Enhancing Security and Efficiency through QR Integration with Hybrid AES-ECC Algorithm in Mobile Apps for Cardless Data Transactions

Noor J. Hamad[*], Abbas A. Abdulhameed[**], Mudhafar H. Ali[***]

[*] College of Engineering, Al-Iraqia University, Iraq
Email: noorjhy@gmail.com
https://orcid.org/0009-0009-4550-3297

[**]Computer Science, University of Mustansiriyah, Iraq
Email: abbasabdulazeez@uomustansiriyah.edu.iq
https://orcid.org/0000-0002-1132-2756

[***] College of Engineering, Al-Iraqia University, Iraq
Email: mudhafar.ali@aliraqia.edu.iq
https://orcid.org/ 0000-0001-8447-5502

Abstract

Today, mobile banking services are widely accepted due to convenience and ease of use. However, increasing reliance on them has brought security challenges. Ensuring the security and integrity of transaction transmission is crucial for building user trust. Data encryption is the best solution for this. We propose a system that uses a mobile application to secure and authenticate cardless transactions by integrating a QR code with a hybrid algorithm combining Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC). The encryption key is generated through the ECC algorithm and decrypted using the ECC private key. The data is stored in the QR code, which users can scan to access the original text. To enhance ease of use and efficiency, the QR code is stored on users' phones instead of servers, with encryption and decryption keys embedded in the hybrid algorithm. Tests conducted with different data sizes measured encryption time, QR code creation time, scanning and decryption time, and the QR code's data capacity. Results showed the system's effectiveness, ease of use, and secure data transfer capability.

Keywords- AES, Authentication, Decryption, ECC, Encryption, QR Code.

I. INTRODUCTION

The spread of mobile banking has brought about a paradigm shift in financial transactions. These services offer convenience, ease of access, security and ease of use. Compared to traditional banking services, they are easier and safer to use, and provide 3A convenience (anytime, anywhere, anyhow). This makes them the preferred choice for users [1]. However, the increase in online transactions has created security challenges and threats such as phishing, fraud and data breaches, which undermine user confidence in financial services. Security is crucial, which necessitates robust systems to counter these threats. It has become necessary to develop solutions that focus on the security and integrity of transactions while maintaining user comfort. 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, requiring strong authentication and encryption mechanisms to protect sensitive data and mitigate risks. In response to these concerns, this research aims to develop a mobile application designed to secure cardless transactions by integrating QR code authentication with the hybrid AES-ECC algorithm, where the encrypted data is saved and stored in the QR code while the data is encrypted through the hybrid algorithm. This meets the urgent need for safe and easy-to-use solutions in the field of mobile banking. [2] [3] [4].

Studying the current literature is important to explore ideas and research related to our proposed system related to the advantages of mobile services, encryption processes, and authentication protocols. The following studies present works that are related to our proposed system, which can be divided into four sections:

- **Mobile banking features**

Many mobile applications facilitate banking services. [5] identified the main features of mobile banking and its suitability. Their study emphasized functions such as bill payment, balance enquiry, and check scanning, and demonstrated the ease with which users can engage in banking activities. via mobile phone, including text messaging and locating nearby ATMs or banks. Researchers noted the efficiency of mobile banking, which is represented by password-protected withdrawals . In the same context, in a study presented by

[6], they focused on mobile banking strategies and evaluated the extent to which these strategies were prepared to provide such services. The study emphasized the importance of security and technological progress in shaping the future path of mobile banking.

- **Security features of the banking application**

On the aspect of security features, [7] highlighted the importance of security concerns and the extent of their impact on mobile banking. They found that enhanced performance, stronger security measures, and increased trust positively affect mobile banking.

Similarly, [8] conducted a comprehensive functional analysis of various mobile banking applications, including Tez /Google Pay, Paytm, Paypal, and Bhim. Their study focused on evaluating the security features of these applications. It included mobile payment applications enhanced with a set of security measures, such as authentication protocols, fraud detection mechanisms, one-time transaction validation, TLS connection mechanisms, and user identification and password requirements to access the application. Table 1 provides an overview of security features across different banking applications.

TABLE 1. Security Feature Comparison of Banking Applications [8]

Basis for comparison	Paytm	BHIM	Tez/Google pay
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

- **Transfer data securely**

In their study, [9] described Security Challenges Facing Mobile Banking Proposed security measures included encryption techniques, identity verification, and digital signatures. Their study indicated that encryption
Especially using AES and ECC algorithms, it plays a vital role in ensuring data confidentiality and integrity. In their proposed approach AES-128 is used to encrypt the data, where ECC generates the encryption key, providing strong security and faster encryption and decryption. On the other hand, [10] A study on the use of symmetric and asymmetric encryption algorithms in cloud computing for data security. Their study confirmed that symmetric algorithms are faster and require less computational power than asymmetric algorithms, and that the 256-bit symmetric AES algorithm is preferred for banking services due to 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. Their results are shown in Table 2.

TABLE 2. Comparison between RSA and ECC [10]

Security	RSA	ECC
80	1024	160
112	2048	224
128	3072	156
256	15360	512

The study conducted by [11] on the application of ECC as a robust alternative to RSA in smart parking system. ECC technology 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. [4] proposed a system designed to counter cybercriminals who try to steal financial transaction information. The use of ECC and QR code technology in this 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. [12]. 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 guarantees the security of the ciphertext and provides an effective solution for secure data transmission in cloud storage technologies.

- **Authentication**

Due to increasing security breaches and fraud cases, secure user identity verification has become critical to the security and authentication of transactions. Accordingly, [13] presented a cardless transaction system, which uses one-time password (OTP) authentication, which is more secure than traditional password-based authentication. However, challenges related to data confidentiality and integrity remain. To reduce security concerns, [14] proposed an OTP authentication method using the one-time pad algorithm, by creating ciphertext unrelated to plaintext to provide a high level of security. However, its use via SMS has been identified as being vulnerable to attacks. For the purpose of minimizing security violations, [15] proposed the use of QR codes to transfer data via the bank server where the decryption and authentication process is managed exclusively by the bank. In contrast, [16] proposed an alternative approach using near-field communication (NFC), involving multi-factor authentication such as transaction details, facial recognition, a 4-digit PIN, and a smartphone with NFC capabilities. To address persistent malware threats, in the same context [17] proposed a solution that includes data encryption. The data can only be accessed by individuals who possess the corresponding private key. The proposed approach combines the elliptic curve coding (ECC) algorithm with quick response (QR) codes. The procedure involves mutual sharing of public keys between the financial institution and the customer. The transaction data is then encrypted using public and private keys. The encrypted data is stored within a QR code, which the customer can scan, to retrieve the data for decryption.

This paper is structured as follows: In Section II, Research methodology, in Section III Proposed approach, In Section IV Design specifications. In Section V Implementing the proposed system and VI Application Security Features Status, In Sections VII. Discussion. Finally, Conclusion and future work.

II. RESEARCH METHODOLOGY

Previous studies have shown that implementing a secure and efficient transaction exchange system requires strong authentication and data encryption mechanisms. Accordingly, we propose a new approach to data security and authentication by encrypting data using a hybrid AES-ECC algorithm and QR code authentication. The QR code is used to hide and retain encrypted data and store it in the mobile application.

Hybrid AES-ECC Algorithm

Recent reserchs have shown that combining ECC and AES algorithms into a hybrid encryption approach enhances the security and efficiency of cryptographic systems. This combination takes advantage of the strengths of both algorithms, leading to many diverse advantages. Including tighter security measures, improved operational performance, and the use of shorter encryption keys [12] [18] [19].

Hybrid ECC-AES Encryption/Decryption Methodology

The methodology includes the following steps:

1. A text file is selected for encryption and decryption operations.
2. The file is encrypted using Advanced Encryption Standard (AES). The AES key used for encryption is not used directly, instead, The AES key itself is encrypted using a key generated by the Elliptic Curve Cryptography (ECC) algorithm.
3. The encrypted file is uploaded to the server.
4. When the encrypted file is recovered, the recipient uses the corresponding ECC private key, linked to the ECC public key used during encryption, to decrypt the ECC-encrypted AES key.
5. When the AES key is retrieved, it is used to decrypt the content of the AES-encrypted text file.

This approach combines the efficiency of AES in handling large-scale data encryption with the security advantages provided by ECC for secure key transmission [20] [21] [22] [23] [24].

It is 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.

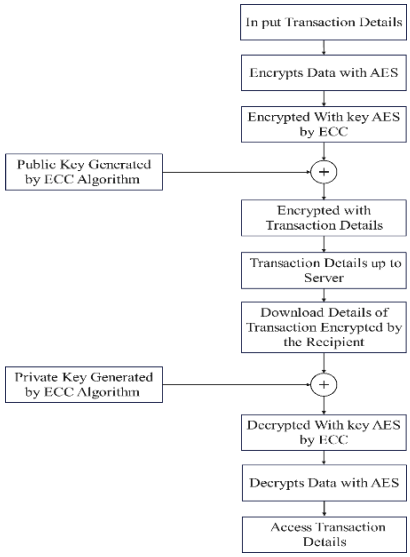

Fig 1 AES-ECC Encryption/Decryption

Operational Procedures for the Application

The hybridization algorithm consists of a 128-bit AES algorithm and a 160-bit ECC algorithm. The system generates a 128-bit random number (K) to serve as the key for the AES-128 algorithm. In return, a 160-bit random number is generated as the private key for the ECC-160 algorithm. This private key is used to generate a public key for AES key encryption. When you enter information, it is encrypted by 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. The operational procedures include two algorithms. These algorithms involve basic mathematical operations, including multiplication, addition, and subtraction, which are primarily concerned with manipulating points on a given elliptic curve within the application [24].

Algorithm 1: Encryption using the public key generated by ECC
Output: ciphertext (C)
$C = [(k * g), (m + k * x)]$
Where: K is a random integer within the range (1, p-1).
M: Represents the ciphertext (AES key) that will be sent and is a point on the curve.
C = (C1, C2)

Algorithm 2: Decryption using the private key generated by ECC
$m = C2 - [d * C1]$
The key K is extracted from M.
K is used to decrypt the ciphertext.

III. PROPOSED APPROAH

The proposed approach includes three distinct phases:

The first stage: registration and installation
It requires users to install the mobile application provided by the banking institution. Registration and access to the application requires providing basic details, including the password and email address specified by the bank.

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 means of transmitting the QR code to the intended recipient. Before sending data an additional layer of authentication is presented, in the form of an additional password.

The third stage: receiving and decoding data
Upon receipt of encrypted data, the designated recipient 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 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.

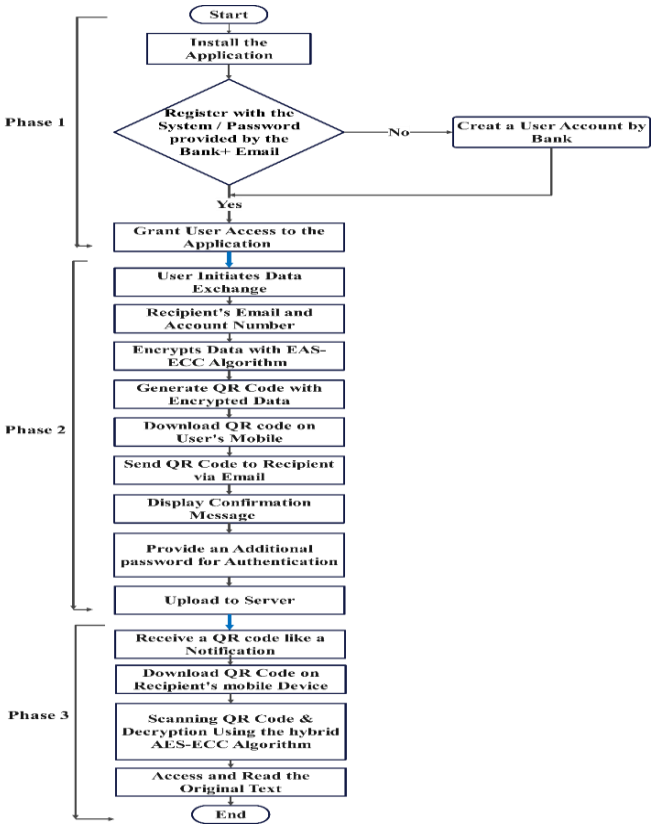

Fig 2 Flowchart Depicting the Basic Stages of the Proposed Approach

IV. DESIGN SPECIFICATIONS

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

System Architecture

The architectural framework includes three components as shown in Figure 3: users, financial institutions, and Firebase servers. These items are connected through a central Firebase server. The "Users" segment represents users who use mobile applications to transact with banking institutions. While 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.

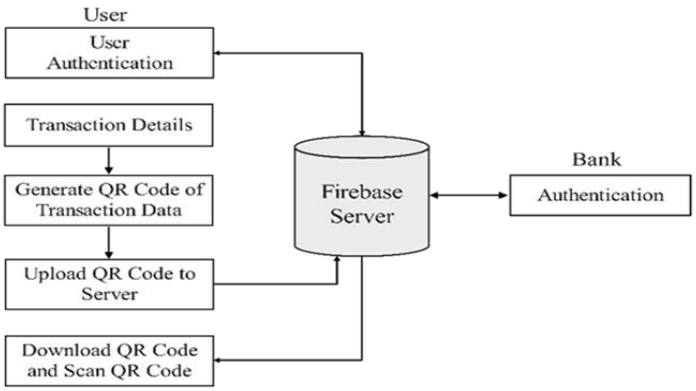

Fig. 3. Architecture diagram System

Proposed system methodology

The user creates an account in the application by providing basic information for authentication. Once the account is created, Firebase creates a unique ID for each user on the server. After logging in, start the process of exchanging transactions. The data is encrypted using the AES algorithm, where the public key is generated by the ECC algorithm, which in turn encrypts the AES key. These keys are included in the hybrid algorithm. The encrypted data is then converted into a QR code. When the QR code is sent to the intended recipient via their email. after logging into the app, the recipient scans the QR code and decrypts the data, using the private key generated by the ECC algorithm embedded in the hybrid algorithm. This private key retrieves the AES key, which is then used to decrypt data encrypted by the AES algorithm. Figure 4 shows a comprehensive step-by-step flowchart of the processes, from initial account creation and data encryption to the final step of decryption by the intended recipient. This visual representation, along with Figure 3, contributes to a comprehensive understanding of the application design and operation.

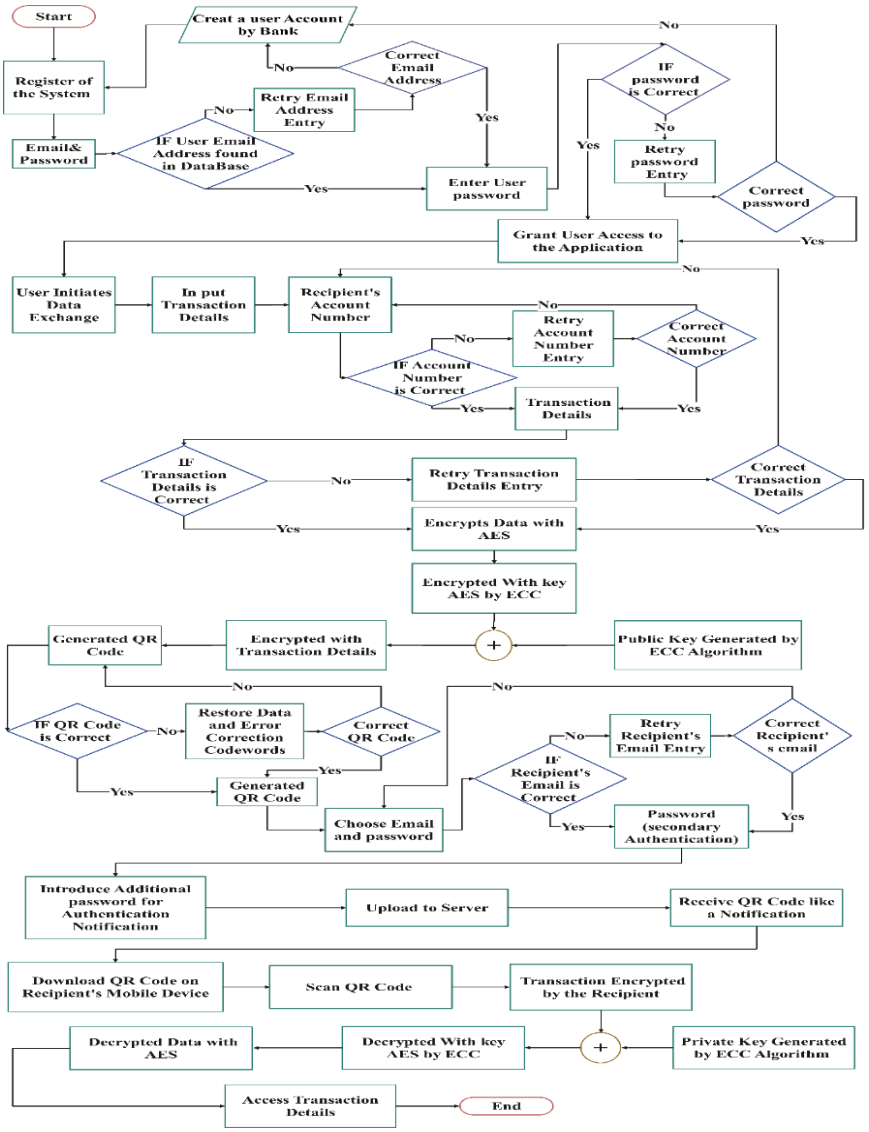

Fig 4 Flow Diagram of the Application

Components of the proposed system

The operation of the system depends on three main components,

1. Mobile application interface

This component allows users to participate in cardless transaction exchanges. It includes several interfaces, such as registration and login, QR code scanning, encryption and decryption options, payment confirmation, viewing transaction history, and configuring settings. Figure 5 contains visual representations of the interface design.

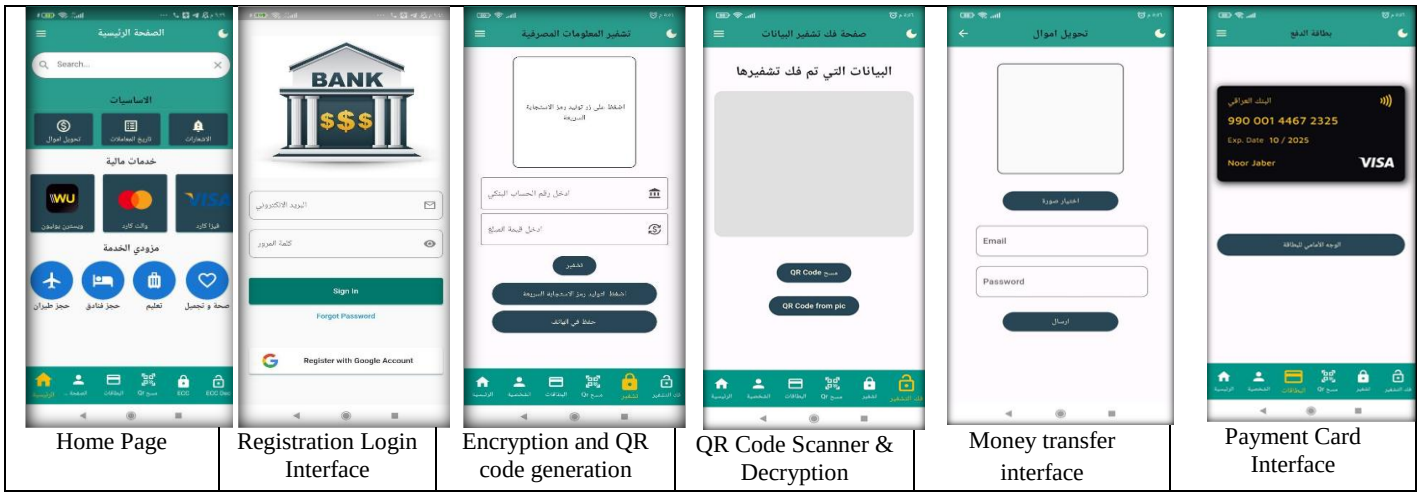

Fig 5 Includes visual representations of the interface design

2.Create and scan QR code

The application is able to generate and scan a QR code. Users can easily enter encrypted transaction details and generate a QR code. It is 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 transfers

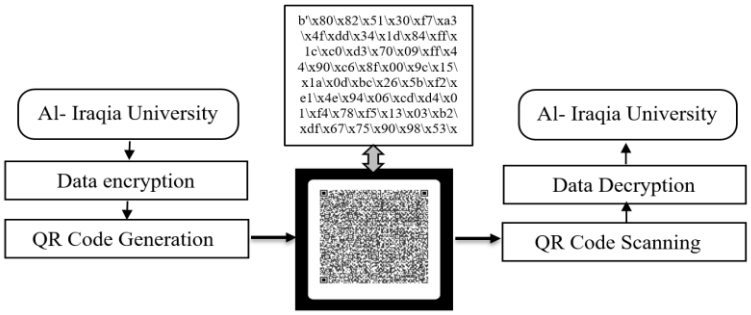

Fig 6 QR Code Creation and Secure Data Transfers

3. Hybrid AES-ECC algorithm

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

V. IMPLEMENTING THE PROPOSD SYSTEM

The proposed system runs on the Android operating system. Visual Studio Code is used to edit the code. Flutter facilitates GUI development in the Dart language. The Pointy Castle library [25] is used to execute code, including implementing the hybrid encryption algorithm. QR code generation is based 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 the email and password credentials specified by the bank.

System Testing and Evaluation

The testing process included five main sections, each focusing on a specific aspect of implementation:

1. Testing and evaluating the mobile application interface
Tests included compatibility with different Android devices and the application's ability to adapt to different screen sizes. A survey was conducted on ten users to collect their opinions about the application. 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.

2 Measure encryption time and generate QR code
Seven files with different data sizes were selected for the purpose of measuring the encryption time and creating the QR code. As shown in Table 3.

TABLE 3: Time required to encrypt data and generate 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

3. Measure the time of QR code scanning and decryption
The time required to scan and decrypt the same data sizes presented in Section 2 was calculated as shown in Table 4.

TABLE 4: QR code scanning and decryption time

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 show that the data encryption time is greater than the decryption time. The time it takes to complete these tasks increases as the file size increases, as shown in Figure 7. It is worth noting 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.

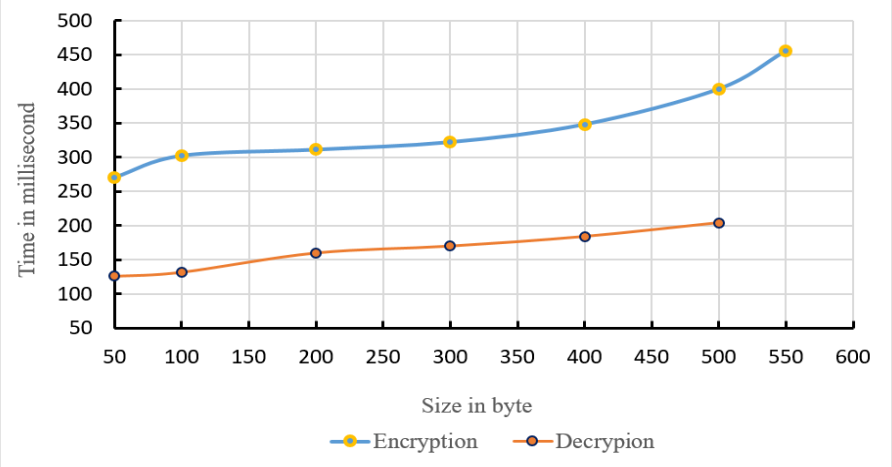

Fig 7 Encryption and Decryption Time in the Hybrid AES-ECC Algorithm

4. Testing the ability of the QR code to accommodate files

The test results shown in Table 4 showed that the maximum data size that a QR code can scan is 550 bytes. Figure 8 shows the different formats of QR codes containing encrypted data of different sizes.

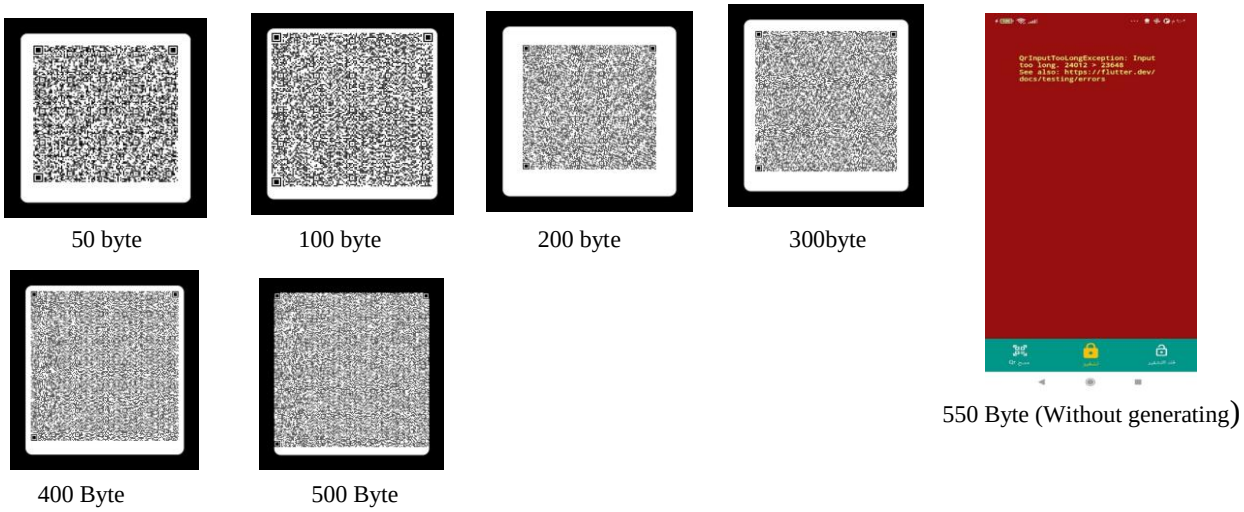

Fig 8 Various Encrypted QR Code Formats

5. Testing the efficiency of AES-ECC versus AES/ECC hybrid encryption

The test focused on comparing the time of encrypting and decrypting data using hybrid AES-ECC versus the time of encrypting and decrypting the same data sizes when using ECC-384 bits and AES-192 bits independently. 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 shown in Table 5 show that when ECC is used to encryption and decryption data, it takes longer compared to the hybrid AES-ECC algorithm. Although the AES algorithm takes less time, this algorithm alone is not as secure as the hybrid system. The reason is that the hybrid system adds an extra layer of security. If a hacker can decrypt one level of encryption, it will be difficult for him to decrypt the second level using the same algorithm. Figure 9 shows the comparison of total encryption and decryption times between AES, ECC, and the AES-ECC hybrid model.

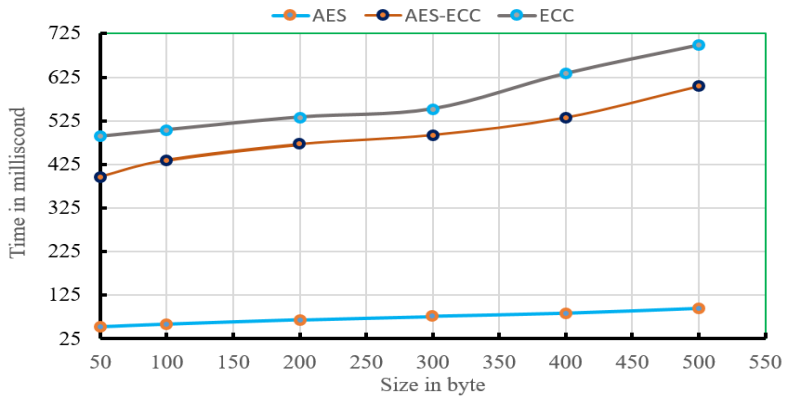

Fig 9 Comparison of Encryption and Decryption Times

VI. APPLICATION SECURRITY FEATURE STATUS

TABLE 6. 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
multiple authentication	Yes	additional password
automatic logout mechanism	yes	Activation of automatic logout
Automatic Logout	yes	Activate Automatic Logout
Update	yes	By changing the keys and modifying the algorithm

Comparison with previous work

A comparison between the proposed system and the method used in [4] was prepared in the research paper entitled “Securing Cardless Transactions on Android Application Using ECC Algorithm and QR Code”. As shown in Table 7.

Table 7. Comparative Analysis of Our Study and Comparative Study [4] in Secure Cardless Transactions

Feature	Our Study	Comparative Study [3]
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	Multiple	Single Authentication

Methods	Authentications	
Usability	Easy to Use	Less User-Friendly
Automatic Logout	Yes	No

Limitations of the proposed system

1- It is limited to securing data transfer and authentication.

2- The size of the data that a QR code can accommodate is 550 bytes, maximum

VII. DISCUSSION

The ability to implement the system as required was verified by conducting several practical tests. In Section VI, first, the mobile application interfaces are tested for performance and usability. Tests have shown its effectiveness and ease of use. Compatibility with different Android devices and positive comments by users confirmed that the design is easy to use. Which indicates that the design not only prioritizes security but also emphasizes a seamless user experience. To evaluate the performance associated with hybrid encryption, Section VI conducted second, 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 account 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. 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 VI, a comparison between the efficiency of the AES-ECC algorithm and the AES, ECC algorithms is presented. Revealed The results indicate 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 security and protection, making it the preferred choice for cardless transaction security.

VIII. CONCLUSION AND FUTURE WORK

The proposed approach focuses on the security of cardless transactions in mobile application by “enhancing security and efficiency by integrating QR with hybrid AES-ECC algorithm in mobile applications for cardless data transactions.” The proposed system aims to improve mobile services, especially in the field of transportation. Secure data and authentication, addressing growing concerns regarding security breaches. After reviewing previous studies, it was found that many technologies have come up with solutions for secure data transmission and authentication. Our proposed approach can provide a viable solution. It takes advantage of hybrid encryption technology, specifically using the hybrid AES-ECC algorithm and QR code authentication. Our approach has the advantage that encryption and decryption keys are combined within the hybrid algorithm, making it 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 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, and implementing a remote app lock feature.

REFERENCES

[1] N. K. and B. Janet, "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), Kottayam, India*, 2018, pp. 1-4, doi: 10.1109/ICCSDET.2018.8821080.

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

[3] T. Isobe and R. Ito, Security Analysis of End-to-End Encryption for Zoom Meetings, IEEE Access, 9(2021) 90677-90689.

[4] S. P. Boraiah, Secure Cardless Transaction Android Application using ECC algorithm and QR code, master's thesis, National College of Ireland, Dublin, 2019.

[5] K. Nimmi, B. Janet, An analysis of the balance between security and utility of mobile applications, International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), IEEE, 2018.

[6] J. Nie, X. Hu, Mobile Banking Information Security and Protection Methods, in : IEEE International Conference on Computer

Science and Software Engineering, Wuhan, China, 2008, pp. 587-590.

[7] F. Mallouli, A. Hellal, N. Sharief Saeed, F. Abdulraheem Alzahrani, A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms, in : IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 2019, pp. 173-176.

[8] M. A. Imran, M. F. Mridha, M. K. Nur, OTP Based Cardless Transction using ATM, in : IEEE International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 2019, pp. 511-516.

[9] S. Wahjuni, R. Pristian, Android-based token authentication for securing the online transaction system, in: IEEE International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2016, pp. 174-177.

[10] F. Mallouli, A. Hellal, N. Sharief Saeed and F. Abdulraheem Alzahrani, "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), Paris, France*, 2019, pp. 173-176, doi: 10.1109/CSCloud/EdgeCom.2019.00022

[11] N. T. T. Lam and L. T. Tra, "Elliptic Curve Cryptography (ECC) algorithm and its application in Smart-Auto Parking Systems," presented in *2021 IEEE Conference on Intelligent Transportation Systems*, 2021.

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

[13] M. A. Imran, M. F. Mridha and M. K. Nur, "OTP Based Cardless Transction using ATM,"in 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 2019, pp. 511-516, doi: 10.1109/ICREST.2019.8644248.

[14] S. Wahjuni and R. Pristian, "Android-based token authentication for securing the online transaction system," in 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2016, pp. 174-177, doi: 10.1109/ICTC.2016.7763462.

[15] D. Kumar, A. Agrawal and P. Goyal, "Efficiently improving the security of OTP," in 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 912-915, doi: 10.1109/ICACEA.2015.7164835.

[16] A. Adukkathayar, G. S. Krishnan and R. Chinchole, "Secure multifactor authentication payment system using NFC," in 2015 10th International Conference on Computer Science & Education (ICCSE), Cambridge, UK, 2015, pp. 349-354, doi: 10.1109/ICCSE.2015.7250269.

[17] B.S.Ponnsamudra “Secure Cardless Transaction Android Application using ECC algorithm and QR code,” M.S. thesis, National College of Ireland, Dublin, 2019.

[18] H. M. Abdul Kader, M. M. Hadhoud, S. M. El-Sayed, D. S. AbdElminaam, 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 (2014)

[19] M. Deepa, M. Parvathi, Adoption of Hybrid Cryptography in an Acknowledgment-Based Intrusion Detection System for MANETs, International Journal, 4(4) (2015).

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

[21] G. Sudha, R. Ganesan, Secure transmission medical data for pervasive healthcare system using android, in: IEEE International Conference on Communication and Signal Processing, Melmaruvathur, India, 2013, pp. 433-436.

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

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

[24] O. Hosam, M. H. Ahmad, Hybrid design for cloud data security using a combination of AES, ECC, and LSB steganography, Int. J. Comput. Sci. Eng., 19 (2) (2019) 153-161

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