[2]، مظفر حسين علي[3]

[1]قسم هندسة الحاسوب/ كلية الهندسة/ الجامعة العراقية

[2]قسم علوم الحاسوب/ كلية العلوم / الجامعة المستنصرية

[3]قسم هندسة الشبكات/ كلية الهندسة/ الجامعة العراقية

الملخص:

لقد حظي استخدام الخدمات المصرفية عبر الهاتف المحمول بقبول واسع النطاق، وذلك بسبب الراحة وسهولة الوصول إليها عبر الهاتف المحمول. ومع ذلك، فإن الاعتماد المتزايد عليها من قبل المستخدمين كان مصحوبًا بتحديات أمنية مثل التصيد الاحتيالي واختراق البيانات. يعد ضمان أمان وسلامة نقل البيانات أمرًا بالغ الأهمية لبناء ثقة المستخدم. يعتبر تشفير البيانات أثناء المعاملات الحل الأمثل لأمن البيانات وسلامتها. ولتحقيق ذلك، نقترح نظامًا يستخدم تطبيقات الهاتف المحمول لنقل وتأمين المعاملات بدون بطاقة، وذلك باستخدام دمج مصادقة رمز الاستجابة السريعة مع خوارزمية AES-ECC الهجينة. تقوم هذه الخوارزمية بتشفير البيانات والمصادقة عليها عن طريق رمز الاستجابة السريعة. تتضمن الطريقة تشفيرًا هجينًا يجمع بين معيار التشفير المتقدم (AES) وتشفير المنحنى الإهليلجي (ECC) . بدلاً من استخدام مفتاح AES مباشرة للتشفير، يتم إنشاء مفتاح من خلال خوارزمية ECC . يتم فك التشفير باستخدام المفتاح الخاص لـ ECC .عندما يتلقى المستخدم رمز الاستجابة السريعة، يمكنه مسحه ضوئيًا للوصول إلى النص الأصلي. يتميز النظام المقترح بتخزين رموز QR على هواتف المستخدمين بدلاً من الخوادم، مع الاحتفاظ بمفاتيح التشفير مضمنة في الخوارزمية الهجينة لمزيد من الكفاءة والسهولة. تم اختبار كفاءة النظام المقترح باستخدام أحجام بيانات مختلفة، لقياس التشفير وزمن إنشاء QR، والوقت اللازم لمسح رمز QR وفك التشفير. بالإضافة إلى ذلك، قدرة رمز الاستجابة السريعة على تخزين البيانات. وأظهرت النتائج فعالية النظام، وسهولة استخدامه، وقدرته على نقل البيانات بشكل آمن.

الكلمات المفتاحية: التشفير، فك التشفير، خوارزمية (ECC-AES) الهجينة، رمز الاستجابة السريعة، الأمن.