COST EFFECTIVE USER-FRIENDLY ENHANCED AUTHENTICATION FOR ATM SYSTEMS

S. Thilakshan, H.P. Ganegoda, S. Vethushan, Dharmasena E.A.S.S. Kavinga Yapa Abeywardena, Amila Nuwan Senarathne, Madhushka Prasad Amarasingha

Department of Information Technology, Faculty of Computing, Sri Lanka Institute of Information Technology

thilak1994@gmail.com, heshie.pg@gmail.com, vethu24@gmail.com, sasekasethumdee@gmail.com, kavinga.y@sliit.lk, amila.s@sliit.lk, madushka.a@sliit.lk

***Abstract* — One of the most important and frequent problems faced by developing countries like Sri Lanka is Automated Teller Machine (ATM) frauds due to poor authentication system used by banks and financial institutions. This leads to ATM card fraudulent and ATM card misuse. This research paper states of a biometric-based multifactor authentication system through the user's smartphone for ATM systems which ensures the authenticity of ATM card ownership. In our proposed system we are considering the face recognition and fingerprint authentication as the multifactor authentication.**

Currently, these ATM cards are not having multi-factor authentication except One Time Passwords(OTP). Though OTP's are used, does not fulfill the security because since the OTP sent is using the SMS gateways which are not reliable all the time. Also, certain OTP systems that are time synchronized can also be compromised by phishing attacks via using the information gained from previous OTP codes through social engineering to predict what OTP codes will be produced in the future [1,2,3, 4].

In addition to that an ATM card also have several vulnerabilities like shoulder surfing, skimming, false ATM fronts, stealing the card, social engineering and pin spying [4]. The above-mentioned vulnerabilities emphasize that the ATM system does not have a proper layer of security. So it is a must to add a multifactor authentication to the ATM in order to avoid the risks based on the above-mentioned vulnerabilities. So to achieve that, a new system should be introduced with biometrical authentication which will mitigate the risks involved in the ATM.

This biometric-based multi-factor authentication system is not limited only within the boundaries of the ATM, but can also be extended and applied as a generic solution for many other similar domains where the multifactor authentication is vital and a must.

Keywords: Authentication, ATM Security, Multi-factor Authentication.

I. INTRODUCTION

Globally in the present scenario, a traditional ATM system accepts only on the Personal Identification Number (PIN) Code security system, enabling anyone with access to the PIN number rather than the owner to access the account easily. This ensures that the traditional Automated Teller Machine (ATM) systems are not fully secure as there is a drastic increase in fraud with the advent of modern technology.

ATM system is not designed to facilitate multi-factor authentication currently. RFID enabled ATM card is vulnerable to RFID skimming attacks. Though the OTP is implemented they are also not reliable since it has been done through the SMS gateway [4,5].

With the above mentioned issues, authentication for credit /debit cards has become less secure. Therefore, in order to resolve this problem, an extra layer of security should be added. This system should ideally provide maximum security possible even if the card is stolen, lost or even cloned. This system should also provide a low-cost solution as this would greatly encourage the adoption of such a system [5].

We are going to address these issues by providing a multifactor authentication to the ATM system. Initially, the user needs to get registered with our application After that whenever a user visits the ATM, our system will send a push notification to the user's mobile through our application then the user needs to give proper inputs to the application in order to perform the multifactor authentication. If the authentication is done successfully then the user can proceed with the ATM else the access will be denied for the ATM.

II. LITERATURE REVIEW

According to the previous researches, it was identified that there are no reliable and consistent authentication mechanisms for the ATM systems. Our objective is to add multifactor authentication to the ATM in order to achieve this we have divided into few components

A. Current authentication methods used widely with the ATM
B. ATM system vulnerabilities
C. Security analysis against vulnerabilities

A. Current Authentication methods used widely with the ATM

a. Authentication using Pin number

It is the formal and widely used authentication method for the ATM system which does not have a good level of security.

b. Authentication using OTP

OTP is sent through the mobile network so in case of a roaming user the particular user needs to trust

many networks so in such situation there is a higher possibility for the intruders to get in. It will also not ensure an additional security layer for the ATM [4].

B. *ATM system vulnerabilities*

By Shoulder Surfing: It is a way of finding the PIN number by standing just behind the customer and overlooks over the customer's shoulder to see the PIN number and memorize it [6].

Skimming: It is a method of collecting the PIN number and the card details without the knowledge of the customer [6].

False ATM fronts: Fraud gangs prepare a duplicate front panel of the ATM and install [7] their own software to that ATM. In such a situation the fraud team will be able to get all the information of the card [6].

Stealing the Card: Thieves watch a gentleman offering help to an elder or uneducated person. In such situations once the actual transaction is over the thief is going to replace the actual card with a duplicate card and hand over the duplicate card to that person (elder or uneducated). After that, the thief can perform the withdrawal with the actual card [6].

Social Engineering: "A person posing to be very responsible and making others believe that he could be trusted for some help in the form of fraud called 'Social Engineering' [6]." These fraudsters will make the calls to the customer imitating as the Bank Officer in order to collect the information of the ATM card holder.

Pin Spying: PIN spying devices are attached in the actual ATM in order to record the PIN number of the user that is being used [6].

Recently Sri Lankan bank ATMs' faced a huge problem by one of the above mentioned vulnerability (Skimming). So that the LankaPay has published the following note.

LankaClear has been educated by a couple of banks that specific fake exchanges have occurred utilizing ATM cards introducing some advances that ought to be taken by the overall population so as to guarantee the security of their financial exchanges such as making a special effort to be cautious when utilizing an ATM and check if any suspicious gadgets are associated with the ATM, seeing any suspicious looking individual wearing a top, protective cap or shades close to an ATM, quickly advise bank security or police and registering for SMS ready administration with your bank for every single electronic exchange [8].

C. *Security Analysis Against Vulnerabilities*

The above-mentioned vulnerabilities emphasize that the ATM system is not having a proper layer of security. So it is a must to add an additional security layer to the ATM in order to avoid the risks based on the above-mentioned vulnerabilities. So to achieve that a new solution should be introduced with biometrical authentication. So that we can mitigate the risks involved with the ATM [6].

III. METHODOLOGY

The proposed system will be carried out under 4 components.

A. Mobile App Implementation
B. Hardware Add – On Module
C. Control Panel
D. Authentication Server and REST API

A. ***Mobile App Implementation***

Users should ideally be able to accept/decline transactions. If users choose to accept a transaction, they would need to provide their biometric authentication details. Furthermore, users will be able to see the notifications regarding their transactions and they also have the ability to turn on/off the authentication system. These tasks will be performed via the mobile application.
Overcoming the challenges, we believe the rapid growth of technologies and globalization will further bring down the cost of mobile phones with these features in the near future. Therefore, our system will make use of a higher adaptation of mobile phones in the future. In response to duplicating identities, we are further strengthening our system by using the user timeout method and user lockout error method. If identity is not verified within a limited time, the session will be termed as timeout and the user will get a warning; and if the user makes multiple failed attempts at registering the fingerprint, face or both - then the user access to the ATM will be denied.

Each and every transaction is going to be authenticated according to the user's choice. In order to do that the user needs to get registered with the application then by using the credentials, the user needs to login to the application. Then whenever the user enters the withdrawal amount in the ATM then the ATM will generate a request to ATM switch following that ATM switch will send the push notification to the mobile through the application.
Once the user logs in to the application, he/she can leave the app without logging out since the session is used to carry the information.

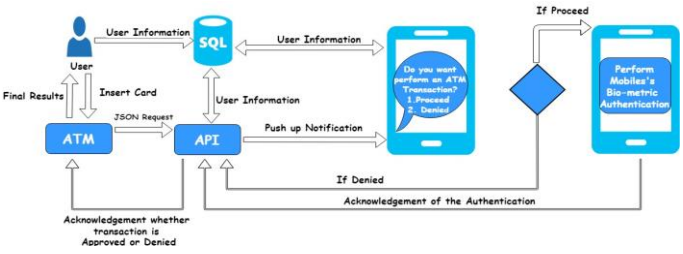

Figure 1: Technical Approach

The application consists of the following features

I. Profile Management
II. Push Notification
III. Performing Multifactor Biometric Authentication
   a. Fingerprint Authentication
   b. Face Recognition

I. Profile Management

In the application, the user is able to view the last fifteen transactions. In addition, the user is able to change the password, if the user forgets the password. A resetting mechanism is used to reset the password - once the user press "Forgot Password", then the application asks for the username and the entered username will be checked against the database. If the entered username is correct, then a "One Time Password" (OTP) is sent to the user's registered mail address. Once the user enters the OTP, then the received OTP is validated against the system. If the OTP is matching, the system automatically generates a random password which is then sent to the same registered mail address. By using the received password, the user will be able to login. After login, the user can use the password changing option to update their new password. Also, the user will have the option to alter their authenticating methods through the app. In order to update the authentication method, the login password is required. If the password is correct only, the authentication method will be updated accordingly.

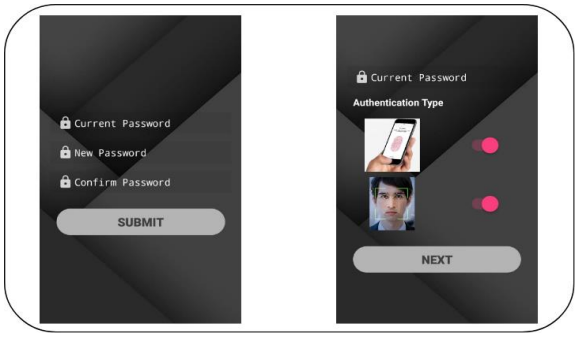

Figure 2: Customizing the Authentication Type

II. Push Notification

The push notification is sent using Firebase Cloud Messaging (FCM). Because FCM provides a reliable and battery-efficient connection between our server and devices that allow us to deliver and receive messages and notifications on Android at no cost. In order to use the FCM, the application should be registered with firebase [9].

Also, FCM uses the token to uniquely identify the end devices in order to send the push notifications. In our case, as soon as user logins to the application - the token will be generated and be stored in the database. Also, it will be deleted automatically when the user logs out from the application which takes the security to the next level.

III. Performing Multifactor Biometric Authentication

In order to authenticate each transaction, multifactor biometric authentications are used. Whenever a transaction happens in the ATM, ATM sends a request to ATM switch and after that, the ATM switch triggers and send the push notification to the application. By clicking that push notification, the transaction will be authenticated by the user.

a. Fingerprint Authentication

The user does not need to register his fingerprint separately for this application but the following conditions should be satisfied to perform the authentication.

- The device should run Android 6.0 [marshmallow] or higher.
- The device should have a fingerprint sensor.
- The user should grant the app permission to access the fingerprint sensor.
- The user should register at least one fingerprint on their device [10].
- The lock screen is secured with at least 1 type of lock.

If the user gets registered for the fingerprint authentication whenever a transaction happens, the push notification will be sent to the device and will require for the user's fingerprint authentication. In such a situation, the user is only allowed for a maximum of three attempts. If the user exceeds the limit, automatically that particular transaction will be denied.

By considering the Fingerprint Storage Security, most storage strategies on Android are insecure, especially when you consider the possibility of root access. But Google has made a noteworthy step in the right direction by moving all print data manipulation to the TEE and providing strict guidelines for fingerprint data storage that manufacturers must follow. TEE is a secure area of the smartphone's main processor. It guarantees the confidentiality and integrity of the code and data loaded inside. This separation enables security and protection from hacks, malware and root access [11].

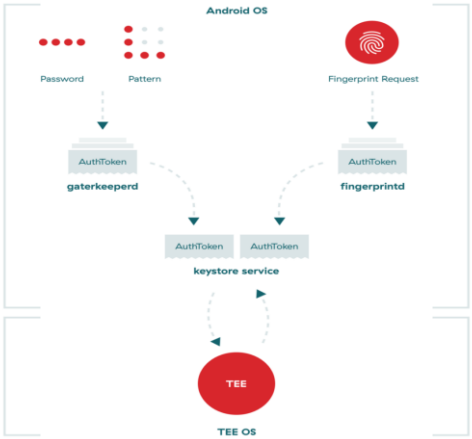

Figure 3: How securely the fingerprint information is stored in a device Source:https://stackoverflow.com/questions/41632225/android-where-and-how-securely-is-fingerprint-information-stored-in-a-device

As per the above figure explains

- “All fingerprint data manipulation is performed within TEE
- All fingerprint data must be secure within sensor hardware or trusted memory so that images of your fingerprint are inaccessible
- Fingerprint data can be stored on the file system only in encrypted form, regardless of whether the file system itself is encrypted or not
- Removal of the user must result in the removal of the user's existing fingerprint data
- Root access must not compromise fingerprint data [11]”.

Finally, fingerprint data is not backed up to the user's computer or Google's servers. It is not synced, shared or used by any other app on your device nor does it ever leave your device. This also means the user has to set up fingerprint authentication on each new device so that this particular authentication is used against each transaction even if the card is lost or stolen nobody can’t do anything with the ATM since it is asking for the owner’s fingerprint authentication. Also, Fingerprints are recognized in less than 600 milliseconds, providing fast and comfortable user experience [11] where the authenticating time also very few since in the ATM it is more sensitive for timeouts. Above mentioned are the definite advantages of using the fingerprint authentication for each transaction.

b. Face Recognition

It is a system that is built to identify the person’s face from the photos. While performing the registration, user needs to give his/her image to the system, and the photo will be stored with his/her username. While storing the photo, the photo is going to be stored in the internal storage so that, “It is always available, files saved here are accessible only by your app, when the user uninstalls your app, the system removes all your app's files from internal storage, internal storage is best when you want to be sure that neither the user nor other apps can access your files [12].”

As a result of this, the application makes sure that a photo taken through the application is not going to misuse anymore it is only being used to perform the authentication purposes.

Face recognition is built and located on the amazon server and Amazon Rekognition is used to build the face authentication server. Faces are coordinated depending on their visual geometry, including the relationship between the eyes, nose, temples, mouth, and other facial highlights [13]. There is an outline around the face, called a bounding box, which determines the only part of the image Rekognition considers in its analysis. The investigation at that point produces object documentation numbers for the picture that show the "area" for the real components of the face [12]. When customers are running a face search, the technology is comparing this data from the source image to each of the images it searches for. From there, the service assigns each face in the image a similarity score [13]. This approach ensures that Amazon Rekognition has no information about the identity of an individual, only the likelihood that one face is a potential match for another [13].

The above server is hosted in amazon ec2 instance (Ubuntu Server). Apache tomcat is installed in the server since we need to run the java jar file performing the face recognition and is embedded with tomcat.

Whenever a transaction happens, the push notification is sent to the device and will require the face authentication. In such a situation, the user is allowed to capture the image, and both images will be sent together to the face recognition server to identify the similarity. If the two faces are matched, then the user will be allowed to perform the withdrawal, or else the particular transaction will be denied. In this case also, the application provides maximum security for the ATM card even it is stolen or lost.

The following figure shows how the multifactor authentication which works in our systems.

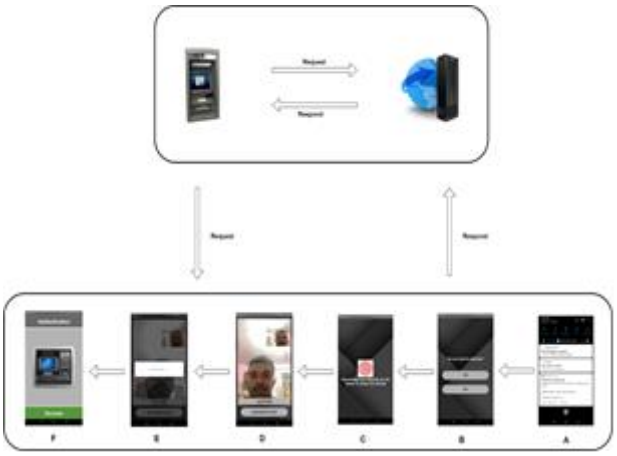

Figure 4: Authentication Process for each Transaction

*B. **Hardware Add-On Module***

Currently, every ATM consists of a computer but in this proposed system an ATM Hardware Module is introduced by using raspberry pi which is cost effective. This Hardware Module is built in a way that the can capable to handle the multifactor authentication request and the access will be decided based on the authentication results.

The Hardware Module component is designed to mimic the functionalities and the features of an actual ATM machine. Raspberry Pi has been used as the primary device to mimic the functionalities of an ATM machine. it has its own sequence of validating the transaction process using several biometric authentication methods. Furthermore, the Hardware Module starts the sequence of the process with the choice of language selection. At present, the current solution only supports Tamil, Sinhala and English as its language mediums.

After choosing the language, the Hardware Module prompts a message to the end user to enter the card number and the pin respectively. After successful validation of both card number and the pin, the Hardware Module displays a screen with the following options

1. Withdrawal
2. balance inquiry
3. PIN Reset

The end user is free to choose one of the given options to proceed with the process. If the end user chooses the withdrawal option, the Hardware Module undertakes the process and prompts a screen to enter the withdrawal amount. After inputting the amount using the on-screen keyboard, the Hardware Module processes the data and sends a private request to the server to validate and verify the identity of the user and to proceed with the transaction. In order to mimic the actual ATM systems, the initial request is sent to ATM switch then for each transaction certain parameters like Date, Time, Terminal ID, Location City, Currency Code, Approval Code, Business Code, Processing code, Card Number, Card Type, Account Number and Transaction Amount are being generated to make sure the Hardware Module is mimicking the actual ATM.

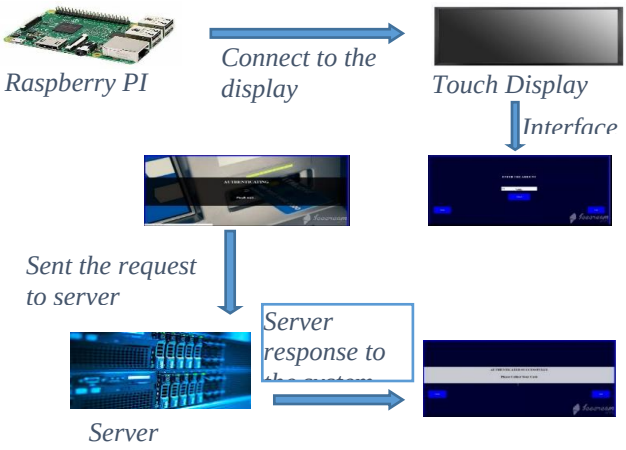

Figure 5: Technical Diagram of the Hardware Add-On Module

Mainly the Hardware Module consists of the following

- Raspberry Pi

An 8gb memory card is inserted into the pc then the memory card is formatted it using formatting software. After that the noobs file is downloaded and extracted the files into the memory card. Then the memory card is ejected from the pc and inserted it into the raspberry pi. After that, keyboard, mouse, monitor, Ethernet and power are get connected. Then I the boot loader is started and selected the raspbian full display version and then the language is selected and starts the installation process. Once installation was finished then reboot the system and start the Hardware Module.

C. *Control Panel*

The control panel maintains logs about Transactions and Errors. It helps the authorized user to view and manage the logs, collect information about actions made by the customers. The authorized user can view the logs. Logs are saves as read-only format. The control panel is secured from the vulnerabilities such as Broken authentic SQL Injection attacks, Cross-site scripting, Session Hijacking

The control panel consists of the followings.

Registration: This is the place where banks are registered to the system by providing their bank details. As a result of the registration, the banks will be offered with credentials (email, password). So that the bank can have the access to the control panel.

Login: In order to login to the control panel user should use valid credentials (email, password). Then the credentials will be validated and the access to the control panel will be decided accordingly. In addition to that forgot password mechanism also are implemented so that user can use that forgot password mechanism if needed.

Profile management: The details of the banks can be viewable and updated. If the needed password can be changed.

Transaction Log: This is the place where all the ATM transactions are being tracked. Those tracked transactions can be viewable and searchable.

Error Log: All the errors that are occurred during the ATM transactions will be stored. Those errors can be viewable and searchable

The control panel is going to secured from the following vulnerabilities.

Broken Authentication: Authentication functions related to the control panel is not implemented properly, it permits hackers to compromise passwords or session ID's or to exploit other operation faults using user's credentials [14].

SQL Injection Attacks: "A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete) and execute administration operations on the database" [15].

Cross-site scripting: "Which malicious scripts are injected into trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user" [16].

Session Hijacking: Secretly the attacker passes generated session ID to the user and once a user login to the system his/her login data will be stored as session data [17].

The above vulnerabilities are prevented in control panel by using OWASP guidelines. Broken authentication is mitigated by using Password length, Password complexity, Protection against brute force [14].SQL Injection attacks are mitigated by using prepared statements and parameterized queries [15]. Cross-site scripting is mitigated by using Parameterized Statements, Object Relational Mapping, Sanitizing Inputs, Escaping Inputs [16]. Session Hijacking is mitigated by recreating a new session ID when the user logs in to the system.

D. *Authentication Server and Representational State Transfer (REST) Application Programming Interface (API)*

Enormous data varying around wide types are handles by APIs. Correspondingly the main involvement of any data provided is how to secure this data specifically. The main target is to provide an extra layer of security based on authentication technologies, maximization of the security to the ATM. The REST API will allow secure communication with the banks and the authentication system. The concept that data should be private, stable, and that it should be available for manipulation is key to any conversation on API data management and handling.
A web application is a very confidential customer running on a web server. Asset proprietors get to the customer through an HTML UI rendered in a client specialist on the gadget utilized by the asset proprietor. The customer qualifications just as any entrance token issued to the customer are put away on the web server and are not presented to or available by the asset proprietor.
By providing an extra layer of security based on authentication technologies, maximization of the security to the ATM. Once the authentication system is completed there is no need for change in existing implementations of the currently used ATMs. The system can be merged externally without any cost. For this purpose, a Rest API is implemented using OAuth 2.0 which allows secure communication with banks and authentication system.

a. The reason to secure an API Key

API keys are commonly not reviewed secure; they are ordinarily available to customers, making it simple for somebody to take an API key. When the key is stolen, it has no termination, so it might be utilized inconclusively, except if the task proprietor repudiates or recovers the key. While the limitations that can set be on an API Key diminishes, there are better methodologies for authorization.
Ensuring the API does not need to be troublesome and as OAuth 2.0 is the initial move towards a progressively secure API [18].

b. Use of OAuth 2.0 in securing an API Key

OAuth 2.0 is a security protocol used to protect a web API. It's utilized to interface sites to one another and it powers local and mobile applications associating with cloud administrations. OAuth is by a wide margin the predominant security strategy on the web today, and its omnipresence has evened the odds for engineers needing to verify their applications [19].

c. Authorization Process within OAuth 2.0

In OAuth, the Web Authorization Protocol, the end client designates some piece of their position to get to the ensured asset to the client application to follow up for their sake. So as to get that going, OAuth brings another part into the framework: the authorization server:
The authorization server (AS) is trusted by the ensured asset to issue exceptional reason security accreditations – called OAuth access tokens – to customers. So as to get this token, the customer initially sends the asset proprietor to the authorization server so as to ask for that the asset proprietor approve this client. The client verifies to the authorization server and is for the most part given a decision of regardless of whether to approve the customer making the demand. The customer can request a subset of usefulness, or extensions, which the client might almost certainly further decrease. When the authorization permission has been made, the customer would then be able to ask for an entrance token from the authorization server. This entrance token can be utilized at the ensured asset to get to the API, as conceded by the asset proprietor.

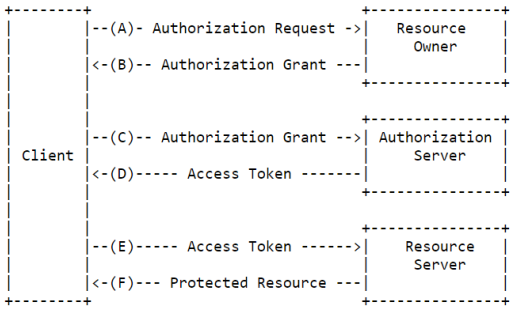

Figure 6: Theoretical OAuth 2.0 stream

The theoretical OAuth 2.0 stream represented in the above figure 6 portrays the collaboration between the four mantles and incorporates the accompanying advances:

(A): - The client demands authorization from the resource proprietor. The authorization demand can be made specifically to the asset proprietor (as appeared), or ideally in a roundabout way through the approval server as a go-between.

(B): - The client gets an authorization concede, which is an accreditation speaking to the resource proprietor's approval, communicated utilizing one of four concede types characterized in this detail or utilizing an augmentation allow type. The authorization concede type relies upon the technique utilized by the customer to ask for approval and the sorts upheld by the authorization server.

(C): - The client asks for an entrance token by verifying with the authorization server and exhibiting the authorization grant.

(D): - The authorization server verifies the client and approves the authorization grant, and if legitimate, issues an entrance token.

(E): - The client asks for the shielded asset from the resource server and confirms by displaying the access token.

(F): - The resource server authorizes the acess token, and if substantial, serves the demand [20].

IV. RESULTS & DISCUSSIONS

The primary outcome of this research project is an efficient and cost-effective solution for bank and finance institutions to reduce the risks associated with ATM transactions.
Our proposed solution of this research project is aimed to implement cost-effective user-friendly enhanced Authentication system for the bank and financial

institutions. The proposed system is a mobile-based biometric authentication scheme for ATM transactions which guarantees the authenticity of card ownership information and non-repudiation of bank transactions. Our main objective is to provide an extra layer of security by using mobile-based biometric authentication.

As for the Mobile Application Component, the users' mobile's biometric authentication is taken to verify the transactions. Here the user does not need to give his biometric details separately instead of the enrolled biometric authentication for the users' mobile will be used.

As for the Rest API Component, the accurate communication between the bank and the authentication system will be maintained. Here the request sent by the ATM should be handled.

As for the Control Panel Component, the user's will be able to view and search using parameters, their transaction log files and error logs. Banks' Profile management will be also handled. In order for further security feature, proper mechanisms are used to secure the control panel.

As for the Hardware Module Component, it is going to mimic the real ATM functionalities in addition to that it is going to facilitate to handle multi-factor authentication request.

V. CONCLUSION & FUTURE DIMENSIONS

"Cost Effective User-Friendly Enhanced Authentication For ATM System" has been implemented as planned, with some slight changes that suited the current technical specifications and time constraints.

The proposed solution is aimed to implement a biometric-based multi-factor authentication for ATM transactions, which adds an additional layer of security to the ATM transaction even if the card is lost or stolen.

The following are some of the major anticipated benefits,

- Our proposed solution main objective is to add an additional layer of security to the to the ATM systems without changing the backend of the ATM.
- With our system, we can reduce most of the risks that are available in the present against the ATM transactions
- The user doesn't need to register his authentication (biometric) details separately.
- Users can increase their security for the ATM without spending much money.
- Easy access to the main control system via API.
- Ability to be integrated with any bank via common ATM switch.

For the future works, instead of creating and using a separated face recognition server we can use the inbuilt face authentication mechanism to authenticate each ATM's transaction at the moment the android does not give the access to take the inbuilt face recognition authentication. Also, like the above research, particular research can be applied to POS transaction as well. The other new Biometric schemes such as Iris pigmentation, and behavioral characteristics of a person, when combined with growing field of Artificial intelligence, They provide a very advanced authentication technology[22]. We believe that our Cost Effective User-Friendly Enhanced Authentication For ATM System for ATM's will help the banking industry to avoid the frauds against the ATM's transaction.

ACKNOWLEDGEMENT

First, we would like to thank Sri Lanka Institute of Information Technology, for providing the opportunity and platform to develop and expose our skills and abilities via performing a research project and for all collective arrangements done to make this project successful. This project would not have been possible without the guidance of our research supervisor Mr Kavinga Yapa Abeywardena and co-supervisor Mr Amila Nuwan Senarathna. We would like to sincerely thank them for their valuable insights and supported us to make this a success. Their guidance and concern provided throughout the research of our project are thoroughly appreciated. Our collective thinking, effort and collaboration are what made our project a success. Also, we would like to thank Mr Madushka Prasad Amarasingha (Research Assistant - SLIIT) for guiding us throughout the project.

REFERENCES

[1] Indrajani Y. Heryadi L. A. Wulandhari and B. S. Abbas, Recognizing Debit Card Fraud Transaction Using CHAID and K-Nearest Neighbor: Indonesian Bank Case, Bina Nusantara University Jakarta, Indonesia, 2016

[2] W. B. Hsieh and J. S. Leu, Design of a Time and Location Based One-Time Password Authentication Scheme, Taiwan, 2011

[3] William Morrison (2014), The Fundamental Problem With OTPs in Two-Factor Authentication[online]. Available: https://www.logintc.com/blog/2014-01-14-one-time-passwords.html [Accessed: 11- Sep- 2018].

[4] Geoauth.info, 2018. [Online]. Available: http://geoauth.info/. [Accessed: 11- Sep- 2018].

[5] S. Acharya A. Polawar P. Y. Pawar, Two Factor Authentication Using Smartphone Generated One Time Password, Pune: India, 2013

[6] Analysis of Vulnerabilities in ATM Transactions, chapter 3 [online]. Available:http://shodhganga.inflibnet.ac.in/bitstream/10603/25038/7/chapter-3.pdf [Accessed: 19th of September 2018]

[7]"Lanka Clear", Lankaclear.com. [Online]. Available: http://www.lankaclear.com. [Accessed: 19- Sep- 2018].

[8]"LankaPay", M.facebook.com, 2019. [Online]. Available: https://m.facebook.com/story.php?story_fbid=2089847187768904&id=534678303285808. [Accessed: 01- Mar- 2019]

[9]"Notifying your users with FCM", Android Developers Blog, 2019. [Online]. Available: https://android-developers.googleblog.com/2018/09/notifying-your-users-with-fcm.html. [Accessed: 12- Feb- 2019].

[10]"Biometric authentication in Android", ProAndroidDev, 2019. [Online]. Available: https://proandroiddev.com/5-steps-toimplement-

biometric-authentication-in-android-dbeb825aeee8. [Accessed: 13- Feb- 2019].
[11]"Android Fingerprint Security", 2019. [Online]. Available: https://infinum.co/the-capsizedeight/android-fingerprint-security. [Accessed: 13- Feb- 2019].
[12]"Save files on device storage | Android Developers", Android Developers, 2019. [Online]. Available: https://developer.android.com/training/data-storage/files. [Accessed: 13- Feb- 2019].
[13]"The Facts on Facial Recognition with Artificial Intelligence", Amazon Web Services, Inc., 2019. [Online]. Available: https://aws.amazon.com/rekognition/the-facts-on-facial-recognition-with-artificial-intelligence. [Accessed: 13- Feb- 2019].
[14]"Broken Authentication and Session Management - OWASP", Owasp.org, 2019. [Online]. Available: https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management. [Accessed: 27- Feb- 2019].
[15]"OWASP/CheatSheetSeries", GitHub, 2019. [Online]. Available: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.md. [Accessed: 27- Feb- 2019].
[16]"OWASP/CheatSheetSeries", GitHub, 2019. [Online]. Available: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.md. [Accessed: 27- Feb- 2019].
[17]K. Teja, "Preventing Session Hijacking in PHP", Packetcode.com, 2019. [Online]. Available: http://packetcode.com/article/preventing-session-hijacking-in-php. [Accessed: 01- Mar- 2019].
[18]O. communication, "Oauth2 vs APIKey in a server to server communication", Information Security Stack Exchange, 2019. [Online]. Available: https://security.stackexchange.com/questions/179236/oauth2-vs-apikey-in-a-server-to-server-communication. [Accessed: 01- Mar- 2019].
[19]"API Keys versus OAuth - How to secure your APIs?", API Friends, 2019. [Online]. Available: https://apifriends.com/api-security/api-keys-oauth/. [Accessed: 01- Mar- 2019].
[20]"RFC 6749 - The OAuth 2.0 Authorization Framework", Tools.ietf.org, 2019. [Online]. Available: https://tools.ietf.org/html/rfc6749#section-4.4. [Accessed: 01- Mar- 2019].
[21]dzone.com. (2019). OAuth 2: Why You Should Care - DZone Web Dev. [online] Available at: https://dzone.com/articles/what-is-oauth-2-and-why-should-you-care [Accessed 1 Mar. 2019].
[22]"Face Recognition and Fingerprint Based New Generation ATM", vol. 3, no. 3, 2018. Available: https://ijisrt.com/wp-content/uploads/2018/04/Face-Recognition-and-Fingerprint-Based-New-Generation-ATM.pdf. [Accessed 1 March 2019].