National College of Ireland

Secure Cardless Transaction Android Application using ECC algorithm and QR code

MSc Internship
Cyber Security

Sumana Ponnasamudra Boraiah
X18100147

School of Computing
National College of Ireland

Supervisor: Mr Ben Fletcher

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sumana Ponnsamudra Boraiah
Student ID:	X18100147
Programme:	MSc in Cyber Security **Year:** 2018/2019
Module:	Academic Internship
Supervisor:	Ben Fletcher
Submission Due Date:	15/12/2019
Project Title:	Secure Cardless Transaction Android Application using ECC Algorithm and QR code

Word Count:4584 **Page Count: 15**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ……

Date: 15/12/19

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Secure Cardless Transaction Android Application using ECC algorithm and QR code

Sumana Ponnasamudra Boraiah
X18100147

Abstract

As financial sector grows, the online exchange or transactions services are tremendously expanding over the internet between customer and organizations. As a solution to the ATM skimmers or card details theft, organizations introduced cardless ATMs which accessed through android application. In the interim, ATM banking using android applications has been dogged by cyber criminals for stealing customer's transaction details by personating android applications on their own mobile, when the login credentials are leaked. Therefore, we propose a system, where the transaction details will be in encrypted format so the one who has respectable key could view the data. This is achieved by ECC asymmetric algorithm, where public keys are interchanged between bank and customer, further the data is encrypted using public key and private key. Now, the customer receives QR code with encrypted transaction details. The QR code is scanned using android application to view data. Here, asymmetric keys are generated using ECDSA, both encryption and decryption are carried out by using ECIES standard.

1 Introduction

As online banking is going popular due to their comprehensible services, which increased in the practice of android applications for payments and other featured services compared to traditional banking. In the field of banking and finance, data security plays a major role while exchanging customers data. Uncertainly, intruder gains authorization to the system by phishing attack and distort or erase confidential data. The bank or financial organizations be supposed to concern about cyberattacks in many terms as personal and private data, digital money, fraudulent, trust of consumers, and time.**(MD Shahabuddin, 2018)** Banks should provide guarantee to the customers by providing secure transmission of data and indestructible system. Here comes the security fact that is encryption of data before transmission between bank to server and server to client over android applications.

When the bank customers perform payment with transaction details, how this transaction details are secured from the intruder attacks?
The only answerable solution to this is data encryption. Encryption is a process of translating data into incomprehensible to prevent from exploitation and make available absolute data to a person who has the key by decryption process. Majorly the encryption can be applied on

data-at-rest and data-in-transit. The data stored on memory card, systems, servers, or external hard-disc signified as data-at-rest. The data-at-rest can be protected by disk encryption and file encryption. The data forwarding from one application to server and server to another application through internet is data-in-transit which can be encrypted in two ways: transport-layer encryption and end to end encryption. The end to end encryption is performed on data shared between sender and receiver when the services are not trustworthy.**("What Should I Know About Encryption? | Surveillance Self-Defense," n.d.2018)** The android banking applications require end to end encryption to accomplish secure communication to counteract data breach. In fact, banks implementing security by encrypting digital information using symmetric encryption standards. The symmetric encryption is employed at transactions to prevent client's or bank's identity theft. Additionally, user authentication and authorization to protect against cyber-attacks.

The base logic under the end-to-end encryption is asymmetric encryption, which makes data protected by allowing it to read only by the sender and the receiver. The chat applications like WhatsApp uses the end to end encryption for communication, while data passed through the server but unread by the server. The asymmetric encryption must be established when you do transactions through online by entering card details. **(information and VoIP, n.d.).** It would be easy for hackers to hack mobile phones compared to credit or debit cards. Unlike the chips in the smartcards, the software on mobile phones is much vulnerable to attack. The number of entities embedded in the mobile for payments adds the load to the security of the mobile. With growing vulnerability and attacks, software developers and security professionals should commit to the multi-level security of the system.

This research work potentially influences banking and financial organizations in optimizing online service offerings with secure data transfer and authentication. More concerned about attacks on automated teller machines (ATMs). The thieves or hackers uses following attacks to steal the money**("The Latest Threats to ATM Security | SecurityWeek.Com," n.d.)**:

- Inserts skimmers in physical devices of the ATM machine to capture the card details of the customer.
- By using malwares like ATMitch and Ploutus-D in ATM machines to take control over ATM servers.

With above discussion, this paper focus on how to avoid payment card details theft by using secure transaction using better asymmetric encryption technology to furnish strong authentication.

The paper structured as: in section 2, related work explained which helps in findings. In section 3 the Research methodology describes the technologies involved in this thesis work and Requirement and design specification explained in section 4. Implementation and evaluation methodology are explained to test the approach for better enhancements in section 5 and 6. At last, conclusion and future work is explained.

2 Related Work

The following papers give background research on the data encryption, authentication and existing transaction applications.

2.1 Security features of banking applications

There are numerous supporting mobile applications for digital transaction. Individual can use the smart phones to pay the bills and send money to any corner on the world by following simple steps. The security features of these application are very important as the threats are increasing. The author **(Li, 2013)** explains about the Perceived risk that is used in explaining user decision making intentions. Perceived risk comprises of financial risk, performance risk, privacy or security risk. Most of the time, users are afraid to use mobile banking due to fear of losing money due to internet disconnection or server breakdown either from user or vendor side which refers to both financial and performance risks. Most of the online shopping applications requires user name, address, phone number. Which leads to fear of losing control on the personal information. All these risk factors are decreasing the faith and trust on the mobile banking service and it would be difficult to accept the service with all these risks.

The functionality analysis of mobile banking applications with security consideration is described by**(K. and Janet, 2018)**. There are applications like Tez/Googlepay, Paytm, Paypal and Bhim includes security attributes as authentication, Machine learning based fraud detection engine, pin number and OTP for transaction, TLS connection mechanism, user's ID and password to access the application. Below table shows comparison security features of banking applications.

Basis for comparison	**Paytm**	**BHIM**	**GooglePay 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

Table 1: Security feature comparison of banking applications **(K. and Janet, 2018)**

2.2 Secure Data Transmission

Mobile Banking is facing more complex security problems than of internet banking. **(Nie and Hu, 2008)** describes issues in mobile banking and information protection methods. The attacker aims to damage transaction systems by SMS Denial of Service Attack and Virus attack through wireless network or Bluetooth. By implementing encryption technology to protect data privacy by using symmetric encryption algorithm AES and asymmetric encryption algorithm ECC. The data is encrypted using AES 128-bit algorithm and encryption key is generated using ECC to encrypt which gives better security and faster encryption and decryption. In order to infringe the system hacker must experiment on AES, which is challenging with existing technology. If the attack is on session key of ECC, they might encounter of ECDLP.

A study on symmetric and asymmetric algorithm performed by **(Mallouli et al., 2019)**, clarifies that in the emerging cloud computing era the data security while transmitting. To achieve confidentiality, authentication, data integrity, non-repudiation and access control, both symmetric and asymmetric algorithm widely supports the cloud computing. The symmetric algorithm is faster than asymmetric algorithms since they need less computation power. In baking services, symmetric AES 256-bit algorithm is used for encryption. For the smaller devices like mobiles, symmetric key length is large to handle suggested to use asymmetric algorithms for android devices is preferred.

As an application to medical sector, which requires the authentication and access control of the healthcare systems. The authors **(Sudha and Ganesan, 2013)** mobile healthcare solution to patients, along with security using Elliptical Curve Cryptography (ECC) algorithm to authenticate user's data.

2.3 Authentication

In today's scenario, online banking or transaction the first stage is authentication. As a solution to authentication vulnerabilities, there many approaches are available in the current system.**(Imran et al., 2019)** proposed cardless transaction system using OTP authentication. The OTP authentication is more vulnerable than password and biometric. The bank cannot determine whether the person one who using application is legitimate or not. The drawback of the system is lack of data integrity and confidentiality of data while transaction in progress.

The authors **(Wahjuni and Pristian, 2016)** has developed an online transaction system for android devices by furnishing token authentication. The author used One-time pad (OTP) algorithm to generate authentication token. Since OTP algorithm is consider as very safe encryption algorithm with short key which is equal to plaintext length and produces the ciphertext with no relation to the plaintext. Because of recent attacks on OTP, it is no longer counted as secure to use authentication over SMS. Hence, to overcome this problem **(Kumar et al., 2015)** proposed Quick Response (QR) as a solution. In their approach, the transaction

details are converted to QR code and sent to client by bank over the server. The bank provides unique decoder, which cannot be modified without consultation of provider.

Another approach for payment system contributed by **(Adukkathayar et al., 2015)**using NFC. Before the method invoked for NFC functionality, the user must submit the transaction details. On confirmation of the receiver, the transaction process is proceeded by face recognition and PIN verification to provide multifactor authentication. In their system, NFC functionality is protected against attack from malicious software by isolating its method from rest of the functionalities. However, malware attacks are constantly sprouting to break structure of the system.

To enhance the authentication, the authors **(Sharma and Bohra, 2017)** introduced a five-phase authentication system, which involves MD5 algorithm for message digest, RSA algorithm for data integrity and privacy. By linking QR code with Unique Id (UID), the authentication is achieved.

3 Research Methodology

Hence from the above study, we recognize that the transaction system must include authentication and data encryption. To complete these requirements, we proposed a novel approach that improves current transaction security. The proposed system contains mutual authentication and data encryption using the Asymmetric algorithm. The QR code is used to hide and hold data, which then stored in the server.

3.1 ECC Algorithm

Elliptical Curve Cryptography is an Asymmetric key algorithm that is considered as unbreakable cryptosystem and requires short keys that are preferred for mobile devices. The key generation involves both public key and private key. ECC is advantageous in internet-based applications like online banking and e-business where huge transactions are on demand. It is especially for mobile devices since mobile requires algorithms that consume low memory and power consumption as mentioned in **(Bafandehkar et al., 2013)** .

Algorithm Execution steps:

1) Initialize ECC curve with parameters
2) Generate keypair using ECDSA standard
3) Get the private key and public key
4) Initialize encryption with transformation algorithm ECIES
5) Generate cipher according to the public key and private key
6) Initialize decryption with transformation algorithm ECIES
7) Decrypt the cipher to plaintext

3.2 Proposed Method

Stage 1: Registration and Key generation

- The first step is to install the secure application in the smartphone and then register with the system by entering details such as username, password, and email address. The user details stored in the server.
- After login, generate the Cryptographic key pair then stored them as the public key and private key files in mobile memory.
- When the transaction or data exchange is required, choose a bank or another user from the list. Send a request to exchange public keys through email.

Stage 2: Encryption

- The bank identifies the user.
- The bank or another user will encrypt transaction data using public key received and own private key. The QR code is generated for the cipher and uploaded to the server, which will be available only to the legitimate user.

Stage 3: Decryption

- The user views the QR code from the server and scan.
- Choose the bank's public key and private key to decrypt.
- If the keys match the transaction data is decrypted.

4 Design Specification

This paper proposes an android application, in which the user can register and take advantage of the application. The design of our method introduces details to step by step operations. This android application targets to secure data transmission by encrypting data and allow users to communicate with each other. The design of the application is explained with the help of a flow diagram and system architecture.

4.1 System Architecture

In figure 1 functional process of the system is represented. The system consists of customers, bank organization and Firebase servers. Here the application from both ends interfaced through a centralized Firebase server for messaging and data exchange.

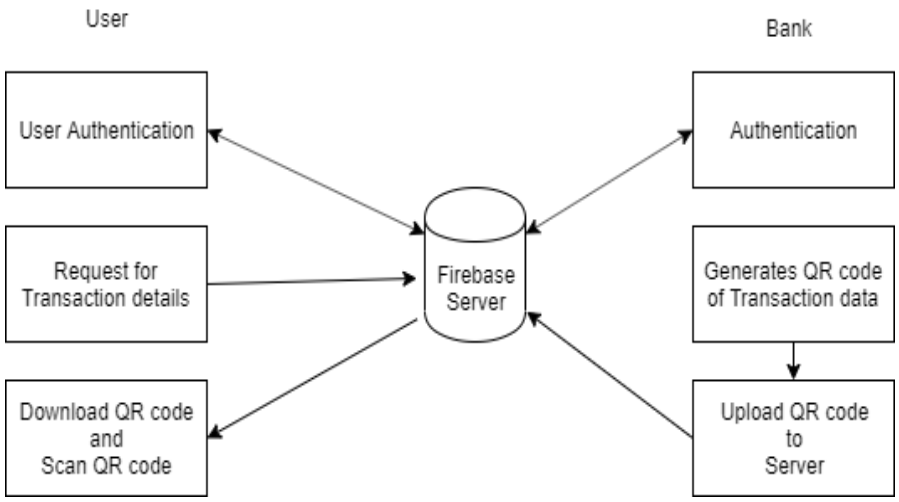

Figure 1: Architecture Diagram of System

4.2 Structure of the Proposed System

The system has no different applications for encryption and decryption. To start up with the system, the user must create an account with the application using an email id and password. Immediate after creating account in application, at the server end firebase creates unique id for each user. The user actions are further handled and tracked by this user id.

After login, the user will enter to generate the keypair of the public key and the private key. The keys will be stored in the internal storage of the android device, these keys are picked up in later stages. The user must select receiver from the list shown to go further with the application functionalities. For the chosen user, the customer can have message conversation or send secure data by encrypting data using the generated private key and shared public key of another user. The cipher data is further converted to QR code image and uploaded to the server. The receiver will download the QR code and follows same procedure as the sender to view the secret data.

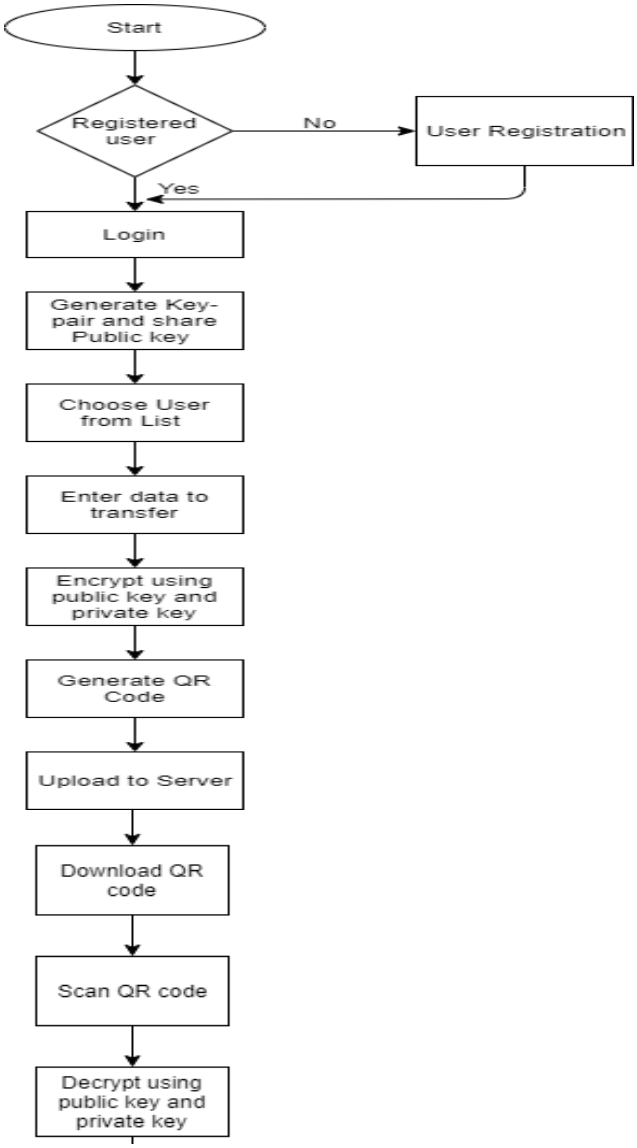

Figure 2: Flow diagram of the application

5 Implementation

The proposed system is implemented by developing an android application. This section narrates the implementation of the application from the core.

5.1 Cryptographic Specifications

For the cryptographic security design of an android application, the android platform has built-in security features. Which supports data isolation, encryption and data protection using cryptography. The Elliptical Curve Cryptography (ECC) implementation supported by java with by forcing SunEC provider. The available algorithm parameters, Key Factory, and key generator is EC and multiple signature with SHA message digest. The ECC algorithm is

implemented using the SpongyCastle provider **("Spongy Castle by rtyley," n.d.)** and java security APIs. In this paper, we used "**secp192r1**" to generate the Elliptical curve. ECDSA (Elliptical Curve Digital Signature Algorithm) is used to generate a 192-bit key which is also included in IEEE 1363 and ISO/IEC 15946-2 standards.
(Gayoso Martinez et al., 2005). For encryption and decryption, ECIES (Elliptical Curve Integrated Encryption Scheme) is used which is introduced by Abdalla, Bellare, and Rogaway in 1998 as stated in **(Gayoso Martinez et al., 2005)**

5.2 Server Implementation

This application is operated by firebase services for login authentication (Auth), cloud messaging, and Real-time database **("Firebase," n.d.)**. Firebase provides a platform for mobile and web application development with services and tools. The application creator must create an account with firebase and make the application available with all the services provided by the firebase. The IDE android studio must be plugin with the firebase and connected to it through developer account. The cloud messaging service is used to chat customer with the intended bank or another user. While chatting the conversations are stored in the firebase storage with specific chat id, where the id entitled with both the user's identity. The authentication with the application is completed by email and password method. The authentication service creates customer ID and made use of it for other functional features. The firebase provides Real-time database, we made use of this to store QR code images. The QR code images are made retrieved to scan through application when it requires.

5.3 Android Application Implementation

The java programming language is used in android activity classes and certain events. The GUI of application is designed using XML layouts, which consists of tags and properties. The data transmission occurs through firebase, which supports QR code storage by proving Real-time database. This section demonstrates the functionalities and performance of the proposed system which involves some phases. The users must install the secure application into their smartphones. The first phase involves the user registration. The user's details are stored in the server, further, they used it for authentication. For the user, friendly usage of the application includes forgot password and signup options.
After login one can generate a cryptographic keypair, includes public and private key saved in phone storage. The user will request the bank or another user to exchange the public key through email or another medium. The bank or another user will select that user from the list, the next step to encrypt the secret data by choosing their private key and user's public key. The encryption and decryption processed using ECC algorithm, which executed parallelly at back of the user interface. Later QR Code generated for the encrypted data is uploaded to the server as shown in figure 3. Now the user will log in to the application and view the QR Code by entering the password. If three times password entered wrong, the data will be erased permanently from the server, which protects data from attackers. The QR code will not be

saved in the smartphone, can only be viewed from the server. The scanned data is decrypted further using shared public key and own private key as shown in figure 4.

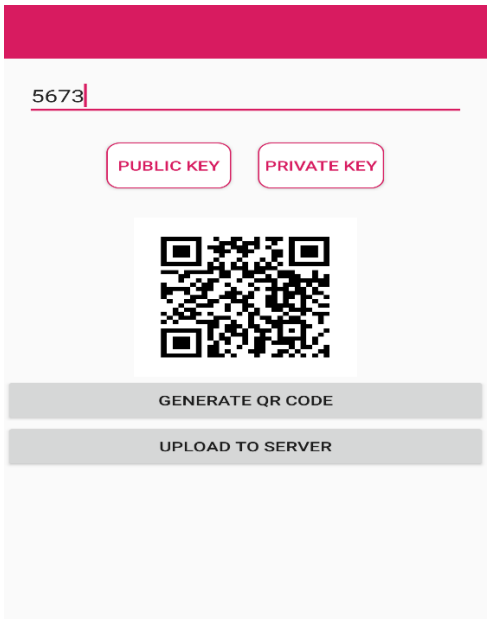

Figure 3: Encryption and QR code generation

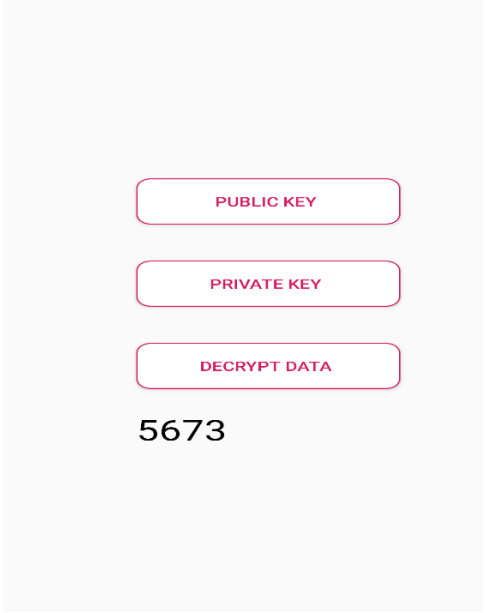

Figure 4: Decryption

6 Evaluation

The evaluation of the proposed method is verified based on the security features as a checklist and the cryptographic functions performance made by checking time elapsed to calculate logical output. The strength of the system is compared with existing work.

6.1 Security Feature

Security Feature	Status	Description
Transit Data Encryption	Yes	Data is encrypted before transmission
Data Hiding	Yes	The encrypted data is converted to QR Code
End-to-End Encryption	Yes	Achieved by Asymmetric key
Data Integrity	Yes	Data is not read in server and protected with QR code
Multi factor Authentication	No	Required to establish during transaction

Table 2: Status of the security features furnished

6.2 Security Function Performance test

This section shows the result of the time taken by the cryptographic functions at different test runs of the application.

Function	Execution 1 (ms)	Execution 2 (ms)	Execution 3 (ms)	Average (ms)
Key Generation	301	318	305	308
Encryption	43	51	48	47.33
Decryption	7	16	8	10.33

Table 3: Time elapsed by the functions

6.3 Comparison with previous work

Title of the paper	Method proposed	Drawback
OTP Based Cardless Transaction using ATM	Focused on authentication, using OTP and BPIN	No protection, when the login credentials are leaked, and OTP is vulnerable to attacks
Secure Cardless Transaction Android Application using ECC Algorithm and QR code	Focused on End-to-end encryption using ECC algorithm, Authentication with Asymmetric key, Data Hiding using QR code	Not flexible to use, users need to have more knowledge on asymmetric key exchange.

Table 4: Comparison between previous approach

6.3 Limitation of the proposed System

The proposed system limited to secure data transmission and authentication. The transaction data must be protected in many security terms of privacy and data consistency.

6.4 Discussion

To verify the proposed system is implemented according to the aim, we followed evaluation by considering a few factors. The work is compared with previous solutions to make sure that the system is providing a better solution to the existing problems. To check the performance of the code, test in subsection 6.2 is made and security checklists in subsection 6.1 are made by taking consideration of future attacks or enhancement. The application aims mainly two purposes. The purpose of the secure transaction over a server and meanwhile the authentication should be established. This is successfully established using android application, ECC algorithm and QR code technology. We observe that the approach was able to reach the aim.

7 Conclusion and Future Work

The objective of this thesis work was to develop the banking android application for authentication and secure data transfer, which is to prevent ATM card skimming and intruder personating to cardless banking applications. In the literature review, we saw many technologies came up with solutions for secure data transmission and authentication. However, our approach can provide a feasible requirement for the current attacks. As the private key is saved in the legitimate customer's mobile device, it is difficult for the attacker to decrypt the information available on the server. Moreover, the QR code is not saved in the customer's mobile phone, instead, it is only allowed to scan. This prototype can be installed on smartphones. The application can upload and retrieve information from the server. The information could be accessed at any time, with an internet connection.

In the future, this work can be extended to improve the online banking websites and payment machines, where it requires authentication and secure transaction. The application that would contain abstract features which mentioned as aim, can be implemented with full features.

References

MD Shahabuddin, 2018. Reasons why cyber security is important for banks. Cyber Secur. Solut. Serv. - IT Secur. URL http://www.infoguardsecurity.com/reasons-why-cyber-security-is-important-for-banks/ (accessed 11.25.19).

What Should I Know About Encryption? | Surveillance Self-Defense [WWW Document], 2018. URL https://ssd.eff.org/en/node/36 (accessed 11.26.19).

information, N.U.N.U. is a former freelance contributor to L. who specializes in, VoIP, communication technology with a focus on, n.d. What is End-to-End Encryption? [WWW Document]. Lifewire. URL https://www.lifewire.com/what-is-end-to-end-encryption-4028873 (accessed 12.15.19).

The Latest Threats to ATM Security | SecurityWeek.Com [WWW Document], n.d. URL https://www.securityweek.com/latest-threats-atm-security (accessed 12.15.19).

Li, F., 2013. Why users adopt mobile banking service: An empirical study, in: 2013 10th International Conference on Service Systems and Service Management. Presented at the 2013 10th International Conference on Service Systems and Service Management, pp. 490–493. https://doi.org/10.1109/ICSSSM.2013.6602554

K., N., Janet, B., 2018. 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). Presented at the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), pp. 1–4. https://doi.org/10.1109/ICCSDET.2018.8821080

Nie, J., Hu, X., 2008. Mobile Banking Information Security and Protection Methods, in: 2008 International Conference on Computer Science and Software Engineering. Presented at the 2008 International Conference on Computer Science and Software Engineering, pp. 587–590. https://doi.org/10.1109/CSSE.2008.1422

Mallouli, F., Hellal, A., Sharief Saeed, N., Abdulraheem Alzahrani, F., 2019. 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). Presented at the 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. https://doi.org/10.1109/CSCloud/EdgeCom.2019.00022

Sudha, G., Ganesan, R., 2013. Secure transmission medical data for pervasive healthcare system using android, in: 2013 International Conference on Communication and Signal Processing. Presented at the 2013 International Conference on Communication and Signal Processing, pp. 433–436. https://doi.org/10.1109/iccsp.2013.6577090

Sharma, N., Bohra, B., 2017. Enhancing online banking authentication using hybrid cryptographic method, in: 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT). Presented at the 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT), pp. 1–8. https://doi.org/10.1109/CIACT.2017.7977275

Imran, Md.A., Mridha, M.F., Nur, Md.K., 2019. OTP Based Cardless Transction using ATM, in: 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST). Presented at the 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), pp. 511–516. https://doi.org/10.1109/ICREST.2019.8644248

Wahjuni, S., Pristian, R., 2016. Android-based token authentication for securing the online transaction system, in: 2016 International Conference on Information and Communication Technology Convergence (ICTC). Presented at the 2016 International Conference on Information and Communication Technology Convergence (ICTC), pp. 174–177. https://doi.org/10.1109/ICTC.2016.7763462

Kumar, D., Agrawal, A., Goyal, P., 2015. Efficiently improving the security of OTP, in: 2015 International Conference on Advances in Computer Engineering and Applications. Presented at the 2015 International Conference on Advances in Computer Engineering and Applications, pp. 912–915. https://doi.org/10.1109/ICACEA.2015.7164835

Adukkathayar, A., Krishnan, G.S., Chinchole, R., 2015. Secure multifactor authentication payment system using NFC, in: 2015 10th International Conference on Computer Science Education (ICCSE). Presented at the 2015 10th International Conference on Computer Science Education (ICCSE), pp. 349–354. https://doi.org/10.1109/ICCSE.2015.7250269

Bafandehkar, M., Yasin, S.M., Mahmod, R., Hanapi, Z.M., 2013. Comparison of ECC and RSA Algorithm in Resource Constrained Devices, in: 2013 International Conference on IT Convergence and Security (ICITCS). Presented at the 2013 International Conference on IT Convergence and Security (ICITCS), pp. 1–3. https://doi.org/10.1109/ICITCS.2013.6717816

Spongy Castle by rtyley [WWW Document], n.d. URL http://rtyley.github.io/spongycastle/ (accessed 12.15.19).

Gayoso Martinez, V., Sanchez Avila, C., Espinosa Garcia, J., Hernandez Encinas, L., 2005. Elliptic curve cryptography: Java implementation issues, in: Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology. Presented at the Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology, pp. 238–241. https://doi.org/10.1109/CCST.2005.1594866

Firebase [WWW Document], n.d. URL https://firebase.google.com/ (accessed 12.15.19).