Jaideep Murkute, Hemant Nagpure, Harshal Kute, Neha Mohadikar, Chaitali Devade / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
www.ijera.com
Vol. 3, Issue 2, March -April 2013, pp.1810-1815

Online Banking Authentication System Using QR-code and Mobile OTP

Jaideep Murkute, Hemant Nagpure, Harshal Kute, Neha Mohadikar, Chaitali Devade

Sinhgad College of Engineering, Department of Information Technology, University of Pune, Pune-411041.

ABSTRACT

This paper explains implementation details of online banking authentication system. Security is an important issue for online banking application which can be implemented by various internet technologies and gap between real world and virtual world can be filled up. While implementing online banking system, secure data transfer need can be fulfilled by using https data transfer and database encryption techniques for secure storage of sensitive information. To eliminate threat of phishing and to confirm user identity, QR-code which would be scanned by user mobile device can be used and weakness of traditional password based system can be improved by one time password (OTP) which can be calculated by user transaction information and data unique at user side like imei number of the user mobile device.

Keywords: banking application, security, QR-code, one time password (OTP), mobile device**.**

1. INTRODUCTION

Despite of wide use of current online banking system, it has many security holes as it's based on traditional password based model, no mutual authentication between user and bank server which leads to threats like phishing(stealing passwords and using them for transactions), intercepting communication lines, database hacking, etc.. To make transactions more secure but also keeping them easy for user, following authentication system can be useful.

In our proposed scheme, we assume the secure communication between the user (PC) ←→ service providers and service providers ←→certification authority. The proposed authentication system ensures the user authentication and digital signatures using authorized certificates by using https communication between user and server. Using user's transfer information (TI), requested transfer time (T) and the serial number (SN) of user's mobile device instead of security card, we generate QR-code, display it on user screen and decode it with user's mobile device to generate OTP. OTP is generated on server side also and OTP generated by user device and by server are verified to proceed [1]. User database should also be encrypted to prevent data leakage. The authentication process of proposed system is shown below:

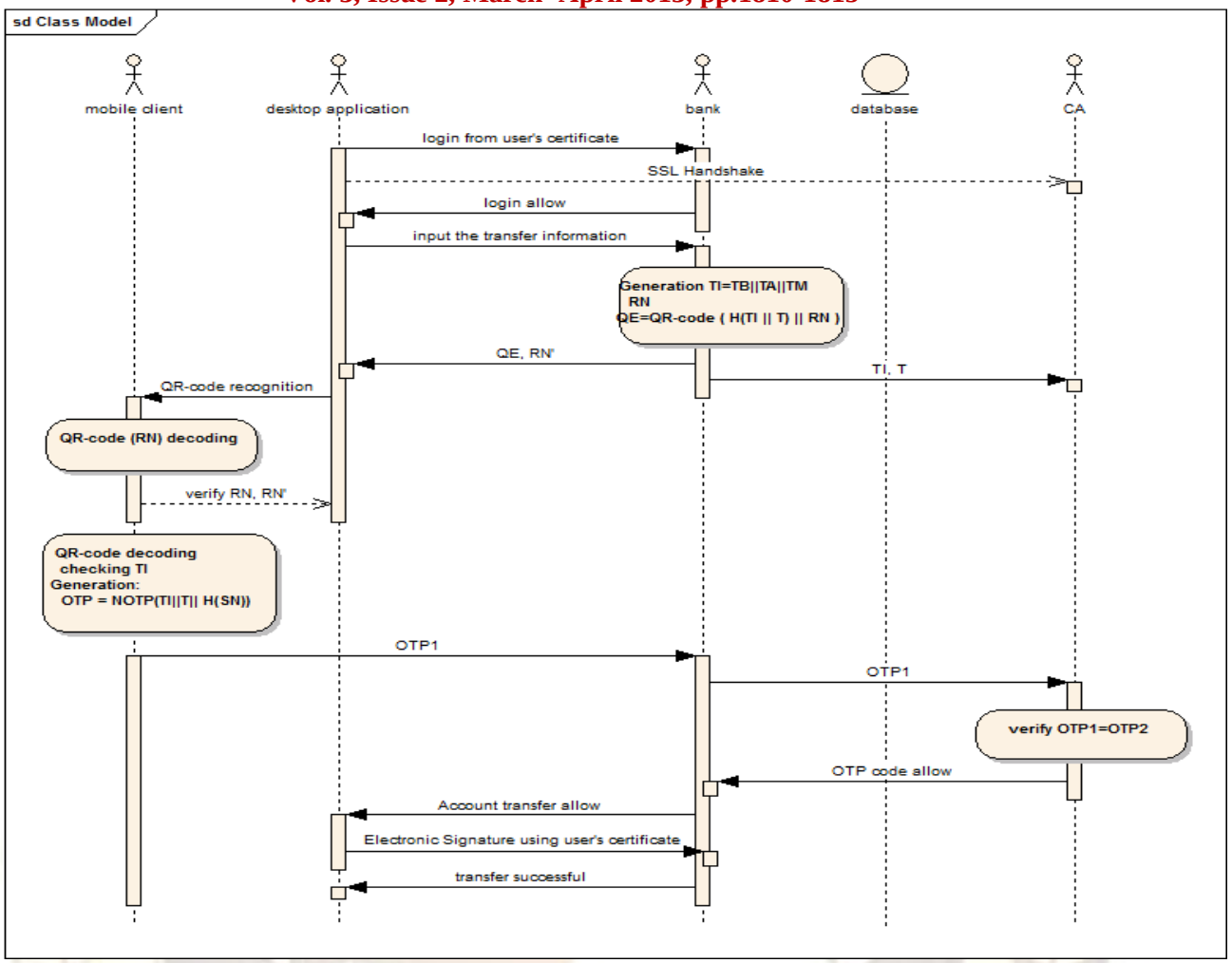

Fig. 1. Working scenario for online banking system

1] User uses his/her own public certificate to login and then transfer information to start the transfer transaction.

Transfer Information (TI)= TB‖T‖ATM TB: Transfer_Bank (Bank code) TA: Transfer_Account TM: Transfer_Money

2] Server indicates and then converted the information to a QR-code with random value (RN`) on the screen using user enters the transfer information (TI), the requested time of transfer (T) and random value (RN). At the same time, the server sent it to certification authority (CA) to inputted

code in the mobile device. If the information does not match, the transfer will be canceled

5] When user execute the generated OTP, mobile device generate the OTP by reads the transfer information (TI), perceived value of time (T) and information of transfer (TI) and the requested time of transfer (T).

3] Certification authority (CA) generated the OTP by received the transfer information (TI), the requested time of transfer (T) and the user's hashed serial number (SN).

4] User will convert the QR-code on the screen using their mobile device and it is divided into two phases. First, user uses their mobile device (phones) to read the random value (RN) which show on the screen to verify the random value (RN`). If the random value is accurate, user will proceed to the next step. And then confirm the converted the information of transfer. If the information is accurate, user will generate OTP hashed serial number (SN) of user's mobile device are shared with the certification authority (CA). And output the generated OTP on the screen of mobile devices.

6] User input the generated OTP code from mobile device on the screen.

7] Server (Bank) sent OTP to certification authority (CA) to received OTP from user.

8] Certification authority (CA) compared by received OTP code (OTP1) and generated the OTP code
(OTP2), sent to server (Bank) to for OTP code approval.

9] When the server (Bank) received approve of OTP from certification authority (CA), it will verify the entered OTP code with user consistent value and user digital signature. If the approve of OTP value does not receive, the transfer will be canceled. OTP is displayed on mobile screen and user types it into desktop application. Desktop client then sends this OTP to server.

10] Authorized user signed his certificates to complete the transfer.

11] Server (Bank) to verify the digital signature and final approve of transfer.

2. RELATED WORK

2.1 Calculation of OTP:

One-Time-Password (OTP) can be used. One time password system can be solution for this weakness which would generate new password every transaction and is based on two important factors: (a) a PIN to unlock the OTP generator (something you know) (b) the OTP smart card itself (something you have)[1].

Here in this system, QR code generated by bank server is displayed on client screen and is decoded by user mobile device. QR code is embedded with the information regarding current transaction, timestamp and data unique for every user device like imei-number.

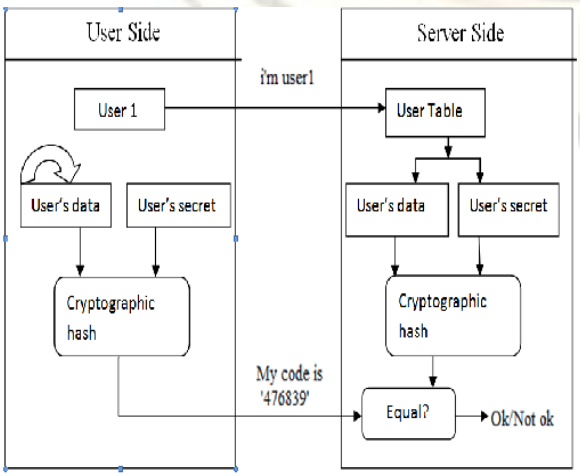

Fig. 2. OTP creation and validation

We can get data string from QR code; append it with imei number which can be obtained from mobile device. Then hashing function like SHA-256 is used to create hashed string of that data. Other hashing algorithms also can be used. But longer the hash code, more it is difficult to guess the OTP for an attacker. Hashed string comprised of both digits and characters. We will select any 6 or 8 digits/character or both of generated hash and use it as OTP.
Same hash of the data will be created on server side also and compared for equivalence, ensuring mutual authentication. If both OTPs are same, transaction is permitted.

Advantages of using hashing algorithm like SHA is same hash is never generated for same data in consecutive attempts, so intercepting data and calculating hash won't be possible for an attacker.
SHA-256("The quick brown fox jumps over the lazy dog")
0xd7a8fbb307d7809469ca9abcb0082e4f8d5651e4
6d3cdb762d02d0bf37c9e592

SHA-256("The quick brown fox jumps over the lazy dog.")
0xef537f25c895bfa782526529a9b63d97aa631564d
5d789c2b765448c8635fb6c

So as per system, OTP for above will be:
53725895 (using first 8 digits).
And timestamp ensures that OTP for transaction generated at different times will be different. This OTP can also be called HOTP as hashing technique is used. We can also use H-MAC codes but it would need an extra input to generate output.

2.2 Database encryption:

One of the major security holes in many critical systems is database security. Though attacker gets invalid access to database, one more level of security can be added by encrypting database. While displaying contents we'll decrypt data and send it to user. Any of the available encryption algorithms can be used but as there will be many database requests for banking application, encrypting-decrypting every time might put large overhead on the application. So care should be taken to choose an algorithm which would provide sufficient security with little overhead.

Base-64 is one of the choices. Algorithm converts data in byte-code. Standard data representation is of 8-bits. We can take 6-bit groups and convert them into characters and replace the original data. Padding can be added in the end of data if necessary. It would represent data by 2^6=64 possible characters, so named base-64[13].

Along with security, another advantage of base 64 is that many internet system don't allow all 128 characters in 8-bit representation so, base-64 can be beneficial.

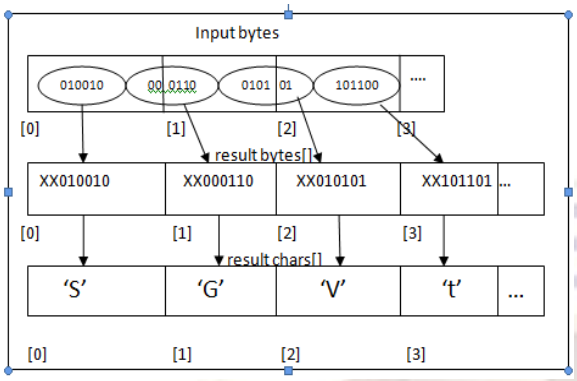

Fig.3. Base-64 working

2.3 Secure Communication Channels:

As important as application security, secure communication channels also of equal importance. Most promising way to do this would be use of digital certificates using PKI architecture for application. PKI provides an additional encryption and signature. HTTPS communication can be used for this purpose. It embeds HTTP data [11] in SSL (Secure Socket Layer) packets. SSL group data into small chunks compresses them and then encrypts using asymmetric keys [12]. Asymmetric keys provide high level of security for communication as one key is used for encryption and another for decryption. For management of keys, digital certificates are used which legitimate documents are provided by certification authority (CA) containing user information and keys.

For asymmetric key generation, RSA (Rivets-Shamir-Adelman) algorithm is used. Public keys are embedded in digital certificates of each end. Data is sent by encrypting it with public key of receiver but can be decrypted only with private key of receiver which is kept secret, thus providing high level of security [9].

2.4 QR-code processing:

The features of this code symbol are large capacity, small printout size and high speed scanning. QR code comprised of following patterns:
finder pattern, timing pattern, format information, alignment pattern, and data cell.

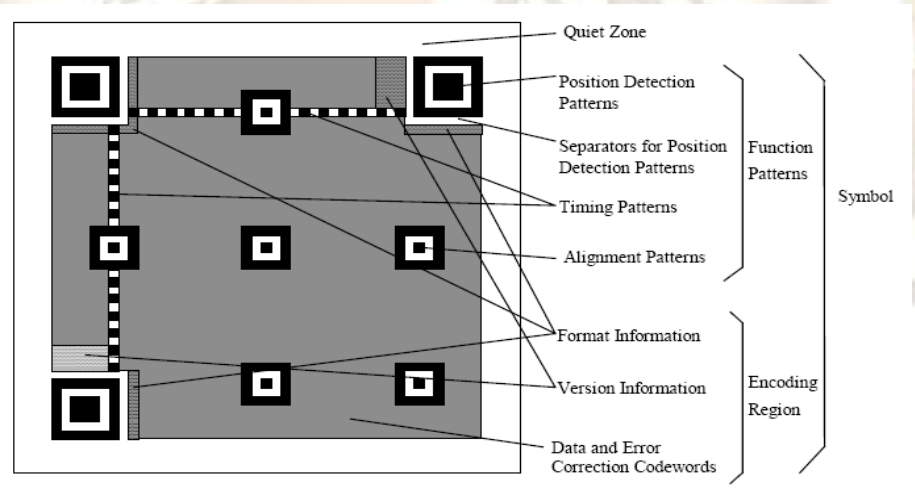

Fig. 4. Structure of QR-code

Use of QR code ensures that data will be decoded by legitimate user only as decoding device will be required to decode it.

2.4.1 QR-code is generated using transaction information, timestamp, random number using following steps[5]:

(I)Conversion into binary format:

First we select mode in which QR-code to be generated depending on type of data:
Extended Channel Interpretation (ECI) Mode
1.Numeric Mode
2.Alphanumeric Mode
3.8-bit Byte Mode
4.Kanji Mode

Each of the modes has got different conversion functions to convert data into binary format.

(II)Appending error correction codewords:

Divide the codeword sequence into the required number of blocks to enable the error correction algorithms to be processed. Generate the error correction codewords for each block, appending the error correction codewords to the end of the data codeword sequence.on of the 4 levels of error recovery(L,M,Q,H) is chosen to generate codewords.

(III)Codeword placement in matrix:

Data blocks are arranged into QR-code according to chosen strategy: either into rectangular blocks or irregular blocks which can accommodate more data.

(IV)Masking:

Data is XORed with predefined bit-string to encode, for dark and light modules to be arranged in a well-balanced manner in the symbol.

(V)Appending format information:

The Format Information is a 15 bit sequence containing 5 data bits, with 10 error correction bits calculated using the (15, 5) BCH code.

(VI)Appending version information:

The Version Information is an 18 bit sequence containing 6 data bits, with 12 error correction bits calculated using the (18, 6) BCH code.

For error detection and correction **"reed-soloman"** codes of data are also embedded in QR code. It gives error correction up to 30%.The generator polynomial g(x) is defined by having α, α^2, ..., α^t as its roots, i.e.,

.

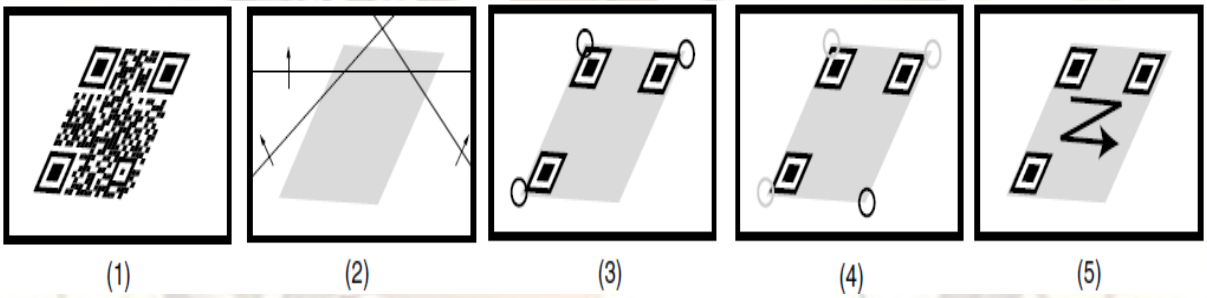

Fig. 5. Steps in QR-code scanning

Scanning can be done by using following five steps:

(I)Pre-processing:

The gray level histogram calculation is adopted.

(II)Corner marks detection:

Three marked corners are detected using the finder pattern.

(III)Fourth corner estimation:

The fourth corner is detected using the special algorithm.

(IV) Inverse perspective transformation:

Inverse transformation is adopted based on the obtained corner geometry positions to normalize the size of the code.

(V) Scanning of code:

Sample the inside of code and output the normalized bi-level code data to host CPU.

The input image has a deformed shape because of being captured from the embedded camera device, and we use the inverse perspective transformation to normalize the code shape. This equation is shown as follows:

$$u = \frac{c0x+c1x+c2}{c6x+c7y+1}$$

$$v = \frac{c3x+c4x+c5}{c6x+c7y+1}$$

$$g(x)=(x-\alpha)(x-\alpha^2)\ldots.(x-\alpha^t)=g_0+g_{1x}+\ldots\ldots+g_{t-1x}{}^{t-1}+x^t$$

The transmitter sends the $N-1$ coefficients of S(x)=p(x)g(x), and the receiver can use polynomial division by $g(x)$ of the received polynomial to determine whether the message is in error; a non-zero remainder means that an error was detected. Let $r(x)$ be the non-zero remainder polynomial, then the receiver can evaluate $r(x)$ at the roots of $g(x)$, and build a system of equations that eliminates $s(x)$ and identifies which coefficients of $r(x)$ are in error, and the magnitude of each coefficient's error.

2.4.2 Scanning of QR-code:

The processing of QR-code detection consists of five procedures starting from image captured from camera to data extraction. Thing that makes this task challenging is that captured image may not be of good quality or might be deformed either by limitation of device or naïve user.

Where u, v coordinates is original image coordinate which is deformed and x, y coordinate is the normalized coordinate. In the above equations, coefficients $c0 \sim c7$ can be obtained from the following four point pairs,

$A(x0, y0) \Leftrightarrow A\_ (u0, v0)$,
$B(x1, y1) \Leftrightarrow B\_ (u1, v1)$,
$C(x2, y2) \Leftrightarrow C\_ (u2, v2)$,
$D(x3, y3) \Leftrightarrow D\_ (u3, v3)$

(C) QR-code decoding:

QR-code is encoded with encryption key, which is then decoded by private key at user and data is obtained. Decoding would be the exact opposite of the encoding scanning different sections according to format of QR-code, checking data with error correction codes, recovering lost data from redundant locations is done while decoding

Random number is matched with the number sent along with the message and if they

match, message is valid. Timestamp is read from the message to get synchronized with the server.

From information in QR-code like TI and T and imei-number of the mobile device, OTP is generated in the device and displayed to user. User then will enter it into desktop application and is sent to CA where also OTP for current transaction is generated and matched with the one sent by user application. If they are same transaction is completed.

Other functionalities required by any banking application should be added into the applicant like user registration, managing user accounts, viewing transaction summary, etc. and application confirming authentic, secure transaction, storage and communication can be developed.

3. SECURITY ANALYSIS

A malicious user can not analyze the content of communications as our propose system use the camera of mobile device to recognize of QR code. Also the user and Certification Authority (CA) has been shared the hashed serial number (SN) of user's mobile device through a secure process in the initial registration phase. If altered the PIN, the OTP value is change.

In our proposed system, the user to prevent Phishing attacks by identifying the value of random number (RN) before to verify the information of transaction in the conversion of QR code.

Our proposed system require a prerequisite input of transaction information using QR code and authorized authentication by the public certificate for the generation of OTP. Therefore it can identify the legitimate users and can block the use of malicious user. As we used the user's requested time of transfer, the time value used to generate the OTP code is not possible to change arbitrarily.

4. CONCLUSION

Now a days, use of online banking application are increased. Security is an important issue for handling such services. Current system provide security card based facility to authenticate user but this is not much more secure and will not be available for any time or situation. To overcome such type of issues we propose online banking authentication system using QR-code and OTP. The bank generates the QR-code using user input transfer information and then user need to recognize as to read the code using their mobile phone, after generate the OTP code using transfer information and the hashed user's mobile device number in their mobile phone. Finally, terminate the transfer by user typing of generated OTP code on the screen.

For any system, security it provides and system overhead are two sides of a coin and should be considered equally while developing critical information of transfer (TI) and the requested time of transfer (T).

REFERENCES

1] Young Sil Lee, Nack Hyun Kim, Hyotaek Lim, HeungKuk Jo, Hoon Jae Lee," Online Banking Authentication System using Mobile-OTP with QR-code", Page(s): 644 – 648, Nov. 30 2010-Dec. 2 2010, E-ISBN : 978-89-88678-30-5.

2] IETF RFC 4226, HOTP: An HMAC-Based One-Time Password Algorithm, Dec. 2005.

3] AntiPhishingGroup, "Phishing Activity Trends Report", from: http://www.antiphishing.org, dec. 2008.

4] Mohammad Mannan, P. C. Van Oorschot, "Security and Usability: The Gap in Real-World online Banking", NSPW'07, North Conway, NH, USA, Sep. 18-21, 2007.

5] Eisaku Ohbuchi, Hiroshi Hanaizumi, Lim Ah Hock," Barcode Readers using the Camera Device in Mobile Phones", IEEE paper.

6] Aidong Sun, Yan Sun, Caixing Liu," The QR-code reorganization in illegible snapshots taken by mobile phones", IEEE paper

7] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O. Ranen ,"HOTP: An HMAC-Based One-Time Password Algorithm" , , RFC 4226, December 2005.

8] Teoh Chin,Yew Mazleena,Salleh Subariah Ibrahim, "Spatial Resource Analysis of Two Dimensional Barcodes", IEEE Paper.

9] R.L. Rivest, A. Shamir, and L. Adleman,"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems",http://people.csail.mit.edu /rivest/Rsapaper.pdf.

10] Robert P. McEvoy, Francis M. Crowe, Colin C. Murphy, William P. Marnane,"Optimisation of the SHA-2 Family of Hash Functions on FPGAs".

11] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee,"Hypertext Transfer Protocol -- HTTP/1.1",Network Working Group, Request for Comments: 2616

12] David Wagner, Bruce Schneier,"Analysis of the SSL 3.0 protocol",http://www.schneier.com/paper-ssl.pdf.

13] Randy Charles Morin,"How to Base64",www.kbcafe.com.