International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 6, June 2015, PP 31-37
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Pervasive and Secure M-Banking using Steganography

Mr. Anup Kadam[1], Mrs. Swati Joshi[2]

[1]Computer Department, Sinhgad College of Engineering, Pune, India (ME Student)
[2] Computer Department, Sinhgad College of Engineering, Pune, India (Associate Prof.)

ABSTRACT

Overwhelming use of smartphone and phenomenal growth in internet user worldwide .Smartphone usage for all your need is mandatory now a day, mobile banking is not an exception for it. Most of the banks extending mobile banking to include a full complement of transactional and interactive services. These security challenges are varied and increasing in number due to huge amount of money flowing across the consumers and banks. Banks are searching for various types of options to preserve privacy of user and protect them from several attacks. In this paper we focus on smartphone mobile banking with providing two factor biometric authentication for a smartphone mobile user i.e. username and password and face recognition [10][11]. We also propose the use of various methods of steganography selecting randomly to improve the communication channel security for any intrusion and detection by the hackers.

Keywords: LSB, SLSB, Random Bit,RGB

INTRODUCTION

New era smartphones are very powerful and they can perform all operation, which personal computer can. Technologies drive the need in every sector and enterprise needs to understand changing need of customer [4]. Financial sector has also no exception. Integrating mobile devices like smartphones and tablets into an enterprise gives employees possibilities to work more productively. In order to satisfy all financial need for customer banks are taking help of smartphone and faster internet by developing smarter and secure applications .However, integrating smartphone with applications has also brought diverse security challenges and risks. In spite of all advantages of mobility, flexibility and robustness of using internet on smartphone, many banks are using conventional security mechanism. Smartphone devices are exposed to a wide range of threats like a personal computer that have to be countered. Simply implementing information security standards from server domains with mobile devices is unlikely to be effective for banks and user. Thus, from banking point of view, security levels are not clear on Smartphone devices [10][11]. Generally, a high level of security might be reached on Smartphone devices by setting a high level of restrictions. This will minimize user acceptance for application and satisfaction factors [7].

Here we make use of total 3 different algorithm i.e. LSB,SLSB ,Random bit steganography .The random selection of any algorithm from LSB ,SLSB ,Random bit for sending username and password in encrypted format to server increase the security . Only single key is sent along with image used in

Steganography (cover image) in interdependent manner. On the basis of key value the server will detect, which algorithm need to use for decryption of username and password .Then server initiate the request for starting of camera on mobile device .Using camera user will take his/her picture and send it and will match the face with available database .Once face is authenticate then further transaction started with secure way by sending all details in image.

RELATED WORK

Methods Used

Faster internet and with the development of digital signal processing, information theory and coding theory, steganography has gone "digital".[7] For a computer, an image is an array of numbers that

Address for correspondence:

anupkadam01@gmail.com

represent light intensities at pixels. These pixels make up the image's raster data up to 300kb. A common image size is 640 × 480 pixels and 256 colours (or 8 bits per pixel). Digital images are typically stored in either 24-bit or 8-bit files. A 24-bit image provides the most space for hiding information. All colour variations for the pixels are derived from three primary colours: red, green, and blue. Each primary colour is represented by 1 byte; 24-bit images use 3 bytes per pixel to represent a colour value. These 3 bytes can be represented as hexadecimal, decimal, and binary values. In many Web pages, the background colour is represented by a six-digit hexadecimal number actually three pairs representing red, green, and blue. A white background would have the value FFFFFF: 100 percent red (FF), 100 percent green (FF), and 100 percent blue (FF). Its decimal value is 255, 255, 255, and its binary value is 11111111, 11111111, 11111111, which are the three bytes making up white.[8,9]

LSB

Least significant bit (LSB) insertion is a common used, with simple approach of embedding information in a cover image [14]. The least significant bit (8th bit) of some of the word or all of the bytes of all word inside an image is changed to a bit of the secret message. When using a 24-bit image, a LSB bit of each of the red, green and blue (RGB) colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An image of pixel size 800 × 600 , can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data .For example a grid for 3 pixels of a 24-bit image can be as follows;

```
(00101101    00011100    11011100)
(10100110    11000100    00001100)
(11010010    10101101    01100011)
```

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

```
(00101101    00011101    11011100)
(10100110    11000101    00001100)
(11010010    10101100    01100011)
```

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary colour like RGB, changing the LSB of a pixel results into small changes in the intensity of the colours. These changes cannot be perceived by the human eye, thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference.

In its simplest form, LSB uses BMP images because of using lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion. For this reason, LSB steganography has also been developed for use with other image file formats.[8,9]

SLSB

SLSB algorithm filters the cover image by use of default filter with hiding information in those areas that get better rate. The e filter is applied to the most significant bits (MSB) of every pixel from the image, leaving the less significant to hide information .This filter confirm the choice of areas in the image for the least impact with the inclusion of information, which affects a greater difficulty of detecting the existence of hidden messages. The retrieval of information is confirmed because the bits used for filtering are not changed, inferring that the reapply the filter will select the same bits in the process of cover-up .It is the most efficient method to hide information [8,9].

Hiding Information in Only one Colour

Most of the algorithm that work in the spatial domain using a LSB method. The algorithm for information hiding, that is, it hide one bit of information in the least significant bit of each colour (i.e.

RGB) of a pixel. The problem steam from the modifying of these three colours of pixel produces a major distortion in the resulting colour of image .This distortion is not visible to the human eyes, but detectable by special statically analysis for images. For example, if a pixel in the cover image with RGB (red-Green-Blue code) colour A8A8a8 # is used binary 10101000-10101000-10101000, and 1 bit with value 1 is set on each LSB bit of each colour component, to hide given message 111, the result would be 101010001-10101001-10101001.

Random Bit Steganography

1) First of all, the particular pixels altered to encode each letter are semi-random.
2) Whatever message we want to embed within an image, the first step is to count the number of letter in the message that we want to embed and divide the total number of rows of pixels in the image by that number.
3) For Example: suppose the image is of 1000 pixels tall and we want to embed a message 100 letter in length, divide 1000 by 100 to get 10 rows per letter .This is the number of rows in the pixels that could be the associated with each letter.
4) Next, take the number of pixels wide image and divide by the number of letters in the alphabet (26). So if the image is of 780 pixels wide, divide by 26 to get 30 columns per letter of the alphabet.
5) By grouping these sets of 10 rows and 30 columns together, we get a grid of set of pixels within the image.
6) Each of these sub grids corresponding to an encoding of a different letter of the message.
7) The groups of rows corresponds to the index of the letter in the message (e.g., first letter, second letter etc.) and groups of columns correspond to the actual letter encode Figure 1.

Mathematically Embedding a Message:

Encoding:

```
1) Cr = Nr/Nl
2) Cc = Nc/C
3) If Index < N
4) Key = h[m]
5) Value = h[Key];
   Row = Cr * index + (random bit)
   Col  = Cc * Value + (random bit)
6) Bit = image [row] [col]
7) Image [row][col] = !bit
```

Decoding:

- By comparing the altered image compared to the original image can easily determine the message.

IMAGE GRID	COLS 0-25	COLS 26-51	COLS 52-77	COLS 78-103	COLS 140-129
Row 10-19	Letter 1=a	Letter 1=b	Letter 1=c	Letter 1=d	Letter 1=e
Row 20-39	Letter 2=a	Letter 2=b	Letter 2=c	Letter 2=d	Letter 2=e
Row 30-49	Letter 3=a	Letter 3=b	Letter 3=c	Letter 3=d	Letter 3=e
Row 40-49	Letter 4=a	Letter 4=b	Letter 4=c	Letter 4=d	Letter 4=e
Row 50-59	Letter 5=a	Letter 5=b	Letter 5=c	Letter 5=d	Letter 5=e
Row 60-69	Letter 6=a	Letter 6=b	Letter 6=c	Letter 6=d	Letter 6=e

Figure1. *Encoding the message*

- Even this randomness since the letter appears in the message in the same order that we find altered bits by going down the row.
- Only decoder knows which groups of columns corresponds to each letter.

Our approach for improving unpredictability

i. Find the last bits of first three columns.

ii. This number will gives us the exact column number on which we check bit variance.

iii. To increase the payload capacity, we can use the previous or next column for embedding bits.

Android Platform

The Android™ platform delivers a complete set of software for mobile devices: an operating system, middleware, and key mobile applications [6]. Android is offering new opportunities for mobile applications by offering an open development environment to be builded on an open source Linux kernel. Real hardware can be accessed through a series of standard API libraries, allowing the user to manage Wi-Fi, Bluetooth, and GPS devices and advance gaming. Open handset alliances and Google support the Android platform and hope to reach the goal of ensuring global mobile services that operate across devices, geographies, service providers, operators, and networks. Google has already released the open source Android platform, providing the opportunity to create new adaptive mobile platform interfaces and applications designed to look, feel, and function as desired. Consequently, the Android platform has recently been ported into mobile devices, such as notebooks, PDAs, home appliances and automotive systems.[6]

Android Software Stack

Figure. 2 illustrates the Android software stack [5], which consists of a Linux kernel, a collection of Android libraries, an application framework that manages Android applications in runtime, and native or third-party applications in the application layer. The following list specifies these Android software stack components:

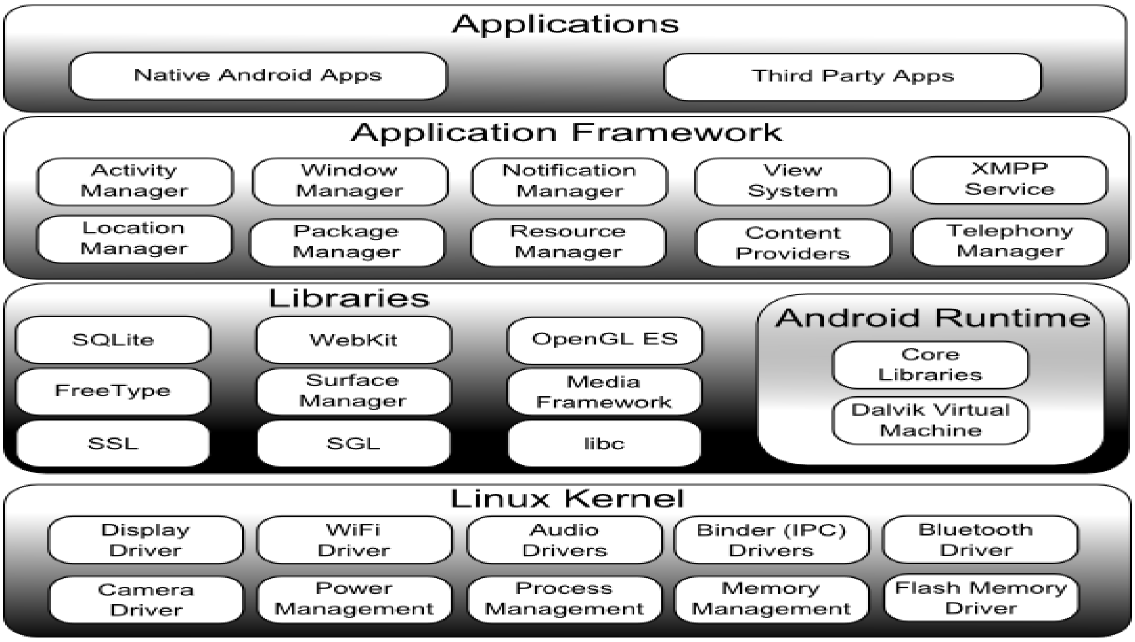

Figure2. *Android software stack*

Authentication Mechanism Overview for Banks

The following are the various options used under each of the three factors [17].

User knows	User possesses	User Is
Username	USB Token	Fingerprint
Password	Smart Card	Palm print
PIN	OTP by	IRIS
Card No.	SMS/token	Voice
CVV2	Swipe cards	Vein pattern
3D Secure/ VbV	Mobile Signature	
Identifiable picture		

Figure3. *Security Mechanism overview of Bank*

Security Pitfalls of Various Schemes

a. Remembering password is chosen by the server and send it to user which might be long, random and difficult for a user used in various schemes.

b. Transmission of login message to the authentication server over insecure channel like internet is risky. In scenario like transaction, it is very important to preserve the privacy of a user because of an adversary sniffing in the communication channel. The communication parties involved in the authentication process to analyse the transaction being performed by the user.

c. Losing of smart cards or random number generator is one of the very serious problems because the lost card or random number generator can copy valid registered user.

d. Information like username and password through the unsecure channel like internet, anyone can extract information by packet sniffing.

PROPOSED PROCEDURE FOR MOBILE BANKING

In India leading banks are using the secure image to send username and password in encrypted format in image. Images has to be selected from limited set of images selected by bank .They are selecting these limited images of some size and of specific resolution provides by bank. They are embedding username and password into the image and sending it through the unsecure network.

Our propose model for increasing the security, we are using mobile client for embedding username and password in image on the mobile .We are also selecting the randomly algorithm from the any of the steganography algorithm like LSB, SLSB, Random bit .So, that the intruder or the hacker even if know the algorithm used for encryption he will have to try all possible algorithm to break it. Second method of authentication, we are using the face detection mechanism for user authentication. The detailed procedure of using the mobile phone for M-banking is depicted as the following.

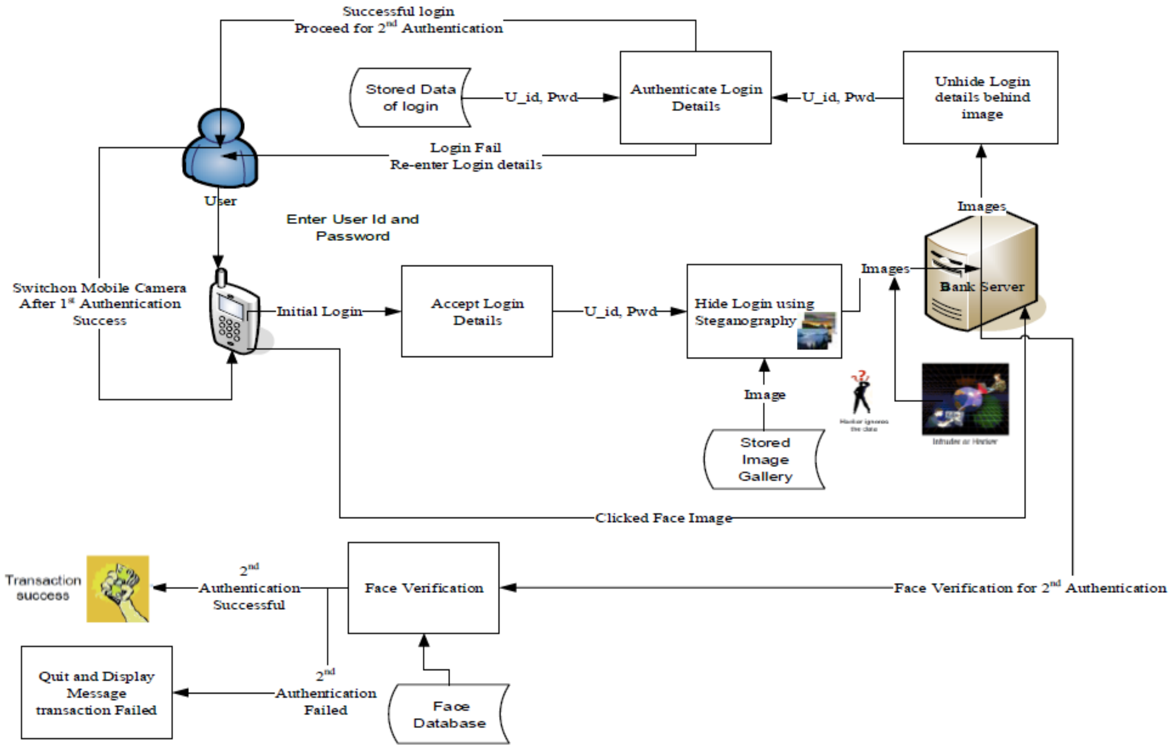
Successful login
Proceed for 2nd Authentication
Stored Data of login
U_id, Pwd
Authenticate Login Details
U_id, Pwd
Unhide Login details behind image
Login Fail
Re-enter Login details
User
Enter User Id and Password
Images
Switchon Mobile Camera After 1st Authentication Success
Initial Login
Accept Login Details
U_id, Pwd
Hide Login using Steganography
Images
Bank Server
Image
Stored Image Gallery
Hacker ignores the data
Intruder or Hacker
Clicked Face Image
Transaction success
2nd Authentication Successful
Face Verification
Face Verification for 2nd Authentication
Quit and Display Message transaction Failed
2nd Authentication Failed
Face Database

Figure4. *Prepose mechanism for Secure Banking*

Step1: User login with their user Id and password to the mobile client application on android device. This user name and password will be encrypted in any image selected by user from SD card. Mobile client will encrypt username and password in image by using random selection of any algorithm from LSB, SLSB, and Random bit. This will be send to server by using the unsecure channel.

Step2: Server side decrypts the user name and password by using decryption mechanism that is used while sending this image. If the password is not match with the password on server side then it will return fail authentication .If match is found then step 3.

Step3: If match found with username and password then server will send the request to mobile client to start the camera instance.

Step4: .Once mobile camera is started then, user will click his image and send it back to the server. Server will match the image with the available database by using face detection algorithm.

Step5: Match is done then further transaction operation will be started.

Step6: Logout and finish the M-banking activity.

IMPLEMENTATION

The android mobile client must be equipped with a smartphone with a camera having good resolution and capability of browsing the internet through Wireless Access Protocol. A dedicated standalone client/server architecture base application is needed for the successful realization of communication between the user and the bank. However, the bank must provide the user with the necessary client software or client application can be downloaded and install on smart phone form the android Paly store. This application can be used on smartphone with android operating system.

EXPERIMENTAL RESULTS

1) LSB
   - MSE:199.22092572338
   - SNR :25.09733954978415
   - PSNR(Max=255):25.137454070569603
   - PSNR(Max=255.0):25.137454070569603
2) SLSB
   - MSE:193.943547313527
   - SNR :25.214137844913044
   - PSNR (Max=255):25.25405026082373
   - PSNR (Max=255.0):25. 25405026082373
3) Random Bit
   - MSE:185.8559960178963
   - SNR :25.399178654567322
   - PSNR(Max=255):25.43903784161051
   - PSNR(Max=255.0):25.43903784161051

The era of mobile banking is no end and the proposed authentication mechanism can be extended to mobile shopping which has also grown quite rapidly with the introduction of online marts. Various organization like government and privet can supply and promotion of electronic services through mobiles. The smartphone functionality can be further extended to various places .Mobile voting using biometric authentication like camera .Which will uniquely identify each individual and they could cast their votes remotely. In all of the above applications the role of authentication becomes very important and our scheme proves to be very robust and secure in such scenarios.

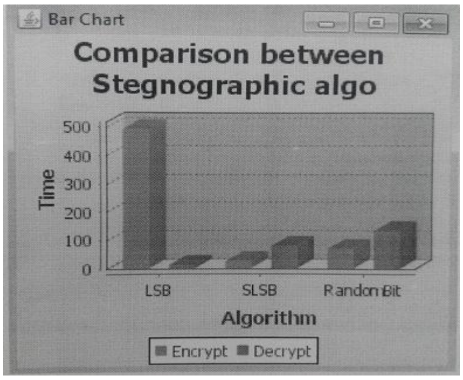

Figure5. *Comparison between Stenographic algorithm*

CONCLUSION

In this paper, we have discuss various technique of steganography for remote user authentication schemes that is used for mobile client. Firstly, we discuss steganography methods used for increasing the security. There are lot of problem Identified in the available security mechanism provided with mobile banking. We try to propose an enhanced biometrics based stenographic approach with face detection, which improves all the identified weaknesses and is more secure and robust for real-life use of mobile banking. The proposed scheme can withstand the fictitious authenticating attacks besides providing better and secure mechanism.

REFERENCES

[1] Rethink the Mobile in Mobile Banking March 2013, Copyright © 2013: Vishal Chaturvedi:, Oracle and/or its affiliates. All rights reserved.

[2] Security for Mobile ATE Applications: Susan Moran Senior Software Engineer, EADS North America Test and Services Irvine, CA: 978-1-4673-0700-0/12/$31.00 ©2012 IEEE.

[3] Secure OTP and Biometric Verification Scheme for Mobile Banking: Chang-Lung Tsai Chun-Jung Chen, Deng-Jie Zhuang: 2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, 978-0-7695-4727-5/12 $26.00 © 2012 IEEE.

[4] Understanding Android Security: Published by the IEEE Computer society: 1540-7993/09/$25.00 © 2009

[5] http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture.

[6] Breaking and fixing the Android Launching Flow: Alessandro Armando, Alessio Merlo,Mauro Migliardi,Luca Verderame : computers & security 39 (2013) 104 e115: 2013 Elsevier Ltd. All rights reserved.

[7] Steganographic Authentications in conjunction with Face and Voice Recognition for Mobile Systems: Dushyant Goyal, Shiuh-Jeng Wang:

[8] Exploring Steganography: Seeing the Unseen: Neil F. Johnson, Sushil Jajodia :George Mason University: 0018-9162/98/$10.00 © 1998 IEEE.

[9] Hide and Seek: An Introduction to Steganography: Niels Provos , Peter Honeyman: University of Michigan: 1540-7993/03/$17.00 © 2003 IEEE.

[10] A Framework along with Guidelines for Designing Secure Mobile Enterprise Applications: Basel Hasan, Viktor Dmitriyev, Jorge Marx Gómez, Joachim Kurzhöfer: 978-1-4799-3532-1/14/$31.00 IEEE ©2014

[11] Security and Privacy Risks in Mobile Applications: Anurag Kumar Jain, Devendra Shanbhag, Tata Consultancy Services: Published by the IEEE Computer Society 1520-9202/12/$31.00 © 2012 IEEE.

[12] https://www.duosecurity.com/blog/the-current-state-of-online-and-mobile-banking-security

[13] http://www.sans.edu/research/securitylaboratory/article/2factor-banks

[14] http://www.pcworld.com/article/2079620/banks-shouldnt-rely-on-mobile-sms-passcodes-security-firm-says.html

[15] http://www.phonearena.com/news/90-of-mobile-banking-apps-have-security-problems_id51390

[16] http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture.

[17] Authentication factors for Internet banking: M V N K Prasad and S Ganesh Kumar: IDRBT Working Paper No. 11