A Multifactor Security Protocol for Wireless Payment-Secure Web Authentication using Mobile Devices

A Project Report

Submitted in the partial fulfilment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

170030193 - Challa Venkata Pranith

170030614 - Kolli Sai Kiran

170031068 - Pulivarthi Goutham

170031340 - Valiveti Lohya Sujith

Under The Supervision of

DR.K V D KIRAN

(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L E F Deemed to be University

Green Fields, Vaddeswaram, Guntur District – 522 502

<March, 2021>

KONERU LAKSHMAIAH EDUCATIONAL FOUNDATION

DEPT OF COMPUTER SCIENCE AND ENGINEERING

(DST-FIST Sponsored Department)

DECLARATION

The Project Report entitled "A Multi-factor Security Protocol For Wireless Payment-Secure Web Authentication Using Mobile Devices" is a record of bonafide work of (170030193, 170030614, 170031068, 170031340) submitted in partial fulfilment for the award of B.Tech in Computer Science And Engineering to the K L University. The results embodied in this report have not been copied from any other Departments/University/Institute.

Signature of the Students

170030193-Challa Venkata Pranith

170030614-Kolii Sai Kiran

170031068-Pulivarthi Goutham

170031340-Valiveti Lohya Sujith

2

KONERU LAKSHMAIAH EDUCATIONAL FOUNDATION

DEPT OF COMPUTER SCIENCE AND ENGINEERING

(DST-FIST Sponsored Department)

CERTIFICATE

This is to certify that the Project Report entitled "A Multi-factor Security Protocol For Wireless Payment-Secure Web Authentication Using Mobile Devices" is being submitted by 170030193, 170030614, 170031068, 170031340 submitted in partial fulfilment for the award of B.Tech in Computer Science And Engineering to the K L University is a record of Bonafide work carried out under our guidance and supervision. The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of the HOD   **Signature of the Supervisor**

 Dr. V. Hari Kiran   **Dr. K .V .D KIRAN**

Signature of the External Examiner

ACKNOWLEDGEMENT

ABSTRACT

The way we live has been reformed by SmartPhones. Cell phones and PDAs have been packed with ubiquity to a great extent and consumers have now started webbased banking, webbased item purchasing and other online administrations. Either the site or the remote flexible channel has been used autonomously by previous web access authentication systems to confirm the personality of faroff clients. To get to the latest online administrations reliably needs a username and secret phrase to verify the client personality.

This is a major weakness because the hidden word can be compromised and later used by the man in the center attack to make illegal entry to the record of the client. We are likely to make a validation system based on a multi-faceted verification method that is both reliable and deeply accessible. It features a proprietary way to deal with rendering a validation system based on spasms (Exchange Recognizable proof code) and SMS (Short Message Administration) in order to enable the conventional Login/secret phrase framework to get an additional security level. In addition, we use an encryption method which depends on a symmetric key and an idea of an iterated block figure. This idea has been used to retain Spasms on PDAs/PDAs as a mystery code but is also used to start secure web transfers using mobile phones/PDAs. Finally, we extend the two-way validation framework that confirms the two players (client and e-specialist organization). A definite threat analysis shows that the proposed system is safe from various types of web attacks such as phishing, man-in-the-middle, infections and anything similar as outlined above. In this paper we are proposing a new system and a protocol for Secure way of transactions by using Multifactor Authentication System.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ABBR	ABBREVATION
MFA	MULTI FACTOR AUTHENTICATION
ACK	ACKNOWLEDGEMENT
AES	ADVANCED ENCRYPTION ALGORITHMS
CA	CUSTOMER AGENT
CB	CUSTOMER BANK
CBAS	CUSTOMER BANK AUTHENTICATION SERVER
CLDC	CONNECTED LIMITED DEVICE CONFIGURATION
OTP	ONE TIME PASSWORD
PKI	A PUBLIC KEY INFRASTRUCTURE
SMS	SHORT MESSAGE SERVICE
TIC	TRANSACTION IDENTIFICATION CODES

CHAPTER 1: INTRODUCTION

As registering become imminent, individuals are progressively relying on the Internet to use their enterprise over the Internet. The Web is generally a preferable choice for online services such as internet enterprise, e-casting a ballot, e-banking e-government, and soon something close to that... Online applications need a defensive tackle highlight to ensure confidential customer information. Protection is an important factor in the online payment system focused on the web. There are numerous network vulnerabilities affecting web protection arrangements and growing electronic sharing hazards. The majority of the verification system relies on passwords, individual identifying evidence numbers and keys to navigate their own record data. This sort of verification framework doesn't really verify or validate the character of the customers that the person in question claims to consistently meet the current electronic administrations requires a username and secret key are done to verify the character of the client. This is a big flaw as the hidden word can even be hacked by the man in the center attack later used to make illegal admission to the client's record.

Single factor authentication expands chances presented by phishing, recognize robbery, online fraud and loss of secret client data. In this way, monetary foundations should actualize a powerful validation to decrease extortion and increment security for applications. Solid client verification is important to authorize security and help monetary organizations to identify and diminish client character robberies

With regard to communications, classified information protection has consistently been an important issue. With equipment advancements that give customers the advantage of transparency used in mobile phones, individuals are currently investing more and more energy in these gadgets. In addition, with the viral popularity of web-based media applications and single sign-on, customers generally do not escape varying potential risks with their data. Multifactor Authentication makes more and shifted dividers to shut out some unacceptable individuals from seeing your data

What is Single-factor Authentication (SFA):

User Enters Username & Password   **User is Authenticated**
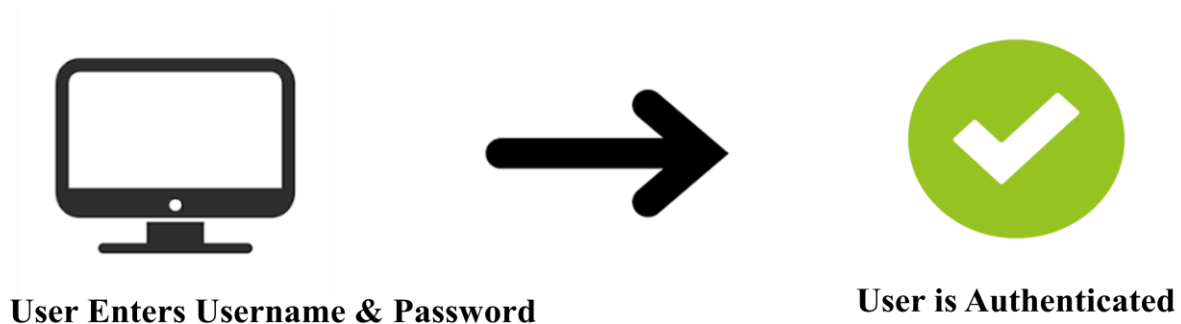
Figure 1: Single-factor Authentication

The most straightforward sort of confirmation strategy is single-factor authentication. A individual matches one credential with the SFA to verify himself or herself online. A password to a username will be the best known example of this. Today, most verification utilizes this kind of verification methodology.

Problems with Single Factor Authentication:

- Vulnerability problems
- The lacking of a backup stronger authentication
- The login credentials need to be strong enough
- We should not login in multiple devices as we cannot secure them.

What is Two-factor Authentication (2FA):

User Enters Mobile Number   User Receives One Time Password   User Enters OTP & Authentication is done

Figure 2: Two-factor Authentication

Two-factor authentication uses the same combination of password/username, but with the addition of being asked to verify who a person is, such as a mobile device, using only something he or she owns. Getting straight: utilizing 2 factors to validate an identity

Problems with Two Factor Authentication:

- Phishing Attempts
- Social Engineering & Brute-Force Attacks
- Broken Logic
- Key Logging

What is Multi-factor Authentication (MFA):

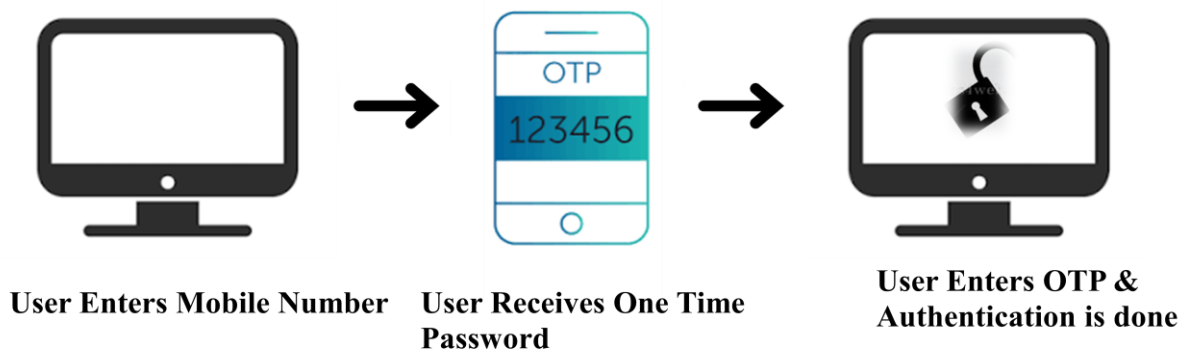A combination of the inclusion procedures is used for Multi-factor Authentication: something you know, something you have and something you are. The 2FA is an MFA subset.
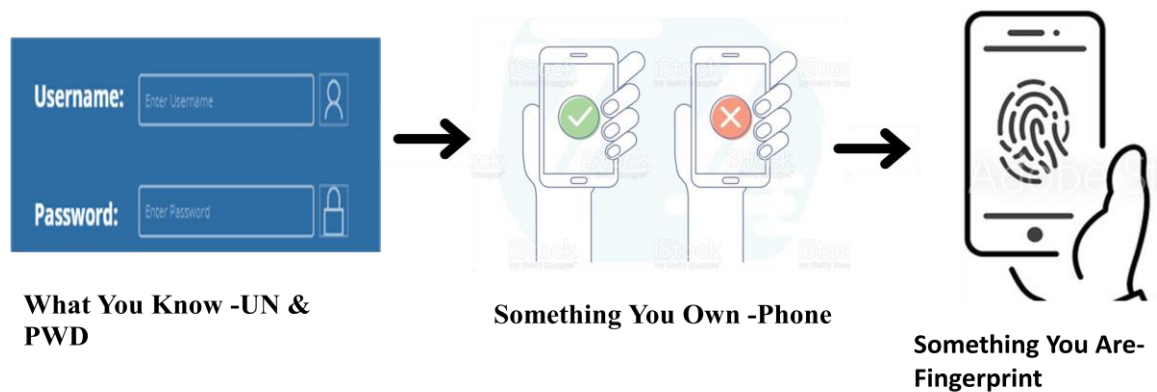

What You Know -UN & PWD

Something You Own -Phone

Something You Are- Fingerprint

Figure 3: Multi-factor Authentication1

Factors of Authentication:

Authentication methods those are supported quite one factor are harder to break and secure than single-factor methods. Efficiently designed and implemented multi Factor authentication methods are more reliable and Secure to scale back online frauds. Selection of proper authentication key plays the key role in securing online transactions. A password may be a sort of secret authentication data which grants access to only authorized users .The password is understood to the sole authorized users and unauthorized persons are unaware of this, and user wish to realize access must be tasted to verify the authenticity of the user by checking the The passwords are universally wont to keep authorization. However, password systems are susceptible many attacks and attackers can recover the user password by some attack. Additional protections are implemented to guard the info channel to form password confidential, but this if often not completely secure against attacks .Many security experts now realize that password aren't secure mechanism to guard user confidential data; passwords are often hacked easily by hacking attacks.

A Physical device that is used for authentication purpose is known as the hardware token. It's a hardware implementation of the authentication device attached to a licensed user's computer. Hardware token maybe a specialized physical device that protects secrets (cryptographic keys) and performs cryptographic operations .The cryptographic operations are wont to secure the communication channel and authentication of parties . There are also few drawbacks drawbacks with hardware tokens ,It Increases the implementation cost, and sophisticated functionality required in implementation and deployment And it is easy use for customers. Software tokens are almost like hardware tokens .It's software implementations of hardware

11

tokens. Software tokens run on the PC or on a separate multi-purpose device but hardware tokens are stored on an external device faraway from the PC . Software tokens support authentication of both parties and protect the used channel to transmit data for authentication. Some of the issues with software tokens are they can be copied easily without acknowledging user. The one-time password (OTP) based system is safer then ordinary password based system .It's difficult break and secured from unauthorized users .During this sort of system passwords get updated constantly, for every access user has new password so it reduced the risk. Special algorithms are wont to generate a series of password .Each password of the succession is named a one-timed password because it is different from the opposite password and each password is employed just one occasion .Many sorts of one time password generation system are available to protection against attacks.

In biometric identification human physical and behavioural characteristics are wont to verify the identity of the user. Biometrics is compatible to local access control instead of remotely access authentication .just in case of remote access supported biometric authentication user's personal data are used for authentication, significant privacy issues arise with the gathering storage and use of such information. Some special measures should be taken in the situation of remote authentication since client's very own information goes on correspondence network for verification. The financial agencies consider that single-factor authentication isn't appropriate for financial transactions or access to customer information or online transfer of funds.

Figure 4: Multi-factor Authentication2

Introduction To Servicenow Platform:

ServiceNow provides features like flexibility and power toattain the main objective of project management. In ServiceNow, a user can select an interface which he likes and it provides detailed information about the interface the user is working on. ServiceNow is cloud infrastructure that simplifies solutions for corporate automation, providing better interactions for consumers and employees. It is a solution which uses an advanced service management approach to move the company to the cloud even faster. The program provides an on-demand service ecosystem that optimizes the usage of the cloud while reducing infrastructure costs and standardizing service requests, permissions, and cloud resource configuring. It stimulates the distribution of cloud services through a centralized, self-service user interface that uses a single method of action to further streamline cloud operations. Users can make management choices based on their provided business resources through the use of ServiceNow.

ServiceNow's benefits improve outstanding service and establish an on-demand service ecosystem that optimizes the usage of the cloud, lowers resource costs, automates service demands, and delivers cloud services. It offers server management for VMware, Amazon Cloud, and Microsoft Azure environments, making it versatile for customers of ServiceNow. The consistency of the results, portability, monitoring, its constant cost model, training and management, and much more are some of the benefits of using ServiceNow. For facilities executives working in IT, it is also a comprehensive facility management approach. It provides ServiceNow consumers with a reliable approach that allows executives to tackle challenging organizational priorities for company stability, efficiency, and employee productivity. The best decision would be if you are on a watch to keep in touch with the clients on the ServiceNow client list

Almost every small scale and large scale companies use ServiceNow. Around 51,000 users are on Now learning platform. Using Servicenow we can develop an application in less amount of time. Servicenow instance page consists of three frames:

   i)   Banner Frame: This frame has options like connect, magnifying symbol, help (to search document), gear icon (settings option).

Scope in Servicenow - The scope helps the administrators to preserve their applications by limiting entry to few application files. The default scope in Servicenow is "Global". Scope provides security and authentication.

   ii)   Application Navigator Frame: This frame consists of filter navigator, application menus and modules. In Servicenow, every table that we create is stored as modules and everything that we create in Servicenow is a "Table"

   iii)   Content Frame: This frame consists of home page dashboards, forms, lists, knowledge base and service catalogues.

Servicenow Studio

Studio is the platform or center place where we create applications and different artifacts like creation of the table, client side and server side scripting. Applications developed in studio are called as "Scoped Applications".

Servicenow Instances

ServiceNow uses a single tenant model, which ensures that every other client in the cloud receives one's own replica or several versions of ServiceNow. These versions are identified as instances of ServiceNow.

Types of instances

i) Developer instance – This instance is used only for practice and available for everyone (one particular user only). This instance will go to hibernation if it is not active for few days (Eg: dev1234.service-now.com).

ii) Production instance – Companies will purchase these instances from ServiceNow (accessible for all employees). One company has one production instance (Eg: xyzcompany.service-now.cometc). This instance will never be expired i.e., cannot hibernate.

iii) Testing instance – This is the instance which is used to test the applications developed in ServiceNow platform.

Notification

Notification in servicenow it provides specific types of services like EMAIL, PUSH, SMS(Direct Message) for the management of team members and contractors who are working as the group of members for the development of any project/ Application. it can be used to inform any updates or any changes that are required to include or indulge in any specific section that they are currently working on. We can send a notification in 2 ways that is 1) inserted or Updated 2)Event Based

i) The First Way which is Inserted or Updated will be working whenever a record is inserted or updated in the Form

ii) The Second way which is Event based will be triggered whenever a record is inserted and the event which is Linked to The table that we have created.

CHAPTER 2 : LITERATURE SURVEY

In 2016, Ashok Nath[1] faced problems and difficulties in the two-factor authentication method as the "store-front" looks genuine, and the user ends up entering directly into the hands of an identify information such as credit card numbers, social security numbers and bank account information. Another type of security violation is if the would-be hacker already has access to the device itself. Then the intruder tries to piggyback the transmission and either execute fraudulent transactions or access protected transactions when the user accesses either company or internet services.

Kumar Abhishek and Prabhat Kumar[2] published a study on multifactor authentication systems in 2017 as new Authentication challenges and opportunities ensure that a customer is who they claim to be. As the authentication process includes more considerations, the appreciation of authenticity increases exponentially.

In 2017, Rajeev Ranjan[1] given the preferred use and interpretation of multifactor authentication that current systems can use. It is referred to as Strong Authentication or Multifactor Authentication when the protection infrastructure uses two or more separate and different forms of authentication mechanisms to protect the well-being of legitimate Multifactor authentication uses combinations of anything to provide more remote authentication than usual, weak single-factor username and password '

In 2018, Aleksander Ometov, Sergey Bezzateev, Sergey Andreev[3] proposed the challenges faced by MFA management as the identification of users is a key element for a cautious and systematic approach to the adoption and implementation of MFA solutions, where most problems arise from opportunities and potential problems.

The Community of the Servicenow which was established a long way ago has got solutions and discussions of the various new ideas and the implementations on the servicenow platform. The way of generating the one time password for the secure transaction was developed by using separate sections in the studio of servicenow.

CHAPTER 3: THEORETICAL ANALYSIS

3.1 Existing System:

Mobile Payment (M-payment) is a wireless payment system that has drawn researchers' interest in recent years. M-Commerce payments can be made in many forms and are based on the paradigm of anywhere, anytime." In M-commerce or E-commerce applications, m-payment is a critical component. M-payment on a mobile device will be very common and provide immense potential for M-commerce in the coming years, according to the worldwide forum. In recent years, the Safe Wireless payment system has become an important field of study. For Internet-based commerce, there are different types of electronic payment solutions available. Mobile payment means M-wireless-based commerce's electronic payment to facilitate point-of-sale/point-of-service (POS) payment transactions using mobile devices such as mobile phones, tablets and personal digital assistants (PDAs) or mobile terminals.

In general, in order to process and sustain m-payment transactions based on wireless commerce applications, M-payment methods are normally used by vendors to make wireless payments, content providers, and information and service providers. The current m-payment schemes can be categorized into three key groups, as discussed in

- Payment mechanisms focused on accounts,

- Payment system for mobile POS,

-E-cash or E-wallets.

Therefore in order to improve more and more reliable protection for online payments, we have proposed a new multi-factor authentication protocol.

The first type is understood as account-based payment systems, during which each customer features a valid account maintained by a Trusted Third Party. The user can initiate pre-paid or postpaid financial transaction. There are three subdivisions of account-based M-payment systems:

1. The payment system supported mobile, which facilitates the customer to try and do commerce and make payment over mobile phones.

2. Credit-Card M-payment systems, the users can make payment via of credit cards using their mobile devices and

3. The open-end credit Payment systems.

The second sort of m-payment system refers to the mobile POS payment systems by which customers can buy products using vending machines or online website with their mobile devices. There are two sorts of POS payment systems.

1. An automatic POS payments, which are frequently utilized in immobile terminals such as vending machines, parking meters, etc;
2. Coming to POS payments/installments, during which versatile clients can make installments with the help from an assistance party like a cab driver, a counter representative, and so on…

Coming to the last part, it's the most used one in this generation, i.e E-wallets or E-cash . During this method customers stores digital take advantage their E-wallet from a open-end credit, mastercard or virtual check. Digital cash is like electronic take advantage virtual bank account where the user can make payment for his or her purchases. E-wallets are frequently utilized in micro payments or small payments .Customers can obtain E-Wallets from their financial organisation .Specific merchants or technology providers to form payments.

Digital cash:

Digital cash or E-cash which may be a method of money transfer, this method of the payment is electronic transfer of fund using computers via internet or on mobile phones via GPRS connection to use internet or Bluetooth. This method of cash transfer is straightforward and versatile to the users. Digital cash is additionally frequently used in day to day life to form small payments. Digital cash system has many advantages which makes customers life easy, it also has some disadvantages which include fraud, device failure, tracking of particular person and reduction in human social interactions. consistent with the facts , fraud over digital cash has been a pressing issue in recent years. It includes illegal access of customers checking account and unauthorized access of user private banking information which has increased the knowledge hacking and fraud.

Electronic Wallet:

Electronic purse may be a system which facilitates customers to not carry cash with them for small payments. It is supported open-end credit technology which is analogous to pre-paid financial card. Some advantages and drawbacks and example of existing system are given below for more detail on Electronic purse please refers.

Advantages:

1. Easy to use.

2. Capable of holding wide range of currencies, easy payment based on P2P.

Disadvantages:

1. Used for small transactions

2. There is no globally recognized standard for this kind of payment structure.

Distributed Wallet:

A distributed wallet is a payments application which starts payment transaction using pre-installed software on the consumer's computer or mobile which connects via Internet to the wallet servers to make transactions. The client machine first establishes connection with the server machine using a connecting protocol for the user authentication and to make transaction. Few advantages and disadvantages and example of present using system are given below

Advantages:

1. It is compatible with Smart Card and Secret PIN's.

2. To make payment via distributed wallet service there is no need for merchant subscription.

Disadvantages:

1. The difficulty of Distributed wallets is greater than remote wallets and it also require large messaging sequence to run complex environment.

2. It activates client side module of the distributed wallet results in multiple wallet servers.

Personal Wallet:

A personal wallet is a program or hardware mounted on a user computer. There is no need of a server, since a payment transaction does not require a wallet server. User's name, Credit information is stored locally on the user's computer. Some of the benefits and The drawbacks and the example of the current system are given below.

Advantages:

1. Implementation costs are minimal.
2. Transaction can be done offline.

Disadvantages:

1. Need a huge storage space on user computer.

3.2 System Specification and Architecture:

In the earlier section, various types of current electronic payments were considered, where we also explored their benefits, drawbacks and scope of applicability. We develop the basic ideas of the present work in the present chapter. The methods of payment and the function of multifactor authentication are discussed first. This is preceded by an overview of the protocol for client and transaction authentication developed by us. This is then expanded to create a two-way authentication protocol.

Mode of payment:

Electronic payment systems are divided into three categories, as shown in

1. Credit card based systems

2. Debit card based systems

3. Electronic checks and Account Transfers

Credit card based systems: Credit cards are a pre-agreement payment scheme, credit card issuers and purchasers are members of a card group, e.g. Your Visa or MasterCard. Issuers are the bank authority. Merchants accept credit cards for payment they need the required facilities, they are affiliated with acquisitions that have credit card payment equipment. Credit card purchases may be made both while they are physically present or over the phone or the Internet.

Debit card based systems: Debit cards are paid against an account with a pre-agreed deposit scheme. It is similar to a card with an ATM. Members of a card association are both the issuers of debit cards and the acquirers, e.g. SBI ATM Card, ATM Card for City Bank, etc. Both of them are normally banks. Merchants are partnered with the acquirer, who provides the retailer with the requisite infrastructure to accept debit card payments. Debit card payments may be made either physically or over the telephone or the Internet.

Electronic Transfer systems: Electronic payment depends on the Internet to make money transfer on the basis of a centralized account model Usually, an agreement between the payer and the financial institutions of the merchant is necessary to make money transfer, or both parties have an account in the same financial institution.

Multifactor Authentication:

For high-hazard exchanges that expect admittance to customer data or the exchange of assets to different gatherings, single-factor authentication is deficient. Multi factor authentication methods need to be used to

provide secure online transactions using mobile phones. We use multi factor authentication in our scheme, using two different modes. TIC and SMS are used to execute the execution. Although in previous approaches to the issue, SMS was used. We are incorporating the latest definition of TIC as a novel method of authenticating a transaction and the user to previous approaches to the problem. Verification of the TIC (Transaction Identification Code) is the strategy used to distinguish both the current running transaction and also the recipient. The TIC code certifies that the correct person has initiated the current transaction and is a legitimate user attempting to access his/her account.

Codes for TICs are:

- Issued to the consumer by a bank or financial institution.
- 8 bit or 16 bit Pseudo-randomly generated code assigned to the clients.
- Complicated sequences of digits or combinations of binary or alpha numeric characters may be possible.
- The generated TIC will be used for a single transaction itself

Here, we presume that financial institutions are responsible for confidentially storing the logic and algorithm of TIC generation and have their standardized to determine the complexity of the TIC format. Financial institutions are also responsible for updating the logic and data for the TIC generation from time to time and also for maintaining it completely secret. As needed, the consumer will receive a list of TIC codes from the bank or financial institution.

During an online web transaction, the bank or financial institution will make a register of the given TIC codes for its customers and match the same code. After each good transaction, a TIC code is cancelled. According to issuing organizational policies, we can also settle on a validity period of TICs, which incorporates an enhanced security functions of the system.

Web-Based Authentication:

First, using a web-based username/password simple authentication method, the user has to prove their identity to the web authentication server.

TIC Authentication:

The web authentication server requires a TIC code from the user after authenticating the user by username/password. To uniquely identify their transaction and prove its authentication to the web authentication server, the cell phone/PDA user can now insert a one-time TIC code. The TIC code will be chosen from the stored list of TICs issued and uniquely assigned to its customers by the approved financial institution.

Email/SMS confirmation:

So as a final authentication to complete our online transaction , The customer will receive a message of OTP from the platform of the servicenow at the end of the transaction to validate their financial transaction. By this our online transaction will be authenticated in multiple Factors.

3.3 Designed Protocol:

Here in the below designed protocol, the security of transactions will be taken care by the encryption modules itself. The module for encryption and decryption is mounted on the user's cell phone/PDA and on the server side. We can see from the above explanation that the user and transaction are authenticated again using the TIC mechanism after authenticating the user using the basic login/password mechanism, and the transaction is checked again using SMS. Therefore using various media, we have multi-factor authentication and verification measures. Now let's get into flow of protocol execution,
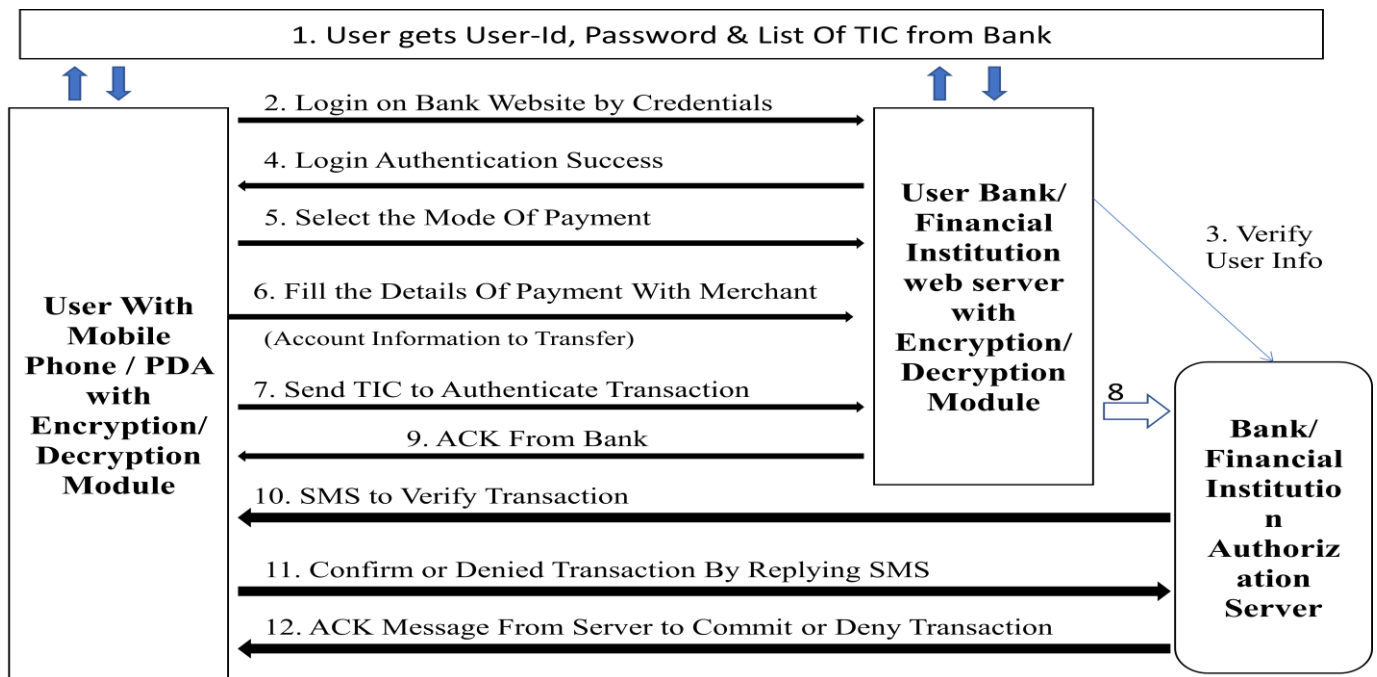
Figure 5: Designed Protocol for Multi-Factor Authentication

Flow of Execution:

1. Initially the client will receive a list of tic code's along with the login credentials from the bank. Each customer has only one username/password in their record, anyway for each online trade, the tic code is uncommon. Customers can obtain a once-over of tic codes as demonstrated by their subtitles from the bank authority or affirmed money related establishment.

2. Basic authentication of a Web-based username/password is used to identify the user to the Web

21

server.

3. A Bank Verification Worker can approve the username and secret key. The client will get a decision screen after client affirmation to continue further.

4. The user will be notified of an effective logging phase with a welcome message. A session key is also produced by this phase.

5. So now here by the client/user has a choice of transactional modes i.e Two payment methods were considered: the system based on credit cards & the electronic transfer based on accounts. Adding other modes to our device is straightforward.

6. By filling in a simple form with details such as merchant bank and branch code information, invoice number and account number to which an amount must be transferred, the user can insert the payment details.

7. By basically picking a tic code from the put away rundown of tic codes, the client can embed a tic codes code. Note that tic codes are put away with nearby encryption on the cell phone, with secret key assurance. Notwithstanding give extraordinary validation to each web exchange, the client will decode and embed the tic codes into the monetary exchange. All exchange information, with the tic codes connected, will be additionally scrambled by the AES encryption method and shipped off the web worker of the bank. It would be sent to the confirmation worker by the bank web worker where it would be unscrambled and coordinated with the rundown of tic codes that were given to the client. Conversations on the technique for encryption/decoding utilized in the proposed conspire.

8. The worker for bank approval decodes the message got and extricates the tic codes. It at that point tests the tic codes got from the client by contrasting it and the tic codes list put away on the data set worker in the client account data. It drops the pre-owned tic codes from its information base if the two tic codes coordinate, and goes to the following stage. In the event that no tic codes co-ordinates those in the information base, the confirmation worker denies the exchange to the client and presentations a blunder message to the client.

9. Now the email/SMS authentication of OTP will take place

10. After the database update for the ongoing transaction is completed, the authentication server will send an SMS to the cell phone of the user to check the initiated web transaction. The user's cellular phone number is available on the authentication server.

11. Here now the validation of OTP/SMS confirmation and validation will go on accordingly as per the flow of execution.

12. The server will notify the user of the successful completion of the transaction or refusal of the transaction through a message.

Generation and Distribution of TIC:

The generation of TIC codes at the financial institution server, distribution of TIC's to the customer and encryption of TIC's before storing them in the client environment is another essential module in the protocol. TIC codes are pseudo random codes and can be generated on the web authentication server with the pseudo random number generation algorithm as previously stated, TIC generation logic is strictly confidential and we assume that the banks will update TIC data generation data and improve TIC generation algorithms.

The financial institution's approved individual is responsible for transmitting TICs to the user's mobile phone through a simple data cable system, and the delivery method involves encrypting TICs in order to protect security confidentiality. We have assumed that TICs are stored in the database on the server side and that the database management system (supported by Oracle 10g) and operating system with stable firewalls are strongly protected to protect server side data.

Architecture Of The Servicenow OTP Generation:
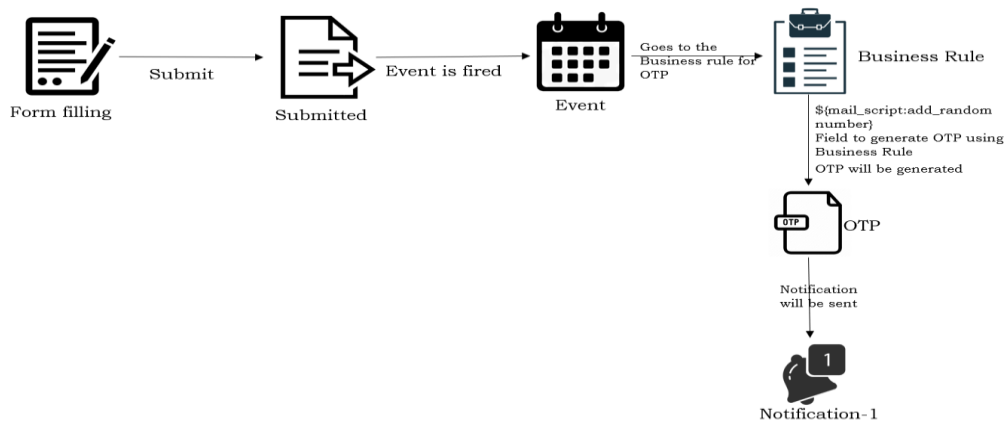
Figure 6: Architecture Of the OTP Generation

1) User Fills the Form with his Name, Email and submits the form.

2) After Submitting the form, An event will be Fired.

3) The Business Rule which uses the Event is now registered and activated for further step and the Generation of OTP is Done within the business rule as the script is executed.

4) The Notifications/Alerts which are connected with the business rule carries the OTP that was generated in the Script and will be sent to Email of the User

5) User will be entering the OTP for secured Transaction

CHAPTER 4: EXPERIMENTAL INVESTIGATIONS

We provided an overview of the protocol proposed in the current work in the previous chapter. In this chapter, we look at our system's comprehensive message flow. This is important in order to undertake a thorough examination of the hazard. In describing the transaction flow of the proposed system, we have considered five major components, as described in the previous chapter:

1) Customer Agent (CA),
2) Customer Bank (CB),
3) Customer-Bank Authentication Server (CBAS),
4) Merchant Bank (MB),
5) Merchant Agent (MA).

The message flow is described in four sections, each of which represents one step of the total process. Message streams are illustrated using sequence diagrams along with detailed descriptions. The phases shall consist of:

1. First authentication of a user by a bank authentication server, i.e. simple user authentication based on username and password.
2. Two-way authentication, i.e. merchant / manufacturer / service supplier authentication.
3. The user's second authentication and the transaction with TIC.
4. Final authentication and transaction confirmation by the user, via SMS.

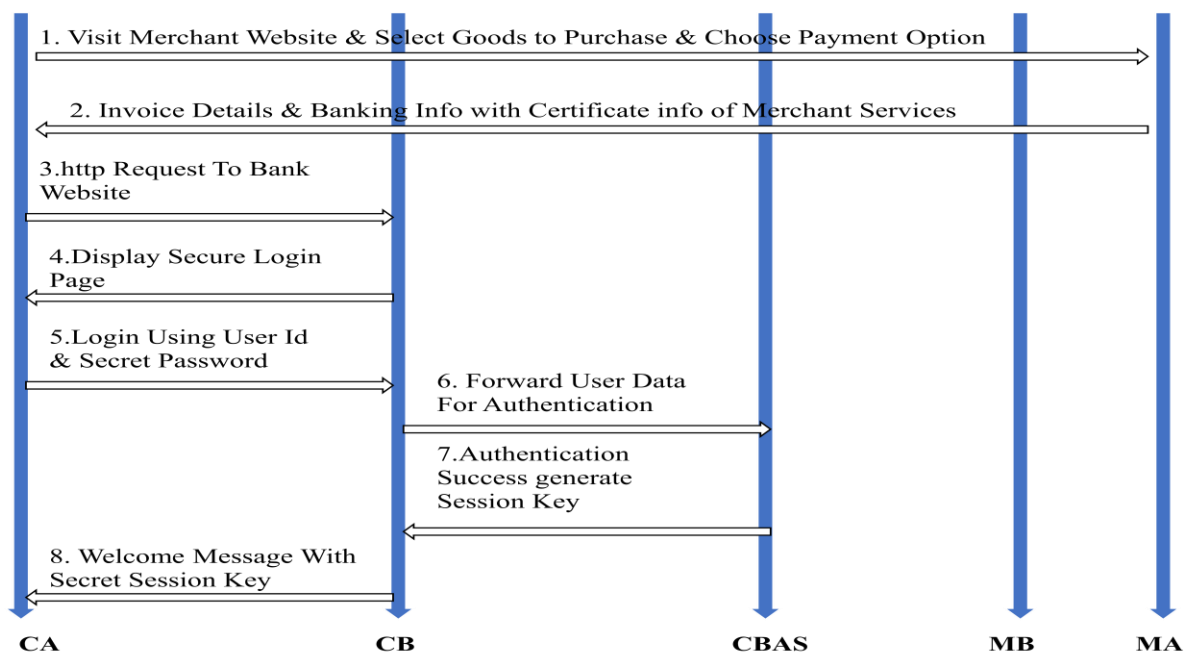**First authentication of a user by a bank authentication server**

Figure 7: First Authentication of user to Bank

1. User (CA) visits the merchant's website to make online transactions, purchases and choose a payment option from the website.

2. The merchant web server (MA) produces descriptions of the invoice and the merchant Banking information is sent to the CA in an encrypted format with a merchant authentication certificate. Note that mobile phones have common capabilities for accessing and transmitting data over wireless cellular networks with encryption/decryption.

3. The user (CA) creates a http request to the web server of his/her bank. Pay transaction, please.

4. The CB web server shows a stable login page for the web server to log in.

5. The user logs in using the user ID and the user's only known hidden password. Until moving user login information from client to server, they will be encrypted using the symmetric key cryptography that is enforced by the standard security protocol on the bank server and the user cell phone.

6. User data will be forwarded to the authentication server of the local bank. Notice that we have suggested that a separate authentication server be established at the bank or financial institution to ensure a robust security mechanism.

7. User login data from the authentication server will be decrypted and matched with its protected customer account information. On completion, a random session key is created that is encrypted by shared secret logic and transferred to the Web server of the Bank.

8. The CB will submit a general text welcome message and a session key to monitor the user's session (CA). If the first user authentication fails, an invalid login message will be sent to the user.After successful completion of the user's first login/password authentication, the two-way authentication will take place as indicated below.

Authentication of merchant to the customer
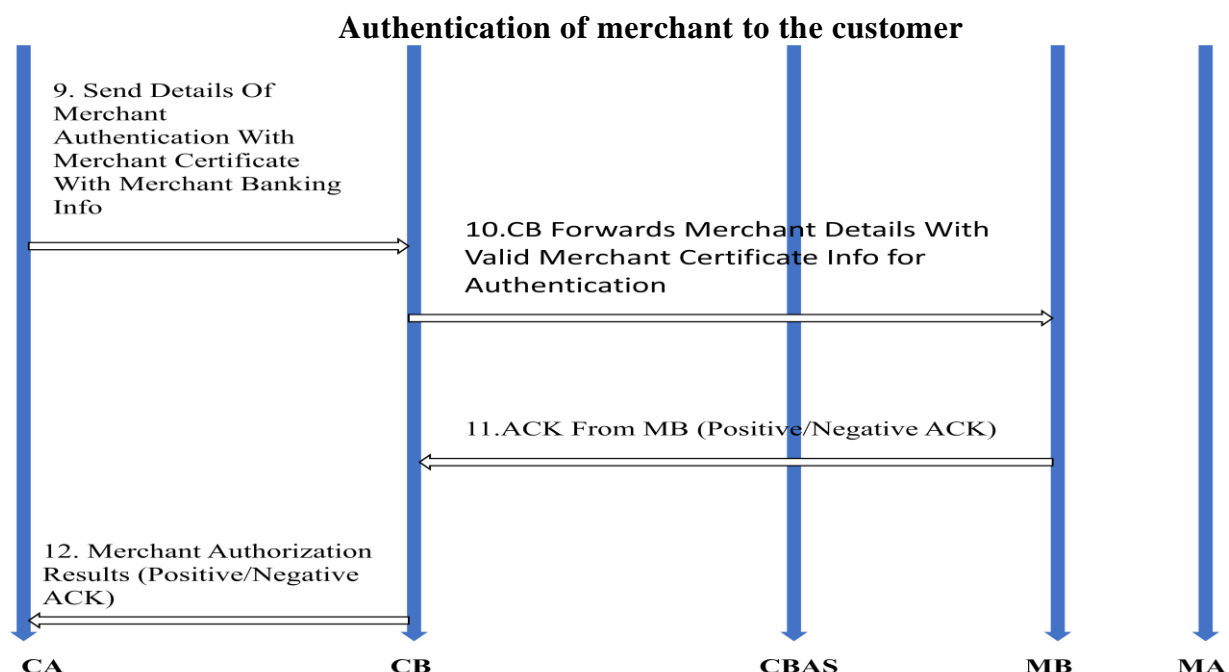
Figure 8: Two way authentication

9. The User (CA) will submit the request to the User's Bank for Commercial Authentication prior to payment to MB. The CA shall forward the description of the merchant obtained from the merchant in order to authenticate the merchant.

10. Here we have assumed that both the Consumer Banks (CBs) and the Merchant Banks (MBs) have a business model and are bound by the legal terms and conditions of the standard policies agreed by their business organizations. The CB would then make a request to the merchant bank to authenticate the merchant. Each merchant shall have a legal authorisation certificate provided by his or her banks or by other centralized financial institutions to authenticate the services of the merchants.

11. The MB will approve the request for authentication of the trader after matching information given by the CB to the merchant. Acknowledgement can be made good or negative depending on the validity of the trader certificate.

12. The CB shall forward the acknowledgment of receipt to the merchant Authentication of the consumer (CA). Note that if MB were to have negative acknowledgement then CB simply terminates the user transaction for a legitimate security reason and if CB receives a positive acknowledgement from MB then it is presumed that the trader is valid and the user will proceed to the payment.

As stated in above in the previous section, it will run a multifactor safe web authentication protocol to make payment. In Part III, a thorough description of the transaction flows of this protocol is given.

The user's second authentication and the transaction with TIC.

Figure 9: Second Authentication of User to the Bank

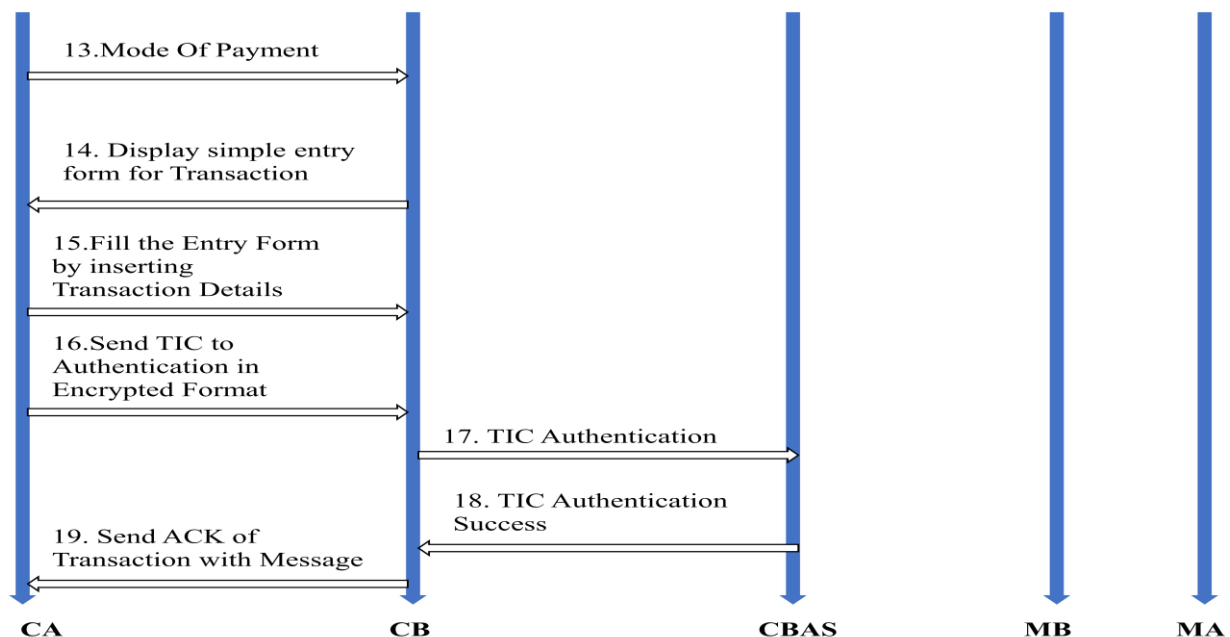13. The customer (CA) can choose the mode of payment to make the payment. Here we have considered two simple modes of payment:

14. Selecting the mode of payment produces an entry form with the necessary fields.

15. By filling basic entries such as amount, account number to which amount has to be transferred, if there is a merchant payment, the user can fill in the transaction information by automatically selecting the invoice number and other merchant details provided by the merchant.

16. The TIC code will be installed by the user (CA) by opening the list of TICs stored on the TIC. Client side setting and transaction request in an encrypted format to the server.

    - The TICs are stored in the mobile basic with the security encryption of mobile built in encryption standards.

    - The user inserts the local TIC password to open the TIC list and can pick any TIC from the list.

    - The local TIC password may be the key for TIC decryption, and the key for TIC decryption may be known to the user only or stored by hidden logic. This key is unknown to even the bank authentication server. It can be altered according to the user's convenience at any moment.

    - TIC transmission from CA to CB is exclusively in an encrypted format.

17. The bank server will forward the received TIC to the TIC authentication CBAS and the received encrypted TIC will be decrypted by CBAS in order to align it with its database.

18. In order to sync it with its database, the bank server will forward the received TIC to the TIC authentication CBAS and the received encrypted TIC will be decrypted by CBAS.

19. If the received TIC matches the user's allocated TIC list, the bank server produces a user's receipt with a message waiting for the SMS. If the TIC does not fit the allocated TIC list of the user, it rejects the current transaction and sends the user a notification that the transaction is cancelled due to invalid TICs. After a good TIC match, the user is free to close the current user session or make a new financial transaction. The next step is that the authentication server produces an SMS for the user.

20. The CBAS customer sends an Email/SMS with transaction information to validate the transaction by the recipient.

21. The user can respond to the SMS by choosing "YES" or NO." SMS response "YES" means that the customer is legitimate and confirms the transaction. SMS reply "NO" means that the user rejects their transaction. Or else we can enter the OTP which is sent to the registered email and we can validate the transaction through that OTP

22. If the bank authentication server receives "YES" from the user's SMS confirmation or The OTP

entered validates perfectly then, it produces a payment notification to the user and commits the user's transaction.

23. The bank server will also submit notice of payment to the MB with the invoice number and other customer details requested.

Final authentication and transaction confirmation by the user
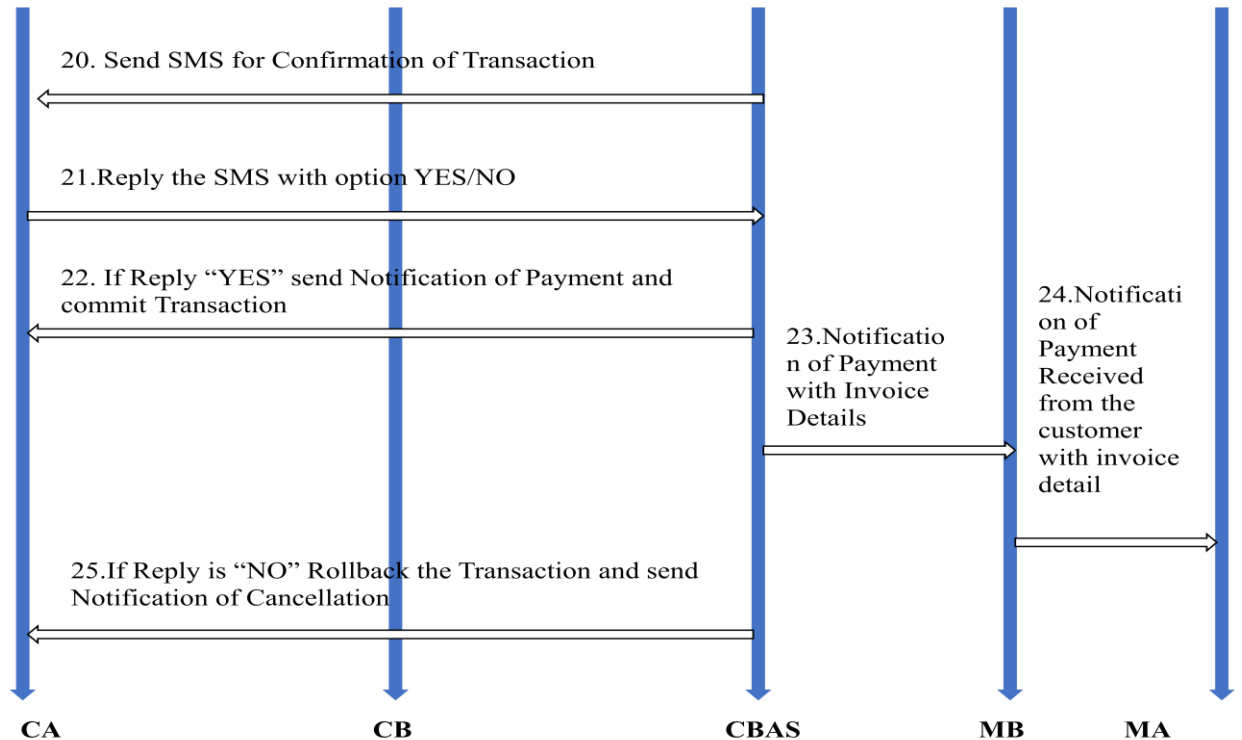
Figure 10 : Third Authentication of user to the Bank

24. The MB shall be responsible for sending notice of the payment received from the customer to the merchant. The notice shall contain payment details such as the invoice number and other customer information requested.

25. If the CBAS gets a wrong OTP validation or the SMS confirmation is not appropriate then, the transaction automatically rollbacks and notifies to the CA.

Procedure Of Generating One Time Password:

2.1 Creation Of Table:

A new Table "User Details Email Generation" is created with the Fields Name(String), email-id(String) and Submit the Table . After Submitting the table refresh the instance page and check whether the table is created or not with the fields that we have given or not.

2.2 Creation Of Event:

Events registry is a list of the events that the device is aware of. Other operations, such as script actions or alerts, may be automated using recorded events. when we create an event the business rule that uses the event must activate it to the further process Activation/Registration of the business rule in the event lets others as Email Notifications and Script Actions, see the event in their list of available events and react to the event when it occurs.

In studio create application file and search for event registry and create it and give the suffix as "randomnumber" under the table "User Details Email Generation" and submit it.

2.3 Creation Of Business Rule:

A business rule is a script on the server side that runs anytime a record is viewed, added, modified, removed, or queried from a table. Server scripts run on a computer or server. When documents and tables are accessed or updated, they can alter the appearance or actions of ServiceNow or act as business rules. Application Programming Interfaces (APLI) server-side Glide offers specifications that you can use to execute serverside functions in scripts. Company rules are used to execute activities such as automatically modifying values in the required fields when other parameters are reached, or to build email update events and script actions. Now open studio, create a business rule for "User Details Email Generation" table. Name the business rule as "BR1" and tick the checkbox for "Advanced" field. At the bottom of the business rule page, go to the advanced tab and Give the Code as:

(function executeRule(current, previous /*null when async*/) {

   var x = Math.floor((Math.random() * 90000) + 100001);

   gs.info('Random: '+ x);

   gs.eventQueue('x_473281_final_yea.randomnumber',current,x,'');

   gs.info('Random');

})(current, previous);

2.4 Creation Of Notification:

Notifications are used for the updation of any process or informing about the updates are to follow up for the next step in any process or automation of any workflow.

In Studio page create a notification named OTP Generator for the table "User Details Email Generation" and Under "WHEN TO SEND" tab select the option called as "Event is fired" for "SendWhen" tab and select the event name that we have created previously and under "WHO WILL RECEIVE" tab select the users/groups in fields and select email field. Under "WHAT WILL CONTAIN" tab give the subject and message as your requirement and the OTP generator field is ${mail_script:add_random_number}This is the field which will generate a random number for every submission of a form.

Cryptography and Key and session management:

There are many facets of security and many implementations, ranging from safe trade and payments to private communications and password protection. A key aspect of secure communications is that of cryptography.

Public-key-encryption techniques are very common methods of encryption and are some examples used in many application areas such as application data protection, security of operating systems, network security and Digital Rights Management (DRM). The Internet Engineering Task Force (IETF) is an agency created to establish cellular network environment requirements for Internet and mobile platforms. Public Key Infrastructure (PKI) is also generally accepted for secure communication on wireless networks in the cellular network environment. In contrast to symmetric-key encryption, public-key encryption requires more computational and processing time. Public-key encryption is also not always sufficient for vast volumes of data communication. Public-key encryption, however, is used to share a symmetric key, which can be used for more data encryption later on.

The Encryption Algorithm:

Here as per our requirement we are seeking for the use of AES Rijndael algorithm. AES Rijndael algorithms support the iterated block cipher, which means that multiple transformations take place on the input block before output is generated. It also supports variable-length blocks with key sizes of 128, 192, or 256-bit of respective data blocks. The algorithm of AES Rijndael has the following characteristics:

1. Resistance

2.Compactness

3.Simplicity

Cipher key Management:

Secure contact between the client and the server is our primary objective. For this purpose, we have introduced a secret session key principle, the session key is created randomly for each client session and then used for data encryption/decryption. The system works as follows: the server uses a 128-bit key. At the start of the user session, the server randomly produces a secret key (128 bits) and stores it individually for each user. The server uses 128 bit shared secret logic to encrypt this secret key, which is a shared secret known to both ends, i.e. the client and the server. This encrypted secret key is forwarded to the client. The client decrypts this secret key and uses it at the later stage of the transaction to encrypt the customer's required account information with the TIC code before it is sent to the server. The same secret key is stored on the server side, which decrypts transaction information and TIC code, then matches the TIC code to the client with the given TICs. If this TIC fits the database, then the transaction will be processed next, otherwise the transaction will be rejected.

A significant problem is the security of shared hidden logic for the client and server. This mutual secret is stored on the server in a database that we believe is protected by a database management system, a secure firewall operating system, and other information security policies. Shared secrets are stored in limited mobile device memory on the client device, ensuring this storage is a little difficult. We propose the following method for preserving the client's mutual secret:

The shared secret is stored after encryption and key is the confidential 128 bits pin code
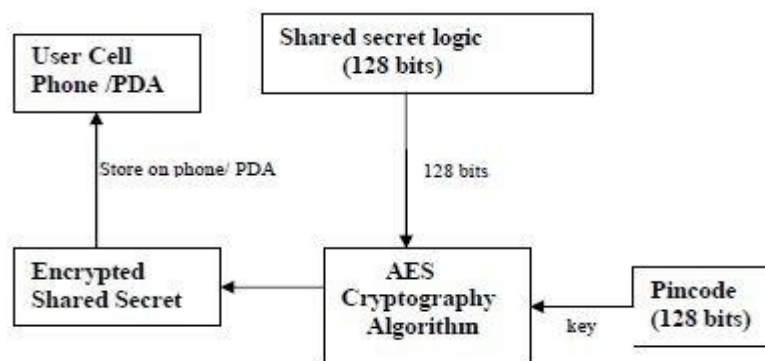
Figure 11: Storing of shared logic of client

Here we propose that post the subscription of mobile banking from the BA, the bank authority makes it possible to install a banking module on a cell phone and store shared secrets on a mobile phone/PDA after encryption with a hidden 128-bit pin client code number. If the user's phone is misplaced or stolen, the criminal cannot use this shared secret logic since the shared secret is stored in encrypted

format and the device decrypts it when the user logs in to the correct client authentication portal. This client pin code is a shared secret decryption key used later in the data/key encryption/decryption process.

In addition, we can make the client more secure by obfuscating Java classes and JAR files. Protects the binary code from the byte code de-compiler .

The most sensitive data stored on the mobile phone/ PDA are the TIC codes in the proposed protocol. To keep the safe, we suggested that TICs be stored in an encrypted format and password protected on the cell phone/PDA as shown in Figure . To open an encrypted list of TICs, the user enters a local TIC password and can pick any TIC from the list to initiate financial transactions. This TIC selection automatically decrypts the selected TIC and displays it on the user screen. The local TIC password could be the key for TIC decryption, or only a secret password to secure TICs and TICs will be decrypted by the user's only known internally stored confidential secret logic. It is unclear to even the server. It can be altered according to the convenience of the user at any time. The local TIC encryption and decryption is also based on the algorithm for the AES symmetric key. The AES cryptographic algorithm is better suited for small devices, improving the speed of cryptographic processing efficiency over small handheld devices rather than degrading the performance of the system.
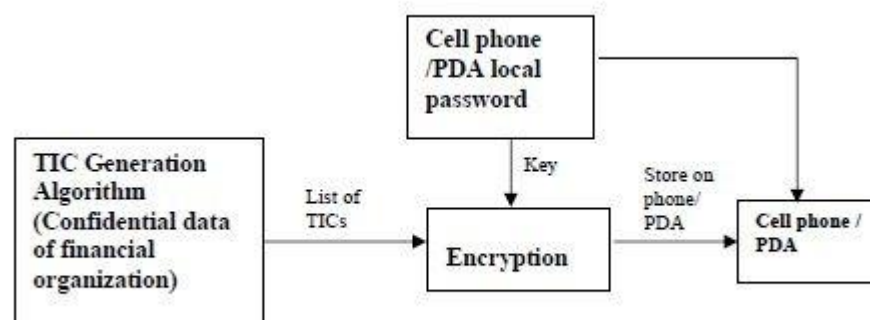
Figure 12 : TIC protection in client Environment

System Analysis with various Internet Threats:

The proposed technique can handle different internet threats, such as phishing, mobile phone failure, etc. We present a thorough overview of our system's resistance to different internet threats in this chapter. In each case, we examine the data that an attacker might have and the particular points in the protocol where a fraudulent transaction would not be carried out by the attacker.

Security against Phishing:

The attack of phishing fraud has become a common user identity theft technique. attacker's fraudulently captures users' sensitive data such as passwords and credit cards, Information for acquiring unauthorized access to the confidential financial data of the user and carrying out the illicit transfer of funds. Phishing is usually done by email or an instant message or through telephone communication.

The proposed protocol is safe from phishing attacks. A secure user authentication multifactor protocol has the ability to secure user data and preserve malware access integrity, confidentiality and access control. We consider different cases below to grasp the origin of the security.

Case I: If attacker acquires login credentials:

Secret codes issued to legitimate account holders are TIC codes and TICs are not publicly available. For each online transaction, it is a one-time code and it is created in nature randomly so that a attacker can not guess the next user account TIC code.

Case II: Transmission of TICs over insecure channel:

The transmission of TIC code from the user's cell phone/PDA to the web authentication server is in a heavily encrypted format, so it cannot be easily decrypted by phishing attackers to access private user information on the server side. In addition, only once is one TIC used and then discarded.

Case III: If attacker acquires login credentials along with the one of the TIC codes:

The TIC codes are pseudo-random in nature, even if a phishing attacker receives a sample of TIC code from any phishing technique, the attacker will not produce the next TIC code because the logic of TIC generation is strictly confidential on the web authentication server and we believe that banks and financial institutions are responsible for updating TIC generation data from time to time and updating TIC generation.

Virus attack on Cell Phones and PDAs:

Various virus attacks can also cause mobile wireless devices to suffer. Viruses of mobile devices such as "Cabir" and "CommWarrior.A" may migrate the data without the knowledge of user through any of the services available for it such as Bluetooth or messaging apps or some other stuff's.

The proposed system is safe even if some mobile device virus attacks the cell phone of the user. No virus may interrupt the data of the user or damage the confidential data stored on the cell phone/PDA. The device is secured from virus attacks by the following points:

1. Users will always bring their mobile phone with them so that SMS confirmation will not be present

in the event of a malicious transaction from any unauthorized user who has access to sensitive user data from virus attacks.

2. The TICs are stored in an encrypted format and password protected, so that the person who has acquired information illegally will still be unable to decrypt the TICs and any virus attacks will not be able to disrupt the data of the consumer.

3. It is always recommended that users check the access permissions in their Bluetooth settings before making a connection and close active Bluetooth connection if users do not use it to avoid downloading the mobile device virus through a Bluetooth connection. Users should install anti-virus apps on many mobile platforms to defend themselves against viruses.

User Session Hijacking:

Using malicious software, the activities of users on the internet are recorded in this attack and users are unaware of it. Malware software tracks all user activities over the internet from the local computer of the user or remotely after user session control known as Session Hijacking. Our stable protocol provides protection in the following stages to resolve this threat:

1. In Figure after successful completion of Step 5 and Step 6 authentication servers Creates session id as shown in the steps in Figure . The secret to this session It will be passed to the user in an encrypted way to create a protected environment. Session, please. Additional Http requests from the user can use the session id to send a request to the server. If the server receives an unauthorized Http request that does not contain the session id created by it the service will be denied.

2. TIC codes, which are one-time codes for each transaction, will be deleted from the database by the web authentication server after each transaction. So if a man assaults the user session in the middle to monitor user behaviors, he/she gets a TIC that has already been canceled.

3. The TIC codes are pseudo randomly generated by a confidential algorithm; this is a complex code that cannot be easily predicted by anyone.

4. Sensitive and confidential transaction details must be encrypted prior to channel transmission.

5. As mentioned above, TIC transmission over the user session is also in a highly encrypted format, and the web authentication server generates a specific secret encryption key.

6. We used 128-bit shared secret logic between the server and the client to transmit the unique secret key to the client on every login. There is therefore no need to relay this shared logic to the dysfunctional medium, as it is understood at both ends.

7. Another security aspect is the confirmation of the SMS as seen in the figure. The SMS is not routed

into the same channel used in the online web transaction. The SMS uses cellular network control networks. Messages are encrypted with an A5/3 algorithm that also makes the device safe.

Cell Phone /PDA Theft:

A big downside to mobile devices is that they can be misplaced or stolen. If the user's phone is lost or stolen, the user can immediately suspend his connection to the wireless service in order to defend himself from charges and unauthorized access. However before it has been de-activated, it is entirely possible that an unauthorized entity could attempt to initiate a transaction with the lost/stolen cell phone. For the following purposes, the suggested protocol protects the user from this contingency.

1. Because of the above factor, we believe that people are losing their cell phones, they should deactivate their wireless service and take the appropriate action to report lost. Once disabled, the user will not be able to receive the generated SMS for their number.

2. Another significant security issue is that if a person has stolen a cell phone/PDA user, the thief does not know the local password of the stored cell phone/PDA user's TICs. If the user's bank account password is revealed to the thief, he/she cannot misuse the user account because the thief has no access to the TIC codes that are stored in an encrypted password-protected format.

3. If the user's mobile phone/PDA is lost or stolen, it is highly advised that the user take the required deactivation action and immediately order the bank to cancel the user's entire list of provided TICs.

CHAPTER 5: IMPLEMENTATION

5.1-TECHNOLOGIES:

HTML:

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

JSP:-

JSP might be seen as a significant level reflection of Java servlets. JSPs are converted into servlets at runtime, consequently JSP is a Servlet; each JSP servlet is stored and re-utilized until the first JSP is altered.JSP permits Java code and certain predefined activities to be interleaved with static web markup content, for example, HTML.

SERVLETS:-

A Java servlet is a Java programming segment that expands the abilities of a worker. In spite of the fact that servlets can react to numerous kinds of solicitations, they most generally execute web holders for facilitating web applications on web workers and in this manner qualify as a worker side servlet web Programming interface.

ORACLE:-

Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is a proprietary multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW) and mixed (OLTP & DW) database workloads.

JDBC(Java Data Base Connectivity):

The JDBC driver is used at server side to establish a communication between java servlets and database. The JDBC API provides facility to connect java program to relational database to access data. In the JDBC API, user can execute SQL statement in java programs, retrieve the fetched results and make changes in data source to update database. The JDBC API can operate in distributed or heterogeneous environment by connecting multiple data source.

The object of JDBC DriverManager class is used to make connection between Java applications and a JDBC driver. The DriverManager is very small in size and simple to operate, it plays a very important role in the JDBC architecture.

SERVICENOW EMAIL NOTIFICATIONS:

Service-Now is a tool that helps to work on digital workflows for enterprise applications, ServiceNow Notifications to oversee framework email, make framework notices, and design how your framework reacts to inbound email.

Making an email notice includes determining when to send it, who gets it, what it contains, and in the event that it tends to be conveyed in an email digest. These are the basic sections of email notifications when to send, whom to send, what to send..

5.2-Source Codes:

Home.html:

```
<!DOCTYPE html>

<html> <head> <title>Multi_Factor_Authentication</title> <style>

body{

 background-image: url('bg6.jpg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: cover;

}
p {
 background-image: url('homebgff.jpeg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: cover;
}
body{
background-color:black;
color:white;
}
button{
padding:13px 10px;
}

h1 {
 text-align: center;
 text-transform: uppercase;
 }
p{
 font-family: "Times New Roman", Times, serif;
}
body {
```

```
  font-family: Arial, Helvetica, sans-serif;
}

.navbar {
  overflow: hidden;
  background-color: grey;
}

.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.navbar a:hover {
  background-color: black;
}

</style> </head> <center> <h1>A multifactor security protocol for wireless payments</h1>
<body>
<div>
</center>
<div class="navbar">
        <a href="#home"  onclick="location='home.html' ">Home</a>
 <a href="#home"  onclick="location='login.html' ">Login</a>
 <a href="#home"  onclick="location='register.html' ">Register</a>
 <a href="#home"  onclick="location='about.html' ">About</a>
</div>
<p><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br
><br></p>
```
 MFA provides a highly secure wireless environment for financial web transactions that is simple to use and deploy, that does not
 require any change in existing wireless networks or protocol. This Protocol for Wireless
 Payment is used to achieve secure web transaction using cell phones / PDAs.This system is based on Multi-factor Authentication
concept to provide secure wireless environment
 to the users to increase faith of the users in online financial web transactions using mobile
 devices.
```
</body> </html>
```

Note: The above header style sections are same for all html & jsp pages

Login.html:

```
<html><head><style>
.image
{
 background-image: url('bg5.jpg');
 color: white;
}
p{
background-color:black;
}
body{
background-color:black;
color:white;
}
button{
padding:13px 10px;
font-size: 15px;
}
```

```
h1 {
 text-align: center;
 text-transform: uppercase;
 }
p{
 font-family: "Times New Roman", Times, serif;
}
body {
 font-family: Arial, Helvetica, sans-serif;
}

.navbar {
 overflow: hidden;
 background-color: grey;
}

.navbar a {
 float: left;
 font-size: 16px;
 color: white;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
}
.navbar a:hover {
 background-color: black;
}

</style>
<title>Login page (1FA)</title>
<center>
<h1> A Multifactor Security Protocol For Wireless Payments </h1>
</head>
<div class="navbar">
 <a href="#home" onclick="location='home.html' ">Home</a>
 <a href="#home" onclick="location='about.html' ">About</a>
</div>
<body>
<div class="image">
<H3>LOGIN</H3>
<img src="207-2070782_new-svg-image-login-icon.png" width=280 height=250>
<br><br><br>
<form action="./client_login">
USER-ID:<input type="text" name="id">
<br><br>
PASSWORD:<input type="password" name="pwd">
<br><br>
<input type="submit" value="LOGIN">
</form><br><br><br><br></body>
<hr> </div>
<h3><i>"Security and Safety were the Reward of Dullness"</i></h3>
</center></html>
```

Login2.html:

```
<html><head><style>
body{
background-color:black;
color:white;
}
button{
padding:13px 10px;
}
```

39

```
.navbar {
  overflow: hidden;
  background-color: grey;
}
.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.navbar a:hover {
  background-color: black;
}
</style>
<title>Login Page (1FA)</title>
</head><body><center>
<BR><BR>
<div class="navbar">
  <a href="#home"  onclick="location='home.html' ">Home</a>
</div>
<H3>LOGIN</H3>
<img src="207-2070782_new-svg-image-login-icon.png">
<br><br><br>
<form action="./custlogin.jsp" method=post>
User-Id:<input type="text" name="id">
<br><br>
Password:<input type="password" name="pwd">
<br><br>
<input type="submit" value="login">
</form></center></body></html>
```

About.html:

```
<!DOCTYPE html>
<html><head>
<title>MFA</title><style>
body{
background-color:black;
color:white;
}
button{
padding:13px 10px;
}
h1 {
  text-align: center;
  text-transform: uppercase;
 }
p{
  font-family: "Times New Roman", Times, serif;
}
body {
  font-family: Arial, Helvetica, sans-serif;
}
.navbar {
  overflow: hidden;
  background-color: grey;
}
.navbar a {
  float: left;
  font-size: 16px;
```

40

```
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}
.navbar a:hover {
    background-color: black;
}


</style></head><center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
<body><div></center>
<div class="navbar">
  <a href="#home" onclick="location='home.html' ">Home</a>
  <a href="#home" onclick="location='login.html' ">Login</a>
</div><br><br>
<img src="mfaabout.png" width="600" height="300" style="float:right">
<h3>What is Multi-Factor Authentication (MFA)?</h3>
```

Multi-factor Authentication (MFA) is an authentication method that requires the user to provide two or more verification factors to gain access to a resource such as an application, online account, or a VPN. MFA is a core component of a strong identity and access management (IAM) policy. Rather than just asking for a username and password, MFA requires one or more additional verification factors, which decreases the likelihood of a successful cyber attack.

```
<br><br><br>
<h4>Three Main Types of MFA Authentication Methods</h4>
```

Most MFA authentication methodology is based on one of three types of additional information:
->Things you know (knowledge), such as a password or PIN
->Things you have (possession), such as a tic codes or smartphone verification
->Things you are (inherence), such as a biometric or voice recognition or otp

The way we live has been reformed by SmartPhones. Cell phones and PDAs have been packed with ubiquity to a great extent and consumers have now started webbased banking, webbased item purchasing and other online administrations. Either the site or the remote flexible channel has been used autonomously by previous web access authentication systems to confirm the personality of faroff clients. To get to the latest online administrations reliably needs a username and secret phrase to verify the client personality.

This is a major weakness because the hidden word can be compromised and later used by the man in the center attack to make illegal entry to the record of the client. We are likely to make a validation system based on a multi-faceted verification method that is both reliable and deeply accessible. It features a proprietary way to deal with rendering a validation system based on spasms (Exchange Recognizable proof code) and SMS (Short Message Administration) in order to enable the conventional Login/secret phrase framework to get an additional security level. In addition, we use an encryption method which depends on a symmetric key and an idea of an iterated block figure. This idea has been used to retain Spasms on PDAs/PDAs as a mystery code but is also used to start secure web transfers using mobile phones/PDAs. Finally, we extend the two-way validation framework that confirms the two players (client and e-specialist organization). A definite threat analysis shows that the proposed system is safe from various types of web attacks such as phishing, man-in-the-center, infections and anything similar as outlined above. In this project we are proposing a new system and a protocol for Secure way of transactions by using Multifactor Authentication System.</body></html>

Register.html:

```
<!DOCTYPE html>

<html><head>
<title>Multi_Factor_Authentication</title>
<style>
.image
{
    background-image: url('bg5.jpg');
    color: white;
}
body{
background-color:black;
color:white;
}
button{
```

```
     padding:13px 10px;
}
h1 {
  text-align: center;
  text-transform: uppercase;
 }
p{
  font-family: "Times New Roman", Times, serif;
}
body {
  font-family: Arial, Helvetica, sans-serif;
}
.navbar {
  overflow: hidden;
  background-color: grey;
}
.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.navbar a:hover {
  background-color: black;
}
</style></head><center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
<body><div></center>
<div class="navbar">
  <a href="#home"  onclick="location='home.html' ">Home</a>
  <a href="#home"  onclick="location='about.html' ">About</a>
</div>
<div class="image"><br><br>
<img src='register.jpg' width="630" height="390" style="float:left">
<br>
<form action="./client_db" >
<center>
CUSTOMER-ID: <input type="number" name="cid" required><br><br>
NAME:&nbsp &nbsp <input type="text" name="name" required><br><br>
USER-ID:&nbsp <input type="text" name="id" required><br><br>
PASSWORD:&nbsp <input type="password" name="pwd" required><br><br>
ADDRESS: <input type="text" name="add"><br><br>
PHONE:&nbsp &nbsp <input type="text" name="ph"><br><br>
AMOUNT:&nbsp &nbsp <input type="number" name="amt"><br><br>
EMAIL:&nbsp &nbsp <input type="text" name="email" required><br><br>
GENDER: &nbsp &nbsp
Male<input type="radio" name="gdr">&nbsp
Female<input type="radio" name="gdr"><br><br>
<input type=checkbox name="tandc" required> I accept all the terms and conditions <br><br>
<input type =submit value=SUBMIT style="height:40px; width:90px">
</center> </form>
<br><br><hr></div><center>
<h3><i>"Security is the Chief Enemy of Mortals"</i></h3>
</center><br><br></body></html>
```

Connection.jsp:

```
<html> <body>
<% @page import="java.sql.*"%> % @page import="java.io.*" %
<%! Connection con;%>
```

42

```
<% Class.forName("oracle.jdbc.driver.OracleDriver");
 con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");
   %></body></html>
```

dbconnect.jsp:

```
<body>
<%@page import="java.sql.*"%>
  <%@page import="java.io.*" %>
   <%! Connection con;%>
    <% Class.forName("oracle.jdbc.driver.OracleDriver");
 con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");
%> </body>
```

Custlogin.jsp:

```
<html>
<%@ include file="dbconnect.jsp" %>
<body>
<%
try
{
String v1=request.getParameter("id");
String v2=request.getParameter("pwd");
Statement stmt=con.createStatement();
System.out.println("Runned");
String s="select custname from cust where username='"+v1+"' and pwd='"+v2+"'";
int n = stmt.executeUpdate(s);
if(n>0)
{
        request.getRequestDispatcher("loggedin.jsp").forward(request, response);
}
else
{
        out.println("Invalid Login");
        response.sendRedirect("login2.html");
}
}
catch(SQLException e)
{
        System.out.println(e);
}
 %>
</body>
</html>
```

Loggedin.jsp:

```
<!DOCTYPE html>
<html>
<head>
<title>WELCOME</title>
<style>
.image
{
 background-image: url('bg1.jpg');
 color: white;
 font-size: 18px;
}
p{
background-color:black;
```

43

```
}
body{
background-color:black;
color:white;
}
button{
padding:13px 10px;
font-size: 15px;
}

h1 {
  text-align: center;
  text-transform: uppercase;
 }
p{
 font-family: "Times New Roman", Times, serif;
}
body {
 font-family: Arial, Helvetica, sans-serif;
}

.navbar {
 overflow: hidden;
 background-color: grey;
}

.navbar a {
 float: left;
 font-size: 16px;
 color: white;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
}
.navbar a:hover {
 background-color: black;
}
</style>
</head>
<center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
<body>
</center>
<div class="navbar">
  <a href="#home" onclick="location='loggedin.jsp' ">Home</a>
  <a href="#home" onclick="location='login.html' ">Logout</a>
  <a href="#home" onclick="location='about.html' ">About</a>
</div>
<div class="image">
<br><br>
<% String v1=request.getParameter("id");
out.println("\n\n Hello "+v1+" , we matched your details!");
%>
</div>
<div class="image">
<br>
Here are the Services that are available for you!!
<br><br>
<center>
<button onclick="location='balenquiry.jsp'">Balance Enquiry</button><br><br>
<button onclick="location='creditcard.jsp'">Credit Card Tranfer</button><br><br>
<button onclick="location='transaction.jsp'">Electronic Transfer</button><br><br>
<button onclick="location='generatetic.jsp'">Generate TIC</button><br><br>
```

44

```
<button onclick="location='merchant.jsp' ">Merchant Payment</button>
<br><br><br><br><br><br><br><br>
<hr></div></center></body>
<center><h3><i>"Security is no replacement for liberty"</i></h3>
</center></html>
```

balenquiry.jsp:

```
<%@ include file="connection.jsp" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<style>
.image
{
 background-image: url('bg3.jpg');
 color: white;
 font-size: 18px;
}
p{
background-color:black;
}
body{
background-color:black;
color:white;
}
button{
padding:13px 10px;
font-size: 15px;
}

h1 {
 text-align: center;
 text-transform: uppercase;
 }
p{
 font-family: "Times New Roman", Times, serif;
}
body {
 font-family: Arial, Helvetica, sans-serif;
}

.navbar {
 overflow: hidden;
 background-color: grey;
}

.navbar a {
 float: left;
 font-size: 16px;
 color: white;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
}


.navbar a:hover {
 background-color: black;
```

45

```
}

</style>
</head><center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>

<body>
</center>

<div class="navbar">
 <a href="#home"  onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home"  onclick="location='login.html' ">Logout</a>
 <a href="#home"  onclick="location='about.html' ">About</a>
</div>
<div class="image">
<h3>Dear Customer your balance details are!</h3><hr>
<center><br><br><br>
DATE/TIME:<br>--------------<p id="demo">Current date and time</p>
<script>
var d = new Date();
document.getElementById("demo").innerHTML = d;
</script>
BANK<br>------<br>State Bank of India<br><br>
BRANCH CODE<br>--------------<br>1111<br><br>
<%
Statement stmt=con.createStatement();
String amt="";
ResultSet rs=stmt.executeQuery("select amount from cust where custusername='lohya' and pwd='1234'");
while(rs.next())
{
        amt=amt+rs.getString(1);
}
out.println("Your current balance is "+amt+"/- only..");
%>
</center>
<br><br><br><br><br><hr></div>
<center>
<h3><i>"We have to Feel Money is Nothing , But Again Money Eveything!"</i></h3>
</center>
</body>
</html>
```

Creditcard.jsp:

```
<center>
<h1>A Multifactor Security Protocol for Wireless Payments</h1>
</head>
<body>
<div>
</center>
<div class="navbar">
 <a href="#home"  onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home"  onclick="location='login.html' ">Logout</a>
 <a href="#home"  onclick="location='about.html' ">About</a>
</div>
</div>
<div class="image">
<br><br>
<center>
</center>
<br>
<br>
<center>Please Enter your credit card details--><br><br>
```

```
BANK: State Bank of India<br><br>
BRANCH CODE : 1111<br><br>
</center>
<form action="./transaction.jsp" method=post ">
<br><center><hr><br><br>
NAME ON CARD : <input type="text" name="cardname"><br><br>
CARD NUMBER: <input type="text" name="cardname" required><br><br>
EXPIRY DATE: <input type="date" name="exp" required><br><br> CVV:<input type=password name=cvv required><br><br>
<input type="submit" value="PROCEED"></form>
<br><br><br><br><br><hr></center></div>
<h3><i><center>"A credit card is a convenient device that saves you the trouble of counting your change"</center></i></h3>
</body>
```

Transaction.jsp:

```
<center>

<h1>A Multifactor Security Protocol For Wireless Payments</h1>
</head>
<body>
<div>
</center>
<div class="navbar">
 <a href="#home" onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home" onclick="location='login.html' ">Logout</a>
 <a href="#home" onclick="location='about.html' ">About</a>
</div>
<div class="image">
<br><br>
<center>
</center>
<br>
<br>
<img src="transaction.jpeg" width="600" height="350" style="float:left">
<center>Enter the following details to complete your transaction--><br><br>
DATE/TIME:<p id="demo">Current date and time</p>
<script>
var d = new Date();
document.getElementById("demo").innerHTML = d;
</script>
BANK: State Bank of India<br><br>
BRANCH CODE : 1111<br><br>
</center>
<script>
function validateform(){
var amt=parseInt(document.myform.name.value);
if (amt>22000){
 alert("Insufficient balance");
 return false;
}
}
</script>
<form name="myform" action="./trans_details" method=post onsubmit="return validateform()">
<center>
AMOUNT : <input type=number name=amount required><br><br>
&nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp IN FAVOUR OF: <input type="number" name=payeid
required> (Account Number)<br><br>
MESSAGE : <input type=text name=info><br>

<br>
<br>
<input type="submit" value="SUBMIT"></form>
```

```
<br><br><br><br><br><br><br><hr></center></div>
<h3><i><center>"At the end of the day, the goals are simple: safety and security"</center></i></h3>
</body>
```

generatetic.jsp:

```
<center>
<h1>A multifactor security protocol for wireless payments</h1>
<body>
<div>
</center>
<div class="navbar">
  <a href="#home"  onclick="location='loggedin.jsp' ">Home</a>
  <a href="#home"  onclick="location='login.html' ">Logout</a>
</div>
<div class="image">
<h4>General Norms of TIC Codes!</h4><br>
*Issued by the bank or financial institution as per requisite of Client.<br>
*8 bit or 16 bit Pseudo-randomly generated code assigned to the clients.<br>
*Complicated sequences of digits or combinations of binary or alpha numeric characters may be possible.<br>
*if required a special TIC for each transaction is used.<br>
*The secret code is mandatory for the generation and selection of TIC codes..<br><br>
<br><br>
<form action="./gen_tic" method="post">
<center>Enter your secret password
<br><br><input type="password" name="pwd"><br><br>
<input type="submit">
</center>
</form><br>
<br><br><br><br><br><br><br><br><br><br><hr></div>
<center>
<h3><i>Privacy is not an option, and it shouldn't be the price we accept for just getting on the Internet</i></h3>
</center>
</body>
```

Merchant.jsp:

```
<center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
<body>
</center><center>
<br><br><br><br><br><br><br><br><br><br>
<div class="navbar">
<form action="./Merchant_details.jsp" method=post">
<p> Enter Merchant Certificate number</p>
<input type="password" name="pwd"><br><br>
<input type="Submit"  value="Verify">
</form>
</center></div></body>
```

Merchant_details.jsp:

```
<center>
        <h1>A Multifactor Security Protocol For Wireless Payments</h1>
</head><body><div></center>
                <div class="image"><br>
                        <center>
                                <%
                                        int v1 = Integer.parseInt(request.getParameter("pwd"))
                                PreparedStatement stmt = con.prepareStatement("select * from merchant where Cno=?");
                                stmt.setInt(1, v1);
```

48

```jsp
        ResultSet rs = stmt.executeQuery();
        if(v1==1234){
        while (rs.next()) {
                out.println("Merchant Authentication Status: VALID");
        %><br>
        <br>
        <%%><br>
        <br>
        <%
                out.println("MERCHANT PERSONAL DETAILS ARE:");
        %><br>
        <br>
        <%%><br>
        <br>
        <%
                out.println("Merchant Code : " + rs.getInt(5));
        %><br>
        <br>
        <%
                out.println("Merchant Name : " + rs.getString(6));
        %><br>
        <br>
        <%
                out.println("Merchant Address : " + rs.getString(7));
        %><br>
        <br>
        <%%><br>
        <br>
        <%
                out.println("MERCHANT BANK DETAILS ARE:");
        %><br>
        <br>
        <%%><br>
        <br>
        <%
                out.println("Account Number: " + rs.getInt(1));
        %><br>
        <br>
        <%
                out.println("Bank Name : " + rs.getString(2));
        %><br>
        <br>
        <%
                out.println("Bank Code : " + rs.getInt(3));
        %><br>
        <br>
                                <center>
        <button onclick="location='Merchant_NextStep.jsp'">Do You Want To continue</button>
        &nbsp &nbsp
        <button onclick="location='merchant.jsp'">NO</button>
</center><%
                }}
        else{
                out.print("Invalid number");
        }

        %>
</center>
<br><br></center></div><hr><h3>
<i><center>"Security used to be an inconvenience
                sometimes, but now it's a necessity of all time"</center></i></h3>
```

Merchant_NextStep.jsp:

```
<center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
<body>
</center>
<center>
<br><br><br><br><br><br><br><br><br><br>
<div class="navbar">
<center><br><br><br>
<button onclick="location='creditcard.jsp'">Credit Card Transfer</button>
                                    &nbsp &nbsp<br><br><br><br><br><br>
                                    <button onclick="location='transaction.jsp'">Electronic Card Transfer</button>
<br><br><br><br></center>
</center></div></body></html>
```

Otp.jsp:

```
<center>
<h1>A multifactor security protocol for wireless payments</h1>
</head><body><div>
</center><div class="navbar">
 <a href="#home" onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home" onclick="location='login.html' ">Logout</a>
</div><div class="image">
<br><br><center><br>
<img src="otp.png" width="250" height="200">
<br><br>
<form action="posttrans.jsp" method=post>
Enter the OTP sent to your registered mail id:<br><br><input type="password" name="otp" required>
<br><br>
<input type="submit" value="SUBMIT">
</form><br><br><br><br><br><br><hr>
</center></div><center>
<h3><i>"The Only Real Security that a man can have in this World is a Reserve of Knowledge, Experience and
Ability"</i></h3>
</center> </body>
```

Postgentic.jsp:

```
<center>
<h1>A multifactor security protocol for wireless payments</h1>
<body><div></center>
<div class="navbar">
 <a href="#home" onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home" onclick="location='login.html' ">Logout</a>
</div><div class="image">
<center><br><br><hr><br><br>
<h3>Dear Customer you have successfully changed your TIC code !!</h3>
On<br><p id="demo">Current date and time</p>
<script>
var d = new Date();
document.getElementById("demo").innerHTML = d;
</script>
<h3><i>Your New TIC code will be reflected at time of your transaction</i></h3>
</center><hr></div>
<center>
<h3><i>Privacy is not an option, and it shouldn't be the price we accept for just getting on the Internet</i></h3>
</center>
</body>
```

50

Postselecttic.jsp:

```
<center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
</head><body>
<div></center>
<div class="navbar">
 <a href="#home" onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home" onclick="location='login.html' ">Logout</a>
</div>
<div class="image"><br>
<H3>Dear Customer Please Confirm Your Transaction details (Y/N)!!</H3>
<center>
BANK: State Bank of India<br><br>
BRANCH CODE: 1111<br><br>
<%
Statement stmt=con.createStatement();
String amt="";
String info="";
String tic="";
ResultSet rs=stmt.executeQuery("select info,amount from transaction where custid='1340'");
while(rs.next())
{
        info=info+rs.getString(1);
        amt=amt+rs.getString(2);
}
out.println("IN FAVOUR OF: 12345");%><br><br><%
out.println("MESSAGE: "+info+"\n");%><br><br><%
out.println("AMOUNT: "+amt+"\n");%><br><br><%
ResultSet rs2=stmt.executeQuery("select tic from tictable where ROWNUM=1");
while(rs2.next())
{
        tic=tic+rs2.getString(1);
}
out.println("YOUR SELECTED TIC: "+tic+"\n");
%>
<br><br>
<button onclick="location='otp.jsp'">YES</button>&nbsp &nbsp
<button onclick="location='transaction.jsp'">NO</button>
</center><br><br><br><br><br><br><br><br><br><br><br></div><hr>
<h3><i><center>"Security is no replacement for liberty"</center></i></h3>
```

posttrans.jsp:

```
<center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
</head><body><div></center>
<div class="navbar">
 <a href="#home" onclick="location='loggedin.jsp' ">Home</a>
 <a href="#home" onclick="location='login.html' ">Logout</a>
</div><div class="image"><br><br>
<center><br><br>
<b>Transaction Successfull...</b>
<br><br>Thank you!
</center>
<br><br><hr>
</div>
<center>
<h3><i>"The good we secure for ourselves is precarious and uncertain <br><br>until it is secured for all of us and incorporated
into our common life"</i></h3>
</center>
</body>
```

Selecttic.jsp:

```
<center>
<h1>A Multifactor Security Protocol For Wireless Payments</h1>
</head>
<body>
<div>
</center>
<div class="navbar">
  <a href="#home"  onclick="location='loggedin.jsp' ">Home</a>
  <a href="#home"  onclick="location='login.html' ">Logout</a>
</div>
<div class="image"><br><br>
<img src="lock.jpg" width="600" height="350" style="float:left"><br><br>
<center>
Select TIC code from the below!!
</center><br><br>
<form action="./postselecttic.jsp" method=post><center>
<input type=radio  name=tic>mx1znkLoxzIwuyyPD851vEZEyFsV+Y7jiBgO3SLIOT4=<br><br>
<input type=radio  name=tic>egGS20rrzP5Ch/Bn1Y9WBWpjCF6pYEbYOCJYh/0nSWY=<br><br>
<input type=radio  name=tic>tWVwhWZzkN1CbE8ZZWWIxYhCHcylY+s3OeWjgE3jX5E=<br><br>
<input type=radio  name=tic required>H7A9fkG/NW7Uowi/Y32cgK5v+9TpSjTQX0RQvDZRUiA=<br><br>
<input type=radio  name=tic>0d0oefPKYS6xkqHz4aPTpyq8yTzGzvBG3sLRL5m7BUo=<br><br>
<input type=radio  name=tic>NimTDe4hNzjbXzxNzWZidl1VYRa1yJwOf4cTP/UsQrQ=<br><br>
<input type="submit" value="submit">
<br><br><br><br><br><br><br><br><br><br><br><br><br><hr>
</center></form></div></body>
<h3><i><center>"Security used to be an inconvenience sometimes, but now it's a necessity all the time"</center></i></h3>
</html>
```

Validate_trans.jsp:

```
<center>

<h1>Demo of MFA</h1></head><body>

<%
Statement stmt=con.createStatement();
String v1= request.getParameter("amount");
String v2=request.getParameter("uid");
String amt="";
ResultSet rs=stmt.executeQuery("select amount from cust where username='"+v2+"' and pwd='1234'");
while(rs.next())
{
        amt=rs.getString(1);

}
if(Integer.parseInt(amt)>=Integer.parseInt(v1))
 %> <jsp:forward page="aes.java" />
</body>
```

Client_db.java:

```
package ep;

import java.util.Random;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;
```

```java
import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


/**
 * Servlet implementation class client_db
 */
public class client_db extends HttpServlet {
private static final long serialVersionUID = 1L;
Random rand = new Random();


   // Generate random integers in range 0 to 999

   int rand_int1 = rand.nextInt(1000);
   /**
    * @see HttpServlet#HttpServlet()
    */
   public client_db() {
      super();
      // TODO Auto-generated constructor stub
   }


/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        String v0 =request.getParameter("cid");
        String v1=request.getParameter("name");
```

```java
String v2=request.getParameter("id");

String v3=request.getParameter("pwd");

String v4=request.getParameter("add");

String v5=request.getParameter("ph");

String v6=request.getParameter("amt");

String v7=request.getParameter("email");

String v8=request.getParameter("male");

String v9=request.getParameter("female");

String v="";

if(v8=="on")

{

        v="male";

}

if(v9=="on")

{

        v="female";

}

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");

System.out.println("connection successful");

PreparedStatement pstmt=con.prepareStatement("insert into cust values(?,?,?,?,?,?,?,?,?)");

        pstmt.setString(1, v0);

pstmt.setString(2, v1);

pstmt.setString(3, v2);

pstmt.setString(4, v3);

pstmt.setString(5, v4);

pstmt.setString(6, v5);

pstmt.setString(7, v6);

pstmt.setString(8, v7);

pstmt.setString(9, v);

pstmt.executeUpdate();

out.println("<h3>Your Data Saved successful- Registration completed</h3>");
```

```java
                    response.sendRedirect("home.html");

                }

        catch(SQLException e)

            {

                        System.out.println(e);

            } catch (ClassNotFoundException e) {

                        // TODO Auto-generated catch block

                        e.printStackTrace();

            }

    }
/** @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response) */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        doGet(request, response);

}
}
```

Client_login.java:

```java
package ep;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/**

 * Servlet implementation class client_login

 */

public class client_login extends HttpServlet {

private static final long serialVersionUID = 1L;

/**

    * @see HttpServlet#HttpServlet()
```

55

```java
     */
  public client_login() {
    super();

  }
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out=response.getWriter();
        String v1=request.getParameter("id");
        String v2=request.getParameter("pwd");
        try
        {
        Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");
    Statement stmt=con.createStatement();
    String s="select custname from cust where custusername='"+v1+"' and pwd='"+v2+"'";
    int n = stmt.executeUpdate(s);
    if(n>0)
    {
        request.getRequestDispatcher("loggedin.jsp").forward(request, response);
    }
    else
    {
        out.println("Invalid Login");
        response.sendRedirect("login.html");
    }
        }
    catch(Exception e)
        {
                System.out.println(e);
        } }
```

```java
/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
}}
```

Gen_tic.java:

```java
package ep;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class gen_tic
 */
public class gen_tic extends HttpServlet {
private static final long serialVersionUID = 1L;
  /**
    * @see HttpServlet#HttpServlet()
    */
   public gen_tic() {
     super();
     // TODO Auto-generated constructor stub
   }
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
```

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    PrintWriter out=response.getWriter();
    String v1=request.getParameter("pwd");
    try
    {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");
    Statement stmt=con.createStatement();
    String s="delete from tictable where ROWNUM=1";
    int n = stmt.executeUpdate(s);
    if(n>0)
    {
        request.getRequestDispatcher("postgentic.jsp").forward(request, response);
    }
    else
    {
        out.println("<h3>Invalid Credentials/TIC Generation Error!!</h3>");
        response.sendRedirect("generatetic.jsp");
    }
        }
    catch(Exception e)
        {
                System.out.println(e);
        }       }
/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
}
```

Test.java:

```java
package ep;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class test
 */
public class test extends HttpServlet {
private static final long serialVersionUID = 1L;
  /**
     * @see HttpServlet#HttpServlet()
     */
    public test() {
        super();
        // TODO Auto-generated constructor stub
    }
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        String v1=request.getParameter("id");
        String v2=request.getParameter("pwd");
```

```java
        try
        {
        Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");
    Statement stmt=con.createStatement();
    String s="select custname from cust where username='"+v1+"' and pwd='"+v2+"'";
    int n = stmt.executeUpdate(s);
    if(n>0)
    {
        System.out.println("found");
    }
    else
    {
        System.out.println("Invalid Login");
    }
        }
    catch(SQLException e)
        {
                System.out.println(e);
        } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

}       /**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
}


}
```

Trans_Details.java:

```java
package ep;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.Statement;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class trans_details
 */
public class trans_details extends HttpServlet {
private static final long serialVersionUID = 1L;
    /**
     * @see HttpServlet#HttpServlet()
     */
    public trans_details() {
        super();
        // TODO Auto-generated constructor stub
    }
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        String v0 =request.getParameter("amount");
        String v1=request.getParameter("info");
```

```java
        try
        {
        Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");
    System.out.println("connection successful");
    Statement stmt=con.createStatement();
    String s="update transaction set info='"+v1+"',amount='"+v0+"' where custid='1340'";
    int n = stmt.executeUpdate(s);
    if(n>0)
    {
        request.getRequestDispatcher("selecttic.jsp").forward(request, response);
    }
    else
    {
        out.println("Error in performing Transaction");
        response.sendRedirect("transaction.jsp");
    }
        }
    catch(Exception e)
        {
                System.out.println(e);
        }
}
/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
}


}
```

Validate_tic.java:

```java
package ep;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class validate_tic
 */
public class validate_tic extends HttpServlet {
private static final long serialVersionUID = 1L;
 /**
   * @see HttpServlet#HttpServlet()
   */
   public validate_tic() {
     super();
     // TODO Auto-generated constructor stub
   }
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out=response.getWriter();
        String v1=request.getParameter("pwd");
        try
        {
        Class.forName("oracle.jdbc.driver.OracleDriver");
```

```java
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system","manager");

        Statement stmt=con.createStatement();

        String s="";

        int n = stmt.executeUpdate(s);

        if(n>0)

        {

            request.getRequestDispatcher("otp.jsp").forward(request, response);

        }

        else

        {

            out.println("<h1>Invalid Credentials</h1>");

            response.sendRedirect("selecttic.jsp");

        }

            }

        catch(Exception e)

            {

                    System.out.println(e);

            }

}


/**

 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

 */

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        // TODO Auto-generated method stub

        doGet(request, response);

}

}
```

64

Aes.java:

```java
package ep;

import java.io.UnsupportedEncodingException;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.util.Arrays;

import java.util.Base64;

 import javax.crypto.Cipher;

import javax.crypto.spec.SecretKeySpec;


public class AES {


    private static SecretKeySpec secretKey;

    private static byte[] key;


    public static void setKey(String myKey)

    {

        MessageDigest sha = null;

        try {

            key = myKey.getBytes("UTF-8");

            sha = MessageDigest.getInstance("SHA-1");

            key = sha.digest(key);

            key = Arrays.copyOf(key, 16);

            secretKey = new SecretKeySpec(key, "AES");

        }

        catch (NoSuchAlgorithmException e) {

            e.printStackTrace();

        }

        catch (UnsupportedEncodingException e) {

            e.printStackTrace();

        }

    }
```

```java
    public static String encrypt(String strToEncrypt, String secret)
    {
      try
      {
        setKey(secret);
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));
      }
      catch (Exception e)
      {
        System.out.println("Error while encrypting: " + e.toString());
      }
      return null;
    }


    public static String decrypt(String strToDecrypt, String secret)
    {
      try
      {
        setKey(secret);
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
      }
      catch (Exception e)
      {
        System.out.println("Error while decrypting: " + e.toString());
      }
      return null;
    }
}
```

Data Base Tables:

► create table cust (

  custid varchar2(25) NOT NULL ,

  custname varchar2(25) NOT NULL,

  custusername varchar2(25) NOT NULL, password varchar2(20),address varchar2(40),

  phone varchar2(12), amount varchar2(25), email varchar2(25), gender varchar2(7),

  PRIMARY KEY (custid)

);

► create table transaction(

custid varchar2(25) , transid varchar2(25) , info varchar2(40),amount varchar2(25),datetime varchar2(40),

 PRIMARY KEY (transid),

  FOREIGN KEY (custid) REFERENCES cust(custid)

);

► create table tictable(

transid varchar2(25) , tic varchar2(20),

  FOREIGN KEY (transid) REFERENCES transaction(transid)

);

► create table entictable(

transid varchar2(25) , encrytic varchar2(50)

);

► Create table merchant(M_ANO Number(10), M_BankName Varchar2(20), M_BankCode number(10), CNo number(10), M_ID number(10), M_Name varchar2(20), M_Address varchar(20));

CHAPTER 6 : EXPERIMENTAL RESULTS

Home Page:

Registration Page:

Login Form: (1FA):

Services available after logged in:

Balance Enquiry:

A MULTIFACTOR SECURITY PROTOCOL FOR WIRELESS PAYMENTS

Home   Logout   About

Dear Customer your balance details are!

DATE/TIME:

Fri Dec 11 2020 23:50:47 GMT+0530 (India Standard Time)

BANK

State Bank of India

BRANCH CODE

1111

Your current balance is 18982/- only..

"We have to Feel Money is Nothing , But Again Money Eveything!"

Generation of TIC:

A MULTIFACTOR SECURITY PROTOCOL FOR WIRELESS PAYMENTS

Home   Logout

General Norms of TIC Codes!

*Issued by the bank or financial institution as per requisite of Client.
*8 bit or 16 bit Pseudo-randomly generated code assigned to the clients.
*Complicated sequences of digits or combinations of binary or alpha numeric characters may be possible.
*if required a special TIC for each transaction is used.
*The secret code is mandatory for the generation and selection of TIC codes..

Enter your secret password

[••••]

Submit

Privacy is not an option, and it shouldn't be the price we accept for just getting on the Internet

Credit Card Transaction:

After the details of card we will proceed with transaction details followed by selection of TIC Codes(2FA)

A MULTIFACTOR SECURITY PROTOCOL FOR WIRELESS PAYMENTS

Home   Logout

Select TIC code from the below!!

- H7A9fkG/NW7Uowi/Y32cgK5v+9TpSjTQX0RQvDZRUiA=
- mx1znkLoxzIwuyyPD851vEZEyFsV+Y7jiBgO3SLIOT4=
- 0d0oefPKYS6xkqHz4aPTpyq8yTzGzvBG3sLRL5m7BUo=
- NimTDe4hNzjbXzxNzWZidl1VYRa1yJwOf4cTP/UsQrQ=

submit

"Security used to be an inconvenience sometimes, but now it's a necessity all the time"

After Selection and validation of the TIC CODE sms/email verification takes place as a final factor of authentication i.e here we will receive the otp to the registered mail id of the client at the time of registration which will validated at the otp.jsp page

A MULTIFACTOR SECURITY PROTOCOL FOR WIRELESS PAYMENTS

Home   Logout

Enter the OTP sent to your registered mail id:

SUBMIT

"The Only Real Security that a man can have in this World is a Reserve of Knowledge, Experience and Ability"

User Has to Enter the TIC in the Form :

The sample of OTP mail will be as follows,
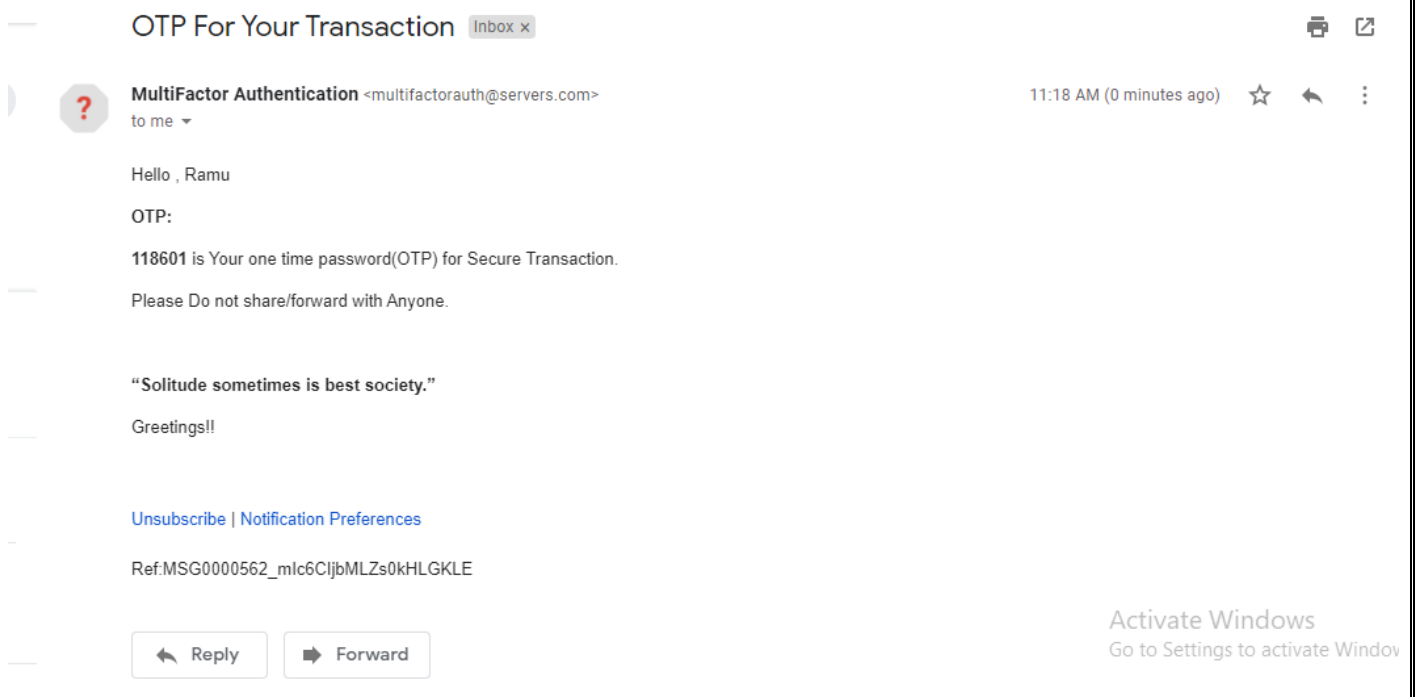
Post the Validation of transaction the Acknowledgement is sent to the client and Displays the status of transaction on the posttrans.jsp page

CHAPTER 7 : CONCLUSION AND FUTURE-SCOPE

Security is a significant issue in web based online framework. There are different web dangers which influence the security arrangement of web and increment hazard for electronic exchange. The current verification strategy for online framework is not exceptionally secure to shield client from wholesale fraud, as a the outcome any aggressor pick up the access on classified data of client like charge card number or record secret phrase also, make illicit exchange of asset which will be charged to the substantial client's record. It is demonstrated from our experience study that solitary factor confirmation expands hazards presented by phishing, recognize burglary, online extortion and loss of client secret data. Thus, monetary foundation should execute a viable confirmation to lessen extortion and make more grounded security for applications. Solid client validation is important to authorize security and help monetary organizations to distinguish and diminish client character robberies.

The current validation protocol for online framework implementation is not to ensure that customers are safe from wholesale fraud, as the effect is that any aggressor gains entry into confidential customer data such as Master-card number or record secret key and allows illegal shop trade that will be charged to the record of the significant customer For remote installation of the system, we zeroed in on an application-layer security response to conduct a start to finish verification and privacy of information between remote client and java-based protected worker. The introduced work suggested another web client validation convention focused on a multifaceted verification method that is fully safe and easy to execute.

We based on an application-layer protection solution for the presented protocol for Regulation of an end-to-end authentication and data wireless payment system Wireless device confidentiality with java-based secure servers. The work presented is a new web user authentication protocol based on multifactor authentication has been proposed. An approach that is entirely safe and easy to enforce.

One Time password for every payment throughout the life cycle of the transaction/payment is important and need to be secure. ServiceNow provides highly resilient and secure cloud-based services to customers all over the world. The security of the infrastructure and data is paramount - a foundational requirement. The Generation of one time password for the secure transactions can also be implemented in servicenow. Servicenow will have a greater impact on further upcoming projects.

Future work will concentrate on creating a modern and reliable way to produce TIC codes. At the institutions of finance. Installing the TIC code on the mobile phone of the user must also be The simple job is to discourage frequent customer visits to the bank or financial institution. Maintenance, control process and delivery on the server side TIC to fulfill the Demand from a large number of users is also part of potential job prospects.

CHAPTER 8 : REFERENCES

[1] Article on security threats of mobile phones :http://news.zdnet.com/2100-1009_22-5602919.html.

[2] Adi W, Mabrouk A., Al-Qayedi A., Zahro A. , "Combined Web/Mobile Authentication for Secure Web Access Control", In the proceedings of the IEEE conference held on communications and Networking , pp. 677- 681, March 2004.

[3] Dr. Winston Seah, Santhosh Pilakkat, Jaya Shankar P., Seng Kee Tan, Chin Siang Kee, Anindita Guha Roy, Edwin Ng., "The Future Mobile Payments Infrastructure A Common Platform for Secure M-Payments", A Joint Study by Institute for Communications Research and Systems, December 2001

[4] Asoke Nath, Tanushree Mondal , "Issues and Challenges in Two Factor Authentication Algorithms ", In Proceedings of the International Journal of Latest Trends in Engineering and Technology , January 2016

[5] Adi W., Mabrouk A., Al-Qayedi A., Zahro A. , "Combined Web/Mobile Authentication for Secure Web Access Control", In Wireless communications and Networking conference,IEEE Communications Society, Atlanta, GA USA,volume 2, pp. 677- 681, March 2004

[6] Qiong Pu, "An Improved Two-factor Authentication Protocol", In Proceedings of 2010 Second International Conference on Multimedia and Information Technology, 24-25 April 2010.

[7] Teppo Halonen, "A System for Secure Mobile Payment Transactions", Supervisor: Professor Teemupekka Virtanen, Helsinki University of Technology, Department of Computer Science and Engineering, January 2002.

[8] Ayu Tiwari, Sudip Sanyal, Ajith Abraham, Sugata Sanyal & Svein Knapskog, "A Multi-factor Security Protocol for the Wireless Payment-Secure Web Authentication using Mobile Devices", held on February 2007,IADIS International Conference, Applied Computing 2007, Salamanca, Spain, pp.

[9] Sameer Saxena1 , Sonali Vyas2 , B. Suresh Kumar3 , Shaurya Gupta, "Survey on Online Electronic Paymentss Security" In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), 4-6 Feb. 2019

[10] J. Gao, J. Cai, K. Patel, and S. Shim , "Wireless Payment", In Proceedings of the Second International Conference on Embedded Software and Systems (ICESS'05),Xian, China, pp

[11] https://docs.servicenow.com/

[12] https://www.servicenow.com/

[13] https://docs.servicenow.com/bundle/paris-platform-administration/page/administer/platform-events/concept/c_EventRegistry.html

[14] https://docs.servicenow.com/bundle/paris-servicenow-platform/page/administer/notification/task/t_CreateANotification.html

[15] https://www.servicenowelite.com/blog/2016/6/11/learn-business-rules

[16]   https://www.cmu.edu/iso/news/mfa-
        article.html#:~:text=Multi%2Dfactor%20authentication%20should%20be,protect%20highly%20sensitiv
        e%20personal%20information

[17]   https://ieeexplore.ieee.org/document/8701960

[18]   https://ieeexplore.ieee.org/document/8356384

[19]   https://ieeexplore.ieee.org/document/5069395

[20]   https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8701353

[21]   https://patentimages.storage.googleapis.com/72/48/2b/771c690320fa7b/US20050143051A1.pdf

[22]   https://www.tifr.res.in/~sanyal/papers/Multifactor%20Secure%20Authentication.pdf