International Conference on Applied Soft Computing Techniques ICASCT-2018

RESEARCH ARTICLE OPEN ACCESS

A Novel Method for Securing and Safe Transfer of One Time PasswordInNetBanking.

SrivastavBudugutta*, SaiBrahmaNikhileshVutukuri**,RamaPrabha KP***
*(MTech (S.E), VIT University, Vellore.
Email: srivastavbudugutta95@gmail.com)
** (MTech (S.E), VIT University, Vellore.
Email :Vutukurisainikhil@gmail.com)
*** (MTech (S.E), VIT University, Vellore.)

ABSTRACT:

OTP is a one-time password which is used as a security step in order to authenticate users. Normally the OTP is valid for a particular transaction. So the time the OTP generated and the time it is used is important. Since the time gap between the times the OTP received by the user and the OTP entered by the user can be comprisable. Most off the time the OTPs are send through SMS. The OTPs are usually time based in the sense that they are valid for a particular period of time and some are session based in the sense that the OTP is valid until the OTP has been entered into the system by the user. OTP has become a very common approach for authenticating the user, and most of the OTP are SMS based type and when it comes to Net banking OTP generation, either for online transaction or paying bill is a challenging task while using SMS based OTP. Nowadays many mobile banking apps are being used and these apps are used for mobile banking system, so using this mobile app as a base for the OTP generation we can curb the difficulties or challenges faced by SMS based system and secure OTP generating system. The following paper gives a novel approach of generating OTP and securely transferring of OTP to the users in order to complete the online transaction in order to curb the challenges that are faced by the Net banking systems for the OTP generation.

Keywords—Net Banking, OTP, User Authentication, Security, IMEI number

I. INTRODUCTION

Nowadays internet has become a part of our life. With the help of internet, we are able to connect systems and systems of systems to ease our work and saves our time. Since internet is open to everyone so it is vulnerable to virus attacks, intrusion and other active attacks.

With increase in the internet usage there is an increase in concern of providing security for the information that is available online. Even though there are many security measures applied due to increase in technology it has become very easy for the attackers or hackers to steal the sensitive information. To steal the sensitive information there has been many online attacks and some indirect way of stealing information is through covert channel, and some other means of stealing the sensitive information is through phishing.

The sensitive information includes user banking details like credit card and account details. So stealing of these information can lead to violation of the privacy data and the sensitive data even can be misused. In order to curb these type of attacks there has been a lot of development in the field of internet security especially in providing security to online transactions. When it comes to internet banking the main thing before starting a transaction is to first authenticate the user. The form of authentication is done with the user's card number and with the personal pin that is provided to the user by the bank.

Back to the beginning of the security system for making a transaction the banks started issuing a token to the user in order to make transaction complete. The token numbers that are issued to the users were unique that is no two persons have same token number. This type of token system has been widely used by the banks.

The token system was famous for a bit of time. The uniqueness of the token number acts as a pin for making the transaction. But the main problem with this type of system is that maintaining the records of the token numbers and verifying these token number was a burden not only that if the token number was lost by the user creating a new token number was a burden since the token number should be unique and the newly generated token number has to be updated into the records. So in order to curb these challenges there has been may proposed methods which will be discussed in the following session. Out of these methods OTP is one of the method which acts as a security verification step in order to authenticate the authorised user. The main advantage of the OTP compared to the other methods is that the OTP sent by the server is valid for a particular period of time and gets invalidated once it has been used that is it gets discarded once the user has used the OTP for making the transaction. Even though the OTP system authenticates the user there has been cases where the OTP gets lost due to network failure since most of the OTPs are SMS based and it is necessary the OTP need to be sent to the user in a specified period of time so that the session of the transaction wont gets expired. So it necessary to consider how strong is the OTP that is the method used for OTP generation is important, and it is also important how the OTP reaches the user.

This paper proposes a strong OTP generation technique with is a combination of what we have (card number) what we know(password, IMEI number) and what we are (finger print). So the proposed paper shows how the OTP generation method is strong and efficient form others and the way of reaching the user.

II. BACKGROUND WORKS

There have been many works proposed on how OTP can be generated and there on the way how OTP reaches people. But the methods that are proposed have both advantages as well as disadvantages.

When we go to the beginning of the development of authentication system there has been many ways the user gets authenticated which include with the help of the token number, using the personal card as well as using a personal pin. So this acts as an authentication step in order know whether the person is as he claims to be.

Some of the commonly known network attacks include eavesdropping, data modification, identity spoofing, password-based attacks, man-in-the-middle attack, and compromised-key attack sniffer attack. Eavesdropping, in most of the cases the network communications occur in an unsecure format, which attract the attackers who has gained access to the network line or the data line to listen in or read the traffic. This method of monitoring the network is also known as sniffing or snooping. Another network attack includes data modification, in this type of attack the attacker reads the data and then modify data without acknowledge of the sender or the receiver. Identity spoofing is method where the network and the operating system use the IP address of a computer to identify a valid entry, in this case it possible for an IP address to be falsely assumed-identity spoofing. The attacker uses a special program to construct IP packets that appears to originate from valid IP address inside the corporate intranet. So the attacker after gaining access to the network with a valid IP address, the attacker can then modify, enroot, or even delete the data. Password-based attack is another type of network attack where the attacker gets the user name and the password through eavesdropping, since most of the old applications do not always protect information that is passed through the network for validation. So after gaining the access to the network with a valid account the attacker then can obtain the list of valid users and network information etc.. The man-in-the-middle is an attack where someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. In the sniffer attack a sniffer application or device is used to read, monitor, and capture network data exchanges and read network packets. Key logging is a method stealing information where every key pressed by the user will be recorded, so by this way he can gain the sensitive information like passwords etc. There havebeen many proposed methods in order to curb these attacks. The following methods that were proposed by the experts which will give information regarding the way to protect OTP from being misused by the intruder.

In the case of OTP generation there have been many proposed ways which include using homographic graph by changed location and angle of figure print feature and so on and for the ways that the OTP reaches user there have been methods includes representing OTP in QR format and sending OTP to mobile with the help of short message service (SMS).

The OTP method have proven a great advantageous over the authentication process. But there have been many attacks. So it is necessary to make the OTP generation process stronger and send the OTP to the mobile in most encrypted form so that it makes harder to decode by the attackers.

When we look into the development of the methods of sending OTP, they are sent to the authorised user by means of SMS, the OTP is send in QR format [1] so that it is well encrypted so that it will be impossible to break the code easily. In this method

At first the OTP is generated in two methods

1. Time based OTP-the OTP changes at frequent intervals
2. Event-based OTP- the OTP gets generated by pressing a button on the OTP device or token

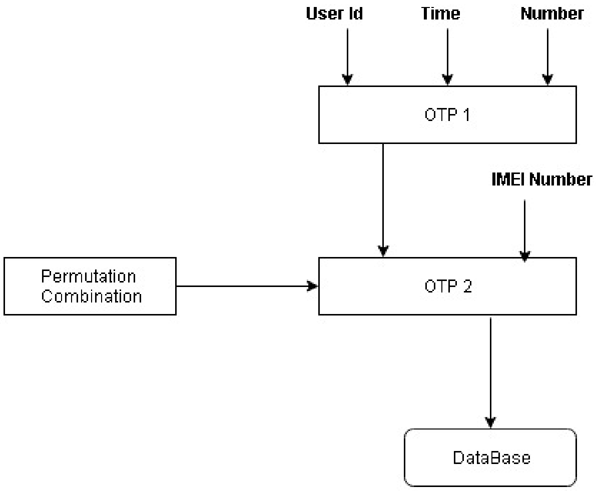


Fig.1 OTP Generation

By using the combination of the 3 parameters which include customer id, current system time and random number OTP1 is generated. That is hidden inside the QR code image.

In this method Permutation combination logic is applied on the two parameters- OTP1+IMEI number and from that OTP2 is generated. So OTP1 is of 4-digit number which is embedded in QR code and OTP2 is 8-digit number that is calculated from OTP1.

In this method the main drawback is that the OTP gets encrypted in QR format which is an image format and reading of the image require a separate device or an app in the mobile in order to read the QR coded image. In the case of the QR code encrypting of the OTP, even though the OTP gets encrypted in the QR format the main drawback is that the OTP is in an image format which require an OTP scanner.

When we discuss about the methods of the sending of OTP to the authorised user, when we look into the SMS based OTP there are far more disadvantages than advantages, for example if a user want to start a Online transaction to send or pay a bill the user is first authenticated using the card that he is having and pin which he enters in the banking website so that the transaction starts after the transaction process begins the bank server than sends a SMS based OTP to the user to ensure that if he is authorised user or not. This OTP process ensures that even though in case if the users card number as well as users pin number are stealing by some other means this OTP method send a onetime password to the user's mobile number so that he gets verified as a legitimate user and complete the online transaction. This OTP is session based.

Another method includes installing a java application on the user smart phone or mobile phone which acts as token generator that is OTP generator which was proposed by indu[5]. In this method a J2ME program is developed and installed on the mobile phone to generate the OTP. The program has an easy to use GUI that is developed using the NetBeans drag and drop interface. The OTP program can run only on J2ME enabled mobiles. The OTP program has two options of (1) generating the OTP locally using the mobile credentials, and other way is by (2) requesting the OTP frim the server via an SMS message. The overall design of this method is described in the Figure (2)

In this model whenever the user starts a transaction the OTP is generated on the application, but the disadvantage is that the user need to install this application for only token generation but cannot be used for any mobile banking and other application etc. And furthermore when we take security into consideration the paper proposed the generation of OTP on client side using an algorithm, and same type of algorithm is also used on the server side and the mechanism used to generate the OTP is based on the PIN and username so it will be easy for the attackers to steal the information easily and can even can know the type of algorithm the mobile app use to generate the OTP.So this proves the inefficiency of the system and lack security.

TABLE 1
OTP METHODS COMPARISION

Method	Advantage	Disadvantage
SMS based OTP	Cheapest and simplest transferring of OTP. Whenever requested, the SMS is directly sent from the server to the authorised users mobile	May gets delayed due to network traffic. Sometimes the SMS won't be reached to the user on time due to the network traffic. This reduces the effectiveness of the system.
QR based OTP	In this method the OTP is encrypted in the QR format which is not directly readable by the user.	Requires an additional device or pp in order to scan or read the QR code.
App based OTP generation	It is a very efficient way of generating OTP. Has high reliability.	Need a separate app just only to generate the OTP and the app cannot be used for any other purpose like mobile banking etc.

So by this we can say that the above proposed methods all have both advantages as well as disadvantages, but in the below proposing method proves that the it has more advantages than previous proposed models.

The proposed paper elaborates on the way the OTP gets generated and the way of transferring the OTP to the user or customer in more efficient rather than conventional SMS way. In the following paper we will be discussing about Advance Encryption Standard (AES) 256 which was proven to be efficient way of encrypting and generating OTP and sending of OTP with the help of mobile banking app. A combination of this will eventually improve the efficiency and security of the OTP generation. It is necessary to note that some of the payment are and must to be made through net banking. So the OTP generation have crucial part in making the transaction successful. Since we have discussed about the challenges that are faced by the OTP generation a detailed description on how these challenges were overcome will be discussed in the following paper.

III. RESEARCH

Due to the increase in the mobile usage by the users, usage of mobiles has made the work simpler. Most of the online payment or transaction are done through mobile banking by using a mobile banking apps or using other apps which integrates with the banks servers and online banking which is done through browser. The main difference between mobile banking and Net banking is that, the mobile banking can be done with the help of the app provided by the authorised bank like IndPay apps etc., but nowadays many online payment apps include Phone Pay app, Paytm app are being used to pay or transfer money from user to another user or organisation.

The apps other than the one that are provided by the authorised banks works by integrating the transaction with the authorised banks. The main difference of the apps provided by the bank and the apps that support online transaction system is that, the apps provided by the banks support any online transaction to be done using that particular users banks account while other apps first integrates with the all banks servers to which user have registered and then allow the transaction to be done using any banks account. When we talk about internet banking, it works with the help of browser. That is, to start a transaction using internet banking it need the assistance of a browser to start a transaction. So nowadays many online transactions are done using mobile apps like Paytm and other apps to save time and ease our work. So the following paper tell how these apps are used to generate OTP whenever a transaction is started by the user while using Internet banking. And also the way the OTP gets generated based on what we know (credit card details), what we have (PIN or IMEI) and what we are (bio metric) so based on that we will be generating a 256-bit number and using Advance Encryption Standard we will be encrypting the OTP and sent it to the user using the mobile banking app or any other banking app that are registered using user mobile number.

The solution to these challenges include generating of OTP on server side and sent it the authorised user mobile app on to which he is currently logged on and the OTP gets encrypted using AES (256) algorithm which was proven to be strong and have less avalanche affect, the results were discussed in the paper A AbhishekGandhi [2]. So using these a strong OTP gets generated, gets encrypted and is sent to the user mobile app in a well encrypted manner so that it will be immune to the networks attacks like masquerading (an entity pretend to be another entity), IP spoofing etc. and other network attacks which was discussed above.

The figure 3 gives an overview of the entire process that will be happening whenever the user requested an OTP. The verification process includes, using the OTP which was received by the bank server the OTP is used as a key to regenerate the plain text by the process of reverse engineering. The reverse engineering process, the OTP generation, keys generation and transferring of OTP will be discussed in the further section of the paper.

In the following session of the paper we will be discussion on how the OTP gets generated and how the OTP will be transferred to the user in an Encrypted format. we further discuss about the how the bank server will know that the notification reached the authorised person. After the OTP is reached the authorised user as a notification in the mobile banking app onto which he is currently logged on we discuss on how OTP gets decrypted on the server side after user have typed the OTP in the necessary fields in the browser. Section A highlight the way on how OTP gets generated which include just overview of how the transfer of OTP takes place and overview of how encryption and decryption takes, it even highlights the entire procedure of how the entire process takes place, Section B discuss about on how gets encrypted, Section C give a very detailed process of how the OTP will be sent to the user and Section 4 gives the method on how the OTP gets decrypted on the server side.

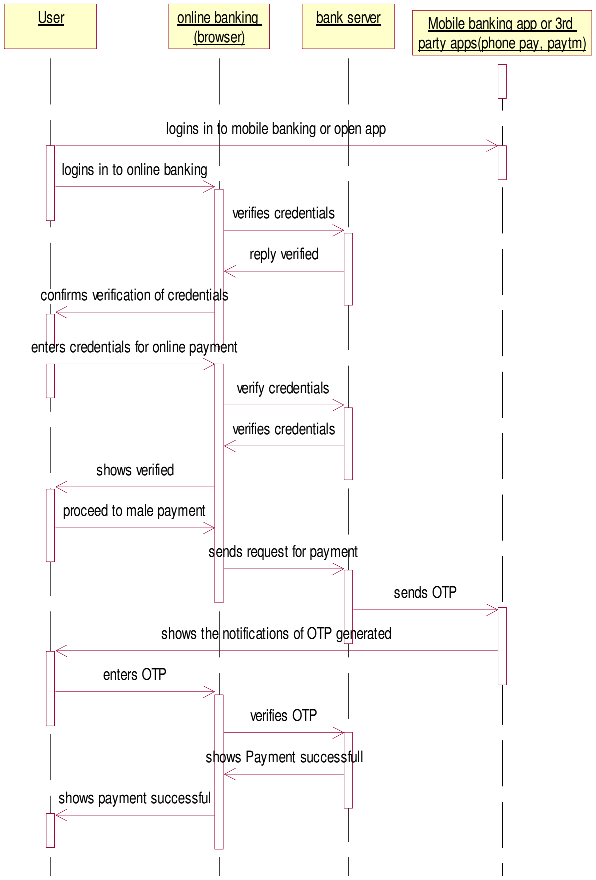

User
online banking (browser)
bank server
Mobile banking app or 3rd party apps(phone pay, paytm)
logins in to mobile banking or open app
logins in to online banking
verifies credentials
reply verified
confirms verification of credentials
enters credentials for online payment
verify credentials
verifies credentials
shows verified
proceed to male payment
sends request for payment
sends OTP
shows the notifications of OTP generated
enters OTP
verifies OTP
shows Payment successfull
shows payment successful

Fig. 3Process Overview.

A. *Generating of OTP*

Normally OTP generation can be done by the following methods: the first type includes generating a new OTP by using mathematical algorithms based on previous, second type is based on time synchronization between authentication server and the client providing password, third way is by again using mathematical algorithm. Now the method proposed by ByungRaeCha [1] where OTP is generated by using changed location and angle of figure print as we have discussed already in the above session. But in the following proposed method we not only use the finger print for OTP generation but also use the combination of our card number, IMEI number and what we are that is using finger print. The IMEI number of the authorised user mobile will be used to verify the users mobile to which OTP will be sent as a notification in the mobile banking app. This mechanism is used to ensure that the OTP will be reached to the authorised person only. In case if the IMEI number of the user mobile doesn't match with the IMEI number which was stored in the banks server then it gives a notification, on the mobile banking app to which he is currently logged on, to the user that the mobile which was being used is not authorised users mobile. Hence by this way the transaction ends and the OTP won't be sent to the user and henceforth the transaction gets automatically gets cancelled and the user have to again start the transaction from the beginning. This procedure will ensure that only the authorised users can make transaction and no one can misuse the mobile banking app for the transaction (like masquerading). Figure 3 will give an idea on how the IMEI number will be verified and how this IMEI number plays a crucial step in authentication process.

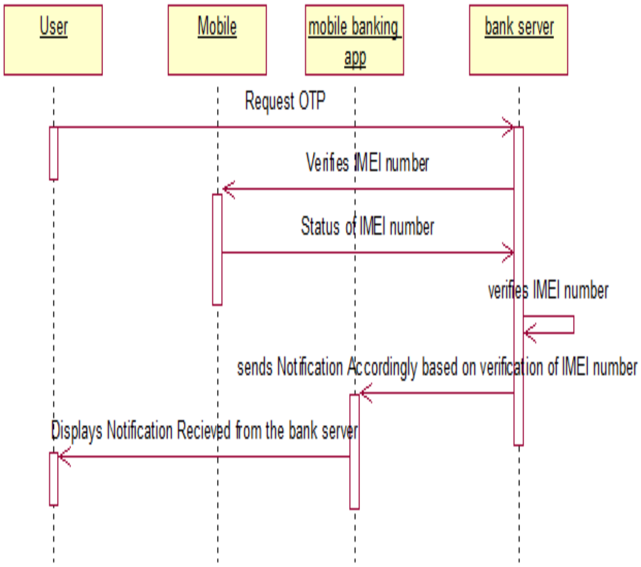

User
Mobile
mobile banking app
bank server
Request OTP
Verifies IMEI number
Status of IMEI number
verifies IMEI number
sends Notification Accordingly based on verification of IMEI number
Displays Notification Recieved from the bank server

Fig. 4 IMEI NumberVerification Process

The figure 4 gives a brief description on how IMEI number gets verified and how the bank server sends a notification to the mobile banking app to which the user is currently logged on. In the above picture at first the user requests an OTP to the bank server in order to make a transaction. The bank server then verifies the user IMEI number with the IMEI number of the authorised user mobile IMEI number and sends the notification accordingly to the mobile banking app on to which the user is currently logged on. So by this verification method even though the unauthorised person knows the authorised person user id and password of the mobile banking app and uses in his own mobile to get OTP, then the bank server sends a notification to the unauthorised users mobile banking app that please use the authorised mobile in order to complete the transaction. So by this way the first step of verifying the user mobile gets complete. Now in the further section of the paper we will be discussing on how the OTP gets generated.

Card number = CDi

Finger print or bio metric = BMi

Now using these number, we will be generating a plain text so that it gets encrypted using AES algorithm and whenever the user types the AES key or OTP in the specified field in order to complete the transaction the OTP will be sent to the bank server. And at the bank server the OTP gets decrypted and the plain text gets generated. So by this means the generation of OTP, encrypting of OTP and Decryption of OTP takes place.

The main purpose of using the AES algorithm is that it was proven to be best as depicted in the paper proposed by Abhishek Gandhi [2]. The results from the paper proves that even though it has avalanche effect, but it still proves it is the best when compared to other algorithms because of it simplicity of the key and using of the XOR operation between the plain text and the keys which is a much faster way for the operation to take place.

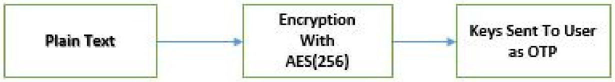


Fig. 5 OTP Sending Process

In the figure 5 we can see that the plaint text gets generated by using the CDi and BMi in a random fashioned manner. After the plain text gets generated the second step includes encrypting of OTP using the AES (256) algorithm and accordingly the keys gets generated. The one half of the keys will be sent to the user as a notification to the mobile banking app and with the help of this app the user get to know the OTP so that the user enters the OTP in the specified fields in order to complete the online transaction. Here the OTP entered by the user is the half portion of the key that was sent by the bank server and other half of the key gets recedes in the server.

Figure 5 will give the overview of how the plain text will be generated back based on the OTP entered by the user

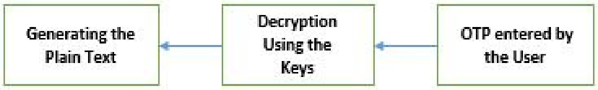


Fig. 6 Decryption Process

From the figure 6 we can see that after the user have entered the OTP that was received by the user through the mobile banking app, the OTP which is half of the generated keys will be sent back to the server and using the keys that was already there in the server and the keys the keys that was sent by the user which was OTP will get concatenated and the keys will be formed.

The keys that was generated after concatenation will be used to regenerate the plain text using the reverse engineering process. The reverse engineering process includes just the reverse process of how the encryption took place that is starting from the last step and going back to the first step. So this is how the OTP encryption, decryption takes place.

Now as we discussed above this section give a detailed procedure of how the OTP gets generated. The card number CDi and the Bio metric BMiwill be first used to generate the plain text. The process of generating the plain text include combining of the BMi number along with the CDi number so that the plain text gets generated. By using these number, a 16 bytes' number gets generated. Every time when ever user requests OTP a particular sequence of randomising the number will be used.

B. Encrypting of the Text

This section contains the brief description on how the plain text gets encrypted using the AES 256 algorithm. The 256 AES algorithm is used to encrypt the plain text in order to generate the OTP (keys) since it is proven to best by paper proposed by Abhishek Gandhi [2]. AES stands for American Encryption Standards.

There are three versions of AES includes:

TABLE 2
VERSIONS OF AES

Type Of AES	Number of keys	Number of Blocks	Number of Rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Now since the card number is standard but by using the changed angle of finger print we can generate OTP by combination of these and by randomising the numbers every time so that the sequence of the number changes every time whenever the user requests an OTP. And IMEI number is used to authenticate whether the user is an authorised person or not by verifying the IMEI number of the user mobile with the IMEI number stored in the bank server which the user has given to the bank while registering.

Now Using AES-256 algorithm we are going to encrypt the generated plain text. Now the plain text gets generated by using the CNi, BMi and randomizing these number we get a 16-byte number. Now we are going to encrypt this generated plain text by using the above mentioned algorithm. AES process include a 4 * 4 column-major order matrix of bytes. If there are 16 bytes from b_0, b_1……, b_{15}, these bytes can be represented in the form of matrix

Now the combination of CDi, and BMi number are first permutated and randomized in such a way that every time whenever the user request the order keeps on changing and the sequence of the order is remembered so that the sequence is not repeated for the next time generation of OTP.

Process of an AES algorithm

1) Take the data into 4 * 4 matrix. In case if there is any of the cell is free fill it with 01 so it completes the matrix.
2) Add Round Key - EXOR operation takes place between each input byte with the corresponding byte of the first round key.
3) Main Rounds.

Sub Bytes- It is a non-linear substitution step where each byte is replaced with another according to the lookup table.

Shift Rows- It is a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

Mix Columns – In this a mixing operation takes place which operates on the column of the state, combining the four bytes in each column.

The description of the procedure followed by AES 256 is given below.

1) Add Round key:

In this XOR operation is performed between the initial sate that is with plain text with cipher key or key at round-0. This phase is called as initial round.

2) Sub bytes: In every state is replaced with input in S- box table also known as substitution box.

3) Shift Row: A round left shift is performed on every byte in each row of the stare. The number of shifts each byte is different is different from each line.

4) Mix column: Do randomisation of data in each column array state.

The main intention of modification of AES 256 is to increase the complexity and increase the immunity of the AES 256 algorithm. The modification will be done on the S-box and shift rows. The main reason for the modification to be done on the S-box is to increase the confusion of AES and modification on the shift Rows is to increase the diffusion of AES.

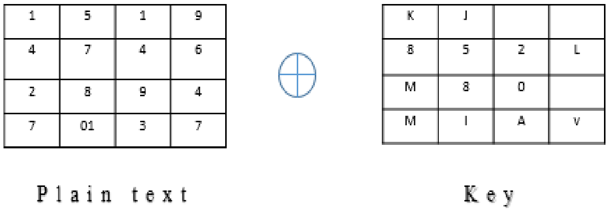
After the encryption is completed a cipher text and a cipher key gets generated from the plain text. After a cipher key is generated it is added with the last 4 digits of the IMi number for making sure that the legitimate user has received the message.

Now in the case of generating an OTP, at first we take the 16-byte number as a combination of CNi, BMi numbers a place it in a 4 * 4 square metrics.

TABLE 3
CNi, BMi NUMBERS

1	5	1	9
4	7	4	6
2	8	9	4
7	01	3	7

Now the plaint text which is fed into the metrics is a complete different sequence of plain text, since we have already discussed above how the plain text gets generated and how the permutation takes place according to the sequence number. The next step is the initial XOR operation to each input byte with the corresponding bytes of the first round key. The first round key keeps on changing every time the user requests an OTP. So the first round key is on always the same. So the above mentioned table XOR with the first round key operation is described in the following figure 7.



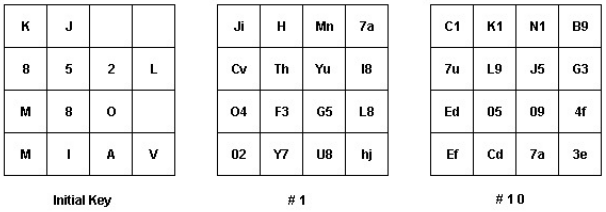
1	5	1	9
4	7	4	6
2	8	9	4
7	01	3	7

$\oplus$

K	J		
8	5	2	L
M	8	0	
M	I	A	V

Fig.7 XORing of Plain Text With Keys

The main reason behind using XOR to apply the key and another aspect it is fast and cheap that is a quick bit flipper. It uses minimal hardware and can it can be done parallel since no carry bits are needed. The next step is the expansion of the keys. The later keys are derived from the initial key using a simple mixing technique that's is very fast. The figure 8 shows how the expansion of keys takes place. We see that each bit of a rounds output depends on every bit from previous two rounds. So to increase the diffusion 4 extra rounds are add. So totally 14 rounds will take place and the plain text gets encrypted using the AES 256 algorithm. After the encryption is done the half of the key is sent to the user and the remaining half resides in the server. So the key acts as an OTP for authentication the user in order to complete the online transaction.



K	J		
8	5	2	L
M	8	O	
M	I	A	V

Initial Key

Ji	H	Mn	7a
Cv	Th	Yu	I8
O4	F3	G5	L8
O2	Y7	U8	hj

1

C1	K1	N1	B9
7u	L9	J5	G3
Ed	05	09	4f
Ef	Cd	7a	3e

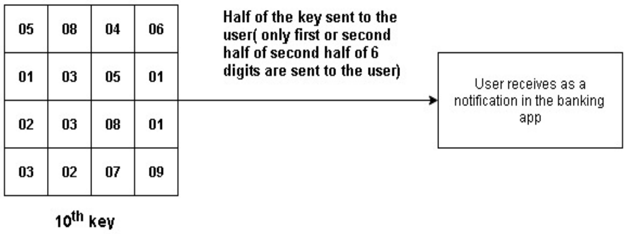
10

Fig. 8 Keys at Different Rounds

The figure 15 shows how the key depends on the previous key. The rest of the keys are generated by using the XOR operation.

C. Sending of OTP

After finishing of the encrypting of the plain text the next challenging task is to send the OTP. The OTP in this paper proposes is the use of the key. After the encryption is done a key will be generated at the 10th round. Off course there will be 10 key. When it comes to OTP the first half or the second half of the key will be sent to the user as an OTP. The choosing of the first half or second half of the key to be sent as OTP depends upon the previous OTP. That is if first half of the key is sent as OTP the for the second time when the user requests the OTP at that time the second half of the OTP is sent to the user. So by this randomised sense of sending of the OTP will be generated. The OTP that will be sent to the user make sure that the key contains only the digits. Even the count of the digits that are sent to the user as an OTP will be exactly 6 digits. So by this the user receives the OTP that is sent by the bank server. The overview of the OTP sending is described in the figure 9.



05	08	04	06
01	03	05	01
02	03	08	01
03	02	07	09

10th key
Half of the key sent to the user(only first or second half of second half of 6 digits are sent to the user)
User receives as a notification in the banking app

Fig. 9 Tenth Key Metrics

From the above figure 9 we can see that the 10th key which is of 4 * 4 matrix consists of 16 bytes of data. Of this data the first six digits of the key will be sent to the user as an OTP. From the above figure we can see that the keys are arrange in the matric column wise. So by taking out the digit's column we get the key as 05010203080303020405080706010109 so this is a 16-byte information when extracted from the metric. Since usually the OTP is of 6 digits the first half or the second half of the key is sent to the user as an OTP.

Figure 17 will give an overview of how the OTP reaches the banking server.

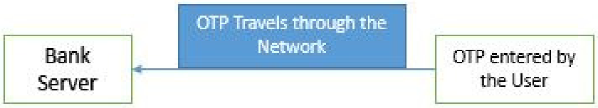


Fig. 10 Transfer Of OTP To Bank Server

From the Figure 10 we can understand that the OTP which was entered by the user travels through the network reach the bank server. After the OTP reaches the bank server the decryption process takes place. The description of the decryption process will be detailed in the next section of the paper.

D. Decrypting of OTP

In this section we will be seeing about the process of decryption. In this section we will be discussing about how the OTP will be used in the process of decryption, how the decryption process takes place and how the plain text is generated using the keys.

The decryption process is a reverse of the encryption process. In the case of encryption process the plain text is encrypted and the key that is produced by the algorithm is sent as a OTP to the user who have requested for the OTP. The user enters the OTP accordingly and the OTP which is half the key of the 10th key concatenates with the other half of the key and the plain text is generated by the process of reverse engineering. The figure 11 gives the overview of the procedure of the decryption process.

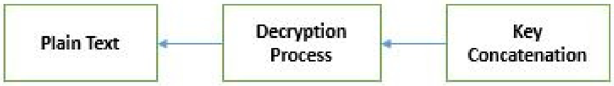


Fig. 11 Decryption to Plain text

The figure 11 show the overview of how the procedure of decryption process takes place. At first the OTP and the remaining half of the key gets concatenated and by using this key the remaining keys will be generated. After all the keys are generated by using the XOR operation on the previous just like in the case of key generation in the process of encryption. After all the keys are generated the 1st key is used to generate the plain text. Figure 11 will show the process of reverse engineering to generate the plain text.

In the case of decryption process at first Add round key takes place, then InvSubBytes, InvShiftRows, InvShiftColumns, AddRoundKey, InvSubBytes, InvShiftRows, AddRoundKey. This is reverse engineering process where we are coming right from the end of the encryption process and retrieve the original plain text with the help of Inverse methods like InvMixColumnsetc...So this is how the decryption process takes place when the users OTP is received by the bank server and the bank server process the OTP and concatenates with the 10th key generated and later using reverse engineering process the plain text is generated.

IV. CONCLUSIONS

The paper has proposed the best methods to the challenges that are faced by the OTP generation and transferring of the OTP to the user. This paper also provides with the best way of encrypting the OTP and sending it to the user and also describes the way of verifying the identity of the authorised user by verifying their mobile with the IMEI number. The paper also has proven to curb the most of the network attacks like sniffing, masquerading etc. in order to steal the sensitive information.

REFERENCES

[1]. Design of New OTP System using homomorphic graph by Changed Location and Angle of Fingerprint Features ByungRae Cha1,HyungJong Kim2,DongSeob Lee3

[2] Advanced Online Banking Authentication System Using One Time Passwords Embedded in Q-R Code Abhishek Gandhi, BhagwatSalunke, SnehalIthape, VarshaGawade, Prof.SwapnilChaudhari

[3] SMS Authentication Code Generated by Advance Encryption Standard (AES) 256 bits Modification Algorithm and One Time Password (OTP) to Activate New Applicant Account Eddy PrasetyoNugroho, RizkyRachmanJudhie Putra, ImanMuhamadRamadhan Department of Computer Science Education, Faculty of Mathematics and Natural Sciences Education

[4] 2FMA-NetBank: A Proposed Two Factor and Mutual Authentication Scheme for Efficient and Secure Internet Banking

[5] A STAND-ALONE AND SMS-BASED APPROACH FOR AUTHENTICATION USING MOBILE PHONE 1 Indu S., 2 Sathya T.N., 3 Saravana Kumar V.,

[6] N. Haller, C. Matz, P. Nesser, M. Straw, "A One-Time Password System", RFC 2289, IETF, 1998.

[7]Introduction to Biometrics, http://ics1.mk.co.kr/file/cd104/biometrics1.pdf

[8] Pankanti, S., Bolle, R. M., and Jain, A., “Biometrics: The Future of Identification”, IEEE Computer magazine, February, 2000.

[9] L. Hong, A. K. Jain, "Classification of Fingerprint Images", MSU Technical Report, MSU Technical Report MSUCPS: TR98-18, June 1998.

[10] Jain, A., and Pankanti, S., “Fingerprint Classification and Matching”, Handbook for Image and Video Processing, A. Bovik (ed.), Academic Press, April 2000. [10] One-Time Password (OTP), http://en.wikipedia.org/wiki/One-time_password

[11] RSA, http://www.rsa.com [12] OATH, http://www.openauthentication.org.

[12] Park Bonggu, Han Sangeun, Cha Byungrae, “Discrete Mathematics using Computer”, KyungMoon Publishing Company, 2003.

[13] JMSL, http://www.vni.com/products/imsl/jmsl.html

[14] ByungRae Cha, "Encryption Seed Generation System and Method using Structure Information of Fingerprint", Patent number: 10-0806365, The Korean Intellectual Property Office, Feb. 15, 2008.

[15] ByungRae Cha, KyungJun Kim, HyunShik Na, "Random Password Generation of OTP System using Changed Location and Angle of Fingerprint Features", CIT 2008, July, 2008.