M-Commerce Security Using Random LSB Steganography and Cryptography

Pratiksha Y. Pawar and S. H. Gawande, *Member, IACSIT*

***Abstract*—M-commerce is one of the main branches of e-commerce. The banking industry is among the leading sectors in adopting and utilizing the Internet and mobile technology on consumer markets. Mobile banking is a subset of electronic banking which underlies not only the determinants of the banking business but also the special conditions of mobile commerce. The development of electronic banking and mobile banking services via multiple channels has made it possible to create new kinds of added value for customers. However, in spite of their advantages, both are facing some challenges as well. One of these challenges is the issue of security of these systems. This paper presents security of these systems using Random LSB steganography and cryptography method. The proposed method is more safe and secure instead of using either steganography or cryptographic method. This paper shows secure and invisible communication in M-banking as well as e-banking. In this paper instead of direct sending information, it is encrypted first using encryption algorithm and then this encrypted information is processed to hide into an image using a password so that stego-image contains hidden message which is not in plaintext form. Another important point is that encrypted information is hidden into an image using "Random LSB Steganography" that is embedding data in non sequential LSB insertion pattern so that it is unintelligible and difficult to detect. The stego-image is put on a web site then the URL of the web site is sent to the user. After receiving the URL, the user downloads the picture by a special program. The user can extract information from the picture only if the password entered is correct. This information will be in encrypted form user will decrypt it using the decryption algorithm so that user will get required information. The proposed scheme has been implemented using J2EE language for e-banking and J2ME language for m-banking. Our implementation supports all java enabled mobiles for m-banking application**.

***Index Terms*—M-commerce, M-banking, E-commerce; E-banking, steganography, cryptography.**

I. Introduction

Electronic banking and Mobile banking are seen as one of the most successful business-to-consumer applications in electronic commerce and mobile commerce [2]. The use of e-banking and m-banking especially in developed countries has grown rapidly. Low fees, time savings and freedom from time and place [9] have been found to be most important elements of e-banking and m-banking. These services are easy to use [8], convenient and compatible with lifestyle [7; 4], speed of service delivery is fast [8]. Electronic banking significantly changed the way in which many customers accessed their bank account.

Banks greatly support this not only because they could meet their customer's need for convenience but also because of the enormous economic impacts in replacing a high-cost channel (bank clerks) through a low-cost channel (a central web server) for simple transactions, with the additional benefit of eliminating the necessity for a media conversion. Since users considered their mobile phone as a personal trusted device making it to an integral part of their lives and most of these devices became Internet-enabled, the regular conclusion was the transformation of banking applications to mobile devices as the next step of electronic banking development.

For mobile banking, the advantages even go much further than for electronic banking: The high penetration of mobile phones reaches all social levels, mobile applications disband the limitations of electronic banking as they allow for a use anytime-anywhere and the subjective and objective security of the device is higher than that of a personal computer [3].

There are two types of services offered in e-banking and m-banking, i.e. A) Notifications and alerts and B) Information, in which the bank sends messages containing information or notification needed by the customer. Although the protocols in the network have increased the security of these messages and prevent disclosure of this information as far as possible, this paper presents a new method for improving security of these messages by using steganography and cryptography method together.

Steganography is an art of hiding information. The goal of steganography is to have invisible communication in completely undetectable manner whereas the goal of cryptography is to secure communication from an eavesdropper. Images are ideal for information hiding [12] because of the large amount of space is created in the storing of images. Steganography consists of methods of transmitting secret messages. These secret messages are transferred through unknown cover carriers. In this method before hiding message into a cover-image; message is encrypted first by using AES algorithm and then this encrypted message is processed to hide into an image so that stego-image contains hidden message which is not in plaintext form. Another important point is that we are hiding encrypted message into an image using "Random LSB Steganography" that is embedding data in non sequential LSB insertion pattern so that it is unintelligible and unreliable to detect.

II. Choice of Image

Images in JPEG format are very poor choice for cover-images. This is because quantization introduced by

Manuscript received May 9, 2012; revised June 13, 2012.

Authors are with the Department of Computer Engineering, Government College of Engineering & Research, Awasari, Pune, India (e-mail: patu_pawar@yahoo.co.in; shgawande@yahoo.co.in).

JPEG compression can serve as a "watermark" or unique fingerprints and you can detect very small modifications of the cover-image by inspecting the compatibility of the stego-image with the JPEG format [10].

Image compression techniques are extensively used in steganography. Among the two types of image compressions, lossy compression and lossless compression, lossless formats offer more promises. Typical examples of lossless compression formats are CompuServe's GIF (Graphics Interchange Format) and Microsoft's BMP (Bitmap) [13].

In this work, for e-banking we are using BMP format and for m-banking we are using GIF format.

A. LSB in BMP for E- Banking

When images are used as carrier in Steganography [Figure 1] they are generally manipulated by changing one or more of the bits of the byte or bytes that make up the pixels of an image. The message is stored in the LSB of one color of the RGB value. A BMP is capable of hiding quite a large message [Fig. 2].

Fig. 1. BMP image as Cover-image

Fig. 2. BMP image as Stego-image

B. LSB in GIF for M-Banking

The amount of information that can be hidden in GIF images [Fig. 3] is less as compared with BMP Images. Embedding information in GIF images using LSB technique [Fig. 4] results in almost same as those of embedding in BMP.

III. Algorithm

Normally in e-banking and m-banking user requests such as credit balance of the account. Information is sent directly after the user request. While sending information directly it is possible that hackers might access and disclose the user's information.

In our method instead of direct sending information in plaintext form we are encrypting this information using "Advanced Encryption Standard" algorithm and then encrypted information is hidden into the picture by using password and "Random LSB Steganography algorithm". This stego-image is placed in another website and address of that website is sent to user. User downloads the picture from website. User extracts information from picture by using password then user gets information in encrypted form. User decrypts information so that he gets result. These methods act as follows;

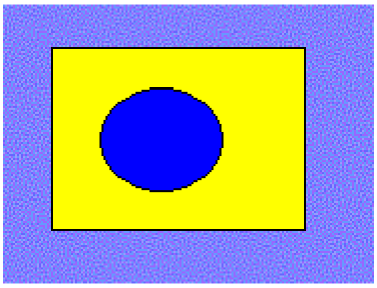

Fig. 3. GIF image as Cover-image

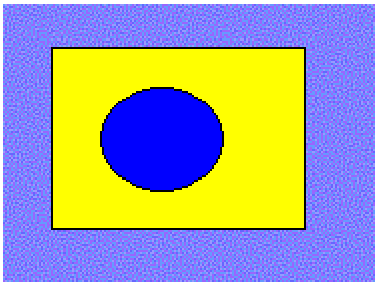

Fig. 4. GIF image as Stego-image

A. Encryption Algorithm

The encryption algorithm that we used is the AES Rijndael algorithm [11]. AES Rijndael is an iterated block cipher, meaning that the initial input block and cipher key undergo multiple transformation cycles before producing the output. The algorithm can operate over a variable-length block using variable-length keys; a 128-, 192-, or 256-bit key can be used to encrypt data blocks that are 128, 192, or 256 bits long, and all nine combinations of key and block lengths are possible. The algorithm is written so that block length and/or key length can easily be extended in multiples of 32 bits, and the system is specifically designed for efficient implementation in hardware or software on a range of processors.

AES Rijndael is a substitution-linear transformation network with 10, 12 or 14 rounds, depending on the key size. A data block to be encrypted by AES is split into an array of bytes, and each encryption operation is byte-oriented. AES's round function consists of four layers. In the first layer, an 8x8 S-box is applied to each byte. The second and third layers are linear mixing layers, in which the rows of the array are shifted, and the columns are mixed. In the fourth layer, subkey bytes are XORed into each byte of the array. In the last round, the column mixing is omitted. The AES Rijandael implementation was taken from the Legion of the Bouncy Castle cryptographic package [14] which provides a Java implementation for the algorithm.

In our application we used a block size of 16 bytes processed with 128-bit keys: this proved to be the best combination for operation on J2ME devices due to the speed and memory limitations of such devices.

The client and the server use two 128-bit keys, one for each direction of data travel. That is, one key is used to encrypt the data in the client and decrypt it in the server, and the other is used to encrypt the data in the server and decrypt it in the client. At the start of every client session, the server randomly generates this pair of keys and stores them in the client's specific entry in the database. The server then encrypts these session keys using the client's 64-bit pin code padded to a 64-bit shared secret known to the client and the server.

B. LSB Steganography

This method hides information in the least significant bits of pixels. In this method each byte of information is hidden in two pixels. For hiding information a byte is divided into a eight bits. By using password two pixels are selected in which a byte of information is hidden. An algorithm in [1] is used to select pixels to hide data.

In this algorithm image is segmented into n block of m pixels. A block is selected according to password and the information is hidden in an empty pixel of this block. The algorithm for selecting a block and an empty pixel in that block as follows:

If the selected block starts with the pixel number x and has m pixels then the number of last pixel is x+m-1. This algorithm uses an array of size m+1 for remembering empty pixels of current block. This array contains the number of pixels having no data. The last cell of the array is the total empty pixels in the current block. According to the password, an empty pixel is selected and the last empty pixel number is copied to this array cell. After this operation the total number of empty pixels on the block decreases by one. This method is also used for selecting a block to hide the information in itself. After selecting the pixels we hide a byte within them. Each pixel has three colors (RGB), and the information is stored in the LSB of these colors.

IV. Implementation

The application we designed and implemented provides a prototype solution for securing sensitive data. This section presents a brief discussion of the design starting with the client environment and moving on to the server environment. In this paper we are describing implementation of m-banking service.

A. The Client Environment

On the client side we used the J2ME wireless toolkit 1.0.4 [6] provided by Sun. The wireless toolkit is a set of tools that provide J2ME developers with the emulation environments, documentation, and examples to develop MIDP-compliant applications. Developers are thus able to check the valid operation of their applications before deploying them on actual physical devices. The MIDP application is packaged inside a Java archive (JAR) file, which contains the application class and resource files. This JAR file is actually downloaded to the physical device (mobile phone) along with the Java application descriptor file.

In the client environment user sends request, receives address of website on which picture is saved, downloads picture, extracts information from picture and decrypts information to get required results. After these operation user can do banking services like account balance, transaction, ministatement, cheque etc. During internal and external transaction user has to give password which user used at the time of extracting information from an image. If password is wrong then transactions get failed.

B. The Server Environment

To benefit from a pure Java solution, we implemented the server-side application in accordance with the J2EE specifications [5]. Servlet classes are packaged in a web archive (WAR) file and deployed on the J2EE application server. We used the J2EE reference implementation server version 1.3.1 provided by Sun. The database server we used is the Microsoft SQL Server.

The Java Servlets communicate with the database using the well-known Java database connectivity (JDBC) API and the new javax.sql package. Some of the services addressed by the javax.sql package are connection pooling, distributed transactions and data source retrieval using logical names. Instead of loading the specific JDBC driver each time we want to connect to the database, we used the Java naming and directory interface (JNDI) to retrieve the data source using its logical name from a JNDI-complaint directory service on the J2EE server.

In the server environment there is authentication Servlet to authenticate client, service Servlet which provides services requested by client like account balance, transaction, ministatement, cheque etc. This environment also contains encryption program and lsb encoding program.

V. Advantages

1. This method is compatible with many types of mobile phone.
2. In this method before steganography in the picture, encrypted information is encoded by a password therefore if person manages to extract information from the picture he will not be able to decode it without having the password.
3. In this method the information is never placed on the internet. Thus, the possibility of disclosure of information is very low.
4. In this method use of combination of steganography and cryptography provides strong secure and invisible communication.
5. The Random LSB steganography algorithm advantages are:
    a. Message is embedded in non sequential LSB insertion pattern so it is difficult to detect LSBs in which message is embedded.
    b. Because the password is used, it is difficult to detect the information hidden in the image.
    c. The decoding program uses a few kilobytes of memory. Also the program is fast enough.
6. For each access to bank service we are using security code which is known to that particular user.

VI. Conclusion

In this paper authors shown that security of e-commerce and m-commerce has been improved using random LSB steganography and cryptography method together instead of using either steganography or cryptography. Steganography algorithm can be changed based on the requirements of concerned m-banking system. Also, we can choose the encryption algorithm depending upon the processing time required to encrypt the information. This method can be used on all types of java enabled mobile devices.

References

[1] M. Shirali Shahreza, "An Improved Method for Steganography on Mobile Phone", *ICS'05 Proc. of the 9th WSEAS International Conf. on Systems*, vol. 4, no. 7, 2005, pp. 955-957.

[2] K. Pousttchi, M. Schurig, "Assessment of Today's Mobile Banking applications from the View of Customer Requirements", *Proc. of the 37th Hawaii International Conf. on System Sciences*, Big Island, Hawaii, vol. 7, no. 7, 2004, pp. 170-184.

[3] Turowski, K.; Pousttchi, "Extending knowledge management to mobile workplaces", *ICEC '04 Proc. of the 6th international conference on Electronic commerce*, 2004, pp. 583-590.

[4] P. Gerrard and J.B. Cunningham, "The diffusion of Internet Banking among Singapore consumers", *International Journal of Bank Marketing*, vol. 21, no.1, 2003, pp. 16-28.

[5] W. Itani and A. I. Kayssi, *J2ME end-to-end security for M-commerce*, vol. 3, no. 20, 2003.

[6] Sun Microsystems, *J2ME Wireless toolkit 1.0.4 User's Guide*, 2002.

[7] N.J. Black, A. Lockett, C. Ennew, H. Winklhofer, S. McKechnie, "Modeling consumer choice of distribution channels: an illustration from financial services", *International Journal of Bank Marketing*, vol. 20, no. 4, 2002, pp. 161-173.

[8] H. Karjaluoto, "Selection criteria for a mode of bill payment: empirical investigation among Finnish bank customers, *International Journal of Retail & Distribution Management*, vol 30, no. 6, 2002, pp. 331-339.

[9] H. Karjaluoto, M. Mattila, and T. Pento, "Electronic banking in Finland: consumer beliefs and reactions to a new delivery channel", *Journal of Financial Services Marketing*, vol 6, no. 4, 2002, pp. 346-361.

[10] J. Fridrich and M. Goljan, "Steganalysis based on JPEG compatibility", *SPIE Multimedia Systems and Applications IV*, vol. 4518, no. 275, 2001.

[11] J. Daemen and V. Rijmen, "Rijndael: the advanced encryption standard," *Dr. Dobb's Journal*, vol. 26, no. 03, 2001, pp. 137-139.

[12] W. Andreas, A. Pfitzmann, "Attacks on Steganographic Systems", Third *International Workshop, IH'99Dresden Germany, October Proc., Computer Science*, vol. 1768, 1999, pp. 61- 76.

[13] T. Jamil, "Steganography: The art of hiding information is plain sight," *IEEE Potentials*, vol. 18, no. 1, 1999.

Pratiksha P. Pawar was born on 19th December 1983. She received B. E. and M. Tech. degrees in computer engineering from College of Engineering Pune, University of Pune in 2005 and 2009 respectively.

Currently She is working as Assistant Professor in Computer Engineering department of Government College of engineering and Research, Awasari, Pune. She is life member of ISTE.

S. H. Gawande was born on 4th July 1979 in small village Deori Tq. Akot, Dist. Akola in Maharashtra State. He completed B.E. degree in mechanical engineering from Amravati University, Amravati in April 2001 and M.E. degree in mechanical engineering with design engineering as specialization in December 2002 from University of Pune. Now he is working as Assistant Professor in mechanical engineering at M.E.S. College of Engineering Pune, India from 2004. His research interests include internal combustion engines, design engineering, and Tribology. He is permanent member of Indian societies like ISTE from 2005, SAE from 2008 and IACSIT Singapore from 2009.